Andrew Ibsen

TOKENIZING AIRCRAFT ASSETS ON A DISTRIBUTED LEDGER

An inventory management solution built for British Airways Engineering.

Submitted as part of the requirement for the award of
MSc in Aircraft Maintenance Management
At City, University of London

I certify that this project is wholly my own work and in accordance with the project regulations.

All material that has been extracted from others has been clearly referenced.

This Project complies with the Project Regulation and Guidelines.

Signed: 

Supervisor: Dr. Paulo Nunes

Date: 28/02/2024

Word Count: 14676

(Excluding Table of Content/ Glossary, Abstract, Appendices and List of References)

(Restricted/UNRESTRICTED Circulation) select which option.

Executive Summary

Blockchain technology has the potential to revolutionize inventory management by providing a secure and transparent way to track and manage inventory (Henry and Kathawate, 2023). By tokenizing inventory items on a distributed ledger, companies can increase accuracy and security of inventory management operations (Bhushan et al., 2023). Thus, it can be hypothesized that implementing a blockchain framework for inventory management in an airline's engineering operation will increase accuracy of part's locations and facilitate maintenance tracking, promoting cost-control. This hypothesis was tested within British Airways Engineering operation at John F. Kennedy Airport. The prototyped solution was developed after analysis of the current system and ongoing consultation of potential users. Initial surveys highlighted moderate levels of satisfaction with the existing system with pain points around item retrieval effectiveness. To leverage the airline's existing secure online environment, a Microsoft Power Apps application was built as a front-end to transact inventory items as it is receipted in and out of storage. The application is a gateway to assets listed with unique identifiers on Microsoft SharePoint. Receipting in the mobile application is replicated in a blockchain (Sepolia Testnet) component. The prototype was deployed to 25 engineers and its use tracked through a logging system. Results reveal moderate user acceptance with moderate-to-good usability and reports of improvement on efficacy of item retrieval. Significant improvements in average accuracy ($p<0.05$) and time spent transacting ($p<0.05$) were obtained when systematically testing the new solution against the legacy system. Spot checks found a significant decrease in misreported parts locations, saving an estimated $150'000 to the operation after the first 3 weeks of deployment only. Thus, blockchain has improved supply chain transparency and traceability, potentially prevented loss of parts and increased efficacy of item retrieval. This, in turn, is expected to reduce operational costs for the line maintenance station and increase aircraft turnover rates. Furthermore, the error-proof data collected could be monitored periodically to optimize stockholdings and further benefit the operation. Because the solution is low-cost and functions within the airline's secure environments, it can easily be scaled to larger BA operations.

Contents Page

Contents

1 Introduction

1.1 Context

In 2024, an Alaskan Airline flight lost, in mid-air, a door plug. In the aftermath, the Federal Aviation Authority (FAA) grounded all Boeing 737 Max-9 aircraft in America. Multiple flight cancellations ensued. Alaskan Airlines reported a related cost of 150 million USD. United Airlines, having 79 Max-9 in its fleet, also attributed to this event falling short on their projected quarterly earnings. The ongoing FAA investigation centers on missing bolts meant to secure the door plug in place. Likewise, in 2019, 300 Boeing's 737 jets were fitted with faulty wing components, engendering consequential losses for all owner airlines. At the time, the FAA proposed a $3.9 million civil penalty against Boeing, highlighting the potential vulnerabilities from third-tier suppliers to final assembly as well as certification, and underscoring the critical need for enhanced traceability for quality control in the aviation supply chain (FAA, 2019). Aircraft are ever-more complex machines that, due to their intense commercial use, incur heavy wear and tear. To maintain high safety standards, each component, from bolts to auto-flight computers, must be thoroughly inspected and properly fitted. Thus, manufacturers and commercial aviation operations must continuously demonstrate high technical knowledge and efficient aircraft maintenance procedures.

In commercial aviation, engineering also has the mandate of ensuring standards within the cabin to paying passengers. The most common demands stem from faults in seats or in-flight entrainment systems, which are central to an airline's hospitality and profitability. In fact, seats are a perishable good, in the words of Abriani, 2019, "you have only one chance to sell a seat prior to departure". Failure in timely maintenance causes seats to be locked out, especially in premium segments such as business and first-class (Nast, 2023) even though those segments, accommodating only 12% of travelers generate over 25% of revenue (Brock, 2022). Not being able to quickly identify if items are available in stores and install them during a transit is thus an important factor in lost revenue due to unsellable seats.

Furthermore, maintenance delays often conflict with crew duty time limits. Regulatory agencies determine maximum amount of time spent in air v. ground time for crews. Once crews are onboard, even if the aircraft has not departed, time starts to be counted against maximum time spent in air. Thus, flights are regularly canceled because of prolonged emergency maintenance conducted while the crew is onboard. As such, difficulty retrieving necessary parts from storage is a contentious point. For instance, recently at John F. Kennedy (JFK), Airport an Iberia aircraft had a malfunctioning CF6 Engine Starter at departure. The engineers on the ground soon realized that the Iberia stock at JFK did not hold a spare starter. This was communicated to Iberia, which sourced a starter from partnering operators at JFK Airport in an hours' time. Unfortunately, when adequate maintenance was eventually completed, the pilots had run out of duty time. Thus, despite successful engineering processes and aircraft readiness, the flight was cancelled. A posteriori, a starter was discovered in stock at British Airway's store's facility. The hour spent sourcing a new starter could have been saved and a $1 million (iagcargo, 2018) loss due to an Aircraft On Ground

(AOG) situation, avoided. The anecdote highlights how inventory lists must be up to date, centralized, and readily available, to fit the time constraints of commercial aviation.

Beyond the importance of time efficacy for part retrieval, misplaced parts are another pitfall. Lost parts are a regular but rarely divulged occurrence in airline operations, with recurrent monetary impact. For the most part, airlines do not have systematic report systems in place to monitor such events. In spring 2023, BA Engineering, serving its mandate as Iberia's maintenance provider at JFK Airport, removed a faulty Integrated Drive Generator (IDG) from an Iberia aircraft and installed a new part. In summer 2023, Iberia contacted British Airways enquiring about the IDG that had been removed and was believed to be in the Iberia's stockholding maintained by BA at JFK. BA Engineering, upon consulting maintenance records, reported that the IDG had been sent back to Iberia in Madrid. Contradictory paperwork had been filed by both airlines regarding the location of this IDG. The conflict was only resolved when, months later, an unaccounted for and unmarked box was uncovered by chance at a shipping warehouse facility in JFK Airport. The part in question is worth, new, $ 250'000, and circa $175'000 used (Aerobay 2023).

Perhaps more frequent that all together lost parts, are arrival of parts with incorrect paperwork. The lack of documentation causes the parts to be returned to the sender, causing stockholdings to be ill-equipped for rapid aircraft turnover. Lack of paperwork, beyond delays, raises a grave safety question. In fact, counterfeit parts are estimated to represent as much as 10% of aircraft parts on the marketplace (Kotzé and Antonopoulos, 2022). In the summer of 2023, the company AOG 24-part was guilty of 86 instances of forged documents. (Sean Broderick, 2023), including EASA Form 1s for high valued engine components. These engine items with fake documentation were found 96 engines, a number that might increase with further inspections and investigations. The rampant counterfeits issue is not successfully addressed by existing controls.

For part storage facilities, regulations are enforced through audits. Audits of stores are conducted by airline's quality departments on an annual to bi-annual basis according to the responsible national aviation authorities' guidelines. The audits verify that all materials, parts, and components are regularly recorded, and stock checked, with each item tracked through stock management and record keeping software such as the System Applications and Processes in Data Processing (SAP). These audits are conducted by random sampling of parts, tools, and time frames. No continuous data is analyzed, thus giving leeway to unidentified errors. For instance, oxygen bottles and electrostatic sensitive devices (ESD), such as a flight control computer, must be kept within a strict temperature and humidity range. A one-day peak in temperature and humidity levels can render such parts unserviceable, incurring high costs to the airline. However, the chances of the next audit specifically enquiring about climate conditions on the exact day the error occurred are sparse. In addition, only grave findings, labeled level 1 findings, such as parts out of calibration, must be reported by the airline's quality department to the overseeing national aviation regulatory body. With this audit structure, it is left to the airline's quality inspector conducting the audit to decide whether a fault qualifies as level 1, a decision often biased by their personal relation to the manager in charge of the engineering station, both co-workers at the same company. Finding

faulty parts when attempting to install them in a quick turnover is detrimental to the airline's revenue. However, and most importantly, the fault in the parts caused by improper storage might not be detected until the aircraft is in the air, endangering lives.
In aviation, parts are often fixed and recirculated, either within a company or through part vendors. In fact, the lifecycle of a part often involves multiple companies and storage facilities around the world. The only common practices and window into the conditions in which parts were kept are regulations and forms, respectively, mandated by the homogeneous guidelines of national aviation regulatory bodies around the world and audited as previously described. When it comes to reporting flaws and finding innovative solutions, companies rarely emphasize stores. The warehouses, although at the technical heart of the operation, in most cases are not optimized for efficiency, do not receive investment in qualified stores keeping personnel or modern automated tools. Investments in engineering center on retaining personnel and training teams to obtain necessary type ratings, a challenge, especially in a post-COVID industry. Some limiting factors do exist, besides audits, an aircraft with missing tech logs for instance, sees it resell value diminished. But these far-down-the-line taxes are viewed as minor issues by airlines. Furthermore, if the consequences of storage and documentation errors will reveal themselves immediately within the causing airline's operation or further along the line, at the next owner's, is unpredictable. When thinking beyond one airline and considering the life cycle of parts, the problem is exponentiated, opportunities for undetected error in storage conditions and mistakes in paperwork abide. Thus, the true consequences and cost of poorly stored parts and lenient quality checks are difficult to trace as they ripple through the whole aviation industry.

1.2 Hypothesis statement and Study Objectives

There is a problem in traceability and accountability in the aviation supply chain. Storage facilities in commercial aviation's engineering departments, a crucial link, lack efficient inventory management systems. Here, we hypothesize that implementing a blockchain framework for inventory management in an airline's engineering operation increases accuracy of part's locations and facilitates maintenance tracking, promoting cost-control.

The study will be conducted at British Airways Engineering operation at JFK Airport, which contracts local maintenance and is responsible of inventory holding for six other international airlines. The study's objectives are to:

- understand the current strengths and limitations of spares management at BA's JFK station, which serves BA Engineering and its client airlines,
- build a user-friendly inventory management system, capable of tokenizing spares,
- test the viability of tokenizing spare parts in BA's hangars,
- qualify and quantify the impact of the proposed solution to tokenize spare parts.

2 Literature Review

An inventory management system is defined as a group of rules and controls that detect, manage, and maintain inventory level (Jacobs and Chase, 2018). Because this study's objective is to build a stores-keeping solution that serves the needs of British Airways Engineering operation at JFK Airport, the review will focus on current inventory management solutions commonly implemented in aviation, novel solutions from other industries with similar constraints and governing regulations within the civil aviation authority. The aim of the review is to extract recommendations for the study's next steps.

2.1 Current solutions commonly implemented for inventory management.

2.1.1 Current Inventory Management Software: AMOS, SAP, IFS (Maintenix)

In the past 20 years numerous airlines have switched from a patchwork of inventory management systems to company-wide use of one software. British Airways has adopted the SAP system, known as the Engineering Wide System (EWS), a comprehensive and integrated enterprise resource planning solution that streamlines aircraft maintenance across 26 hangars and 142 airports worldwide. Replacing the 150 legacy systems required a substantial effort, involving 23,000 man-days of classroom training, on-the-job guidance from 1,000 system advisers, and 200 process transaction manuals. Despite its initial challenges, it ultimately resulted in significant cost-saving and increase in efficiencies (McCue, 2006). Similarly, China Airlines faced challenges with legacy mainframe systems, which were siloed and inefficient. They adopted IFS's (Maintenix), another integrated maintenance software, leading to a 65% reduction in multi-crew job card waiting time in heavy maintenance, 10% increase in line management and maintenance process efficiencies, and 14 additional revenue days per year due to reductions in scheduled and unscheduled maintenance (IFS, 2023). In the past, making the change from legacy inventory management systems to newer, more digitally enabled solutions has benefited airlines.

2.1.2 Inventory applications on mobile devices.

AMOS, a Swiss Aviation Software spin-off of the defunct SwissAir, prides itself of representing "Swiss Quality at its best" and provides software solutions for Maintenance Repair Overall (MRO) to a large and diverse clientele of 245 airlines and independent MRO providers of all sizes across the world. AMOS has recently introduced a customizable mobile version, see Figure 1, for inventory management; however, it does not run on the blockchain. The company highlights the benefits of the app as producing a "Decrease of loan costs and AOG orders due to wrong inventory, reduce likelihood of delays"(AMOS, 2024). This report shows a trend and a demand for practical mobile applications.

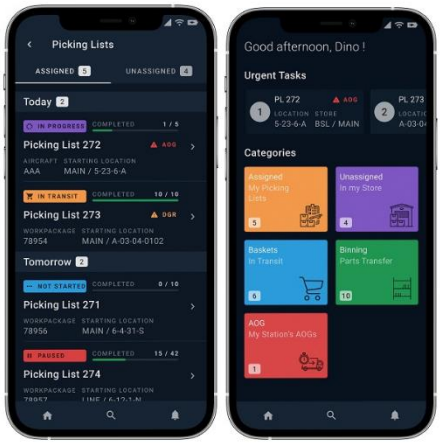


Figure 1 - AMOS Mobile Application, reproduced from AMOS, 2024.

2.1.3 Automated custom mobile apps for inventory management in other industries

In other industries, custom mobile applications for store keeping are commonplace. Ubald et al., 2022, documented their development process for a mobile app prototype for addressing inventory management challenges in the pharmaceutical industry, applying the RUP development methodology and testing various computer tools. The app prototype was built after a detailed analysis of the functionalities needed, to fulfill usability requirements while maintaining a high level of security in a well-presented format. The prototype was then evaluated via a survey of the users and experts. Their results included 92% of users surveyed stating that they completely agreed with the authors regarding the prototype's high security, usability, and functionality. Although the phrasing of the questions could be quite leading, overall customers and experts alike appeared satisfied with the prototype. Interestingly, app aesthetics played an important role in user satisfaction. The systematic approach in development and testing used by the authors is of interest when developing inventory management applications, especially in domains that require high security (Alfredo Leonidas Vasquez Ubald, 2022).

In turn, Yankah et al., 2024, reviewed 38 mobile applications for document management in construction projects. The authors further narrowed their focus to the 21 applications among these that demonstrated best relevance, user friendliness and compliance with the app's goal. In doing so, the authors explore challenges faced by the construction industry's current Document Management Systems (DMS). The paper recommends an ideal mobile app solution capable of streamlining workflows, incentivizing collaborations, and providing access to documents in real-time, these 3 core characteristics ultimately enhancing accuracy and efficiency. Their recommendations can be generalized to solutions for inventory management in other industries where documentation is of essence (Yankah et al., 2024).

In line with developing digital transformation tools, QR codes can streamline the process by instantly updating and accessing item records in a central database, simplifying tasks like adding inventory, tracking movement, and conducting audits. Morales-Hernandez, 2021, provides a step-by-step guide on setting up a QR code inventory system, emphasizing benefits like quick data entry, real-time updates, and cost savings by using smartphones instead of specialized scanning equipment. This approach is interesting for any inventory management by highlighting the role of QR technology in enhancing operational efficiency and accuracy (Morales-Hernández et al., 2021). However, efficient QR coding for managing inventory also implies methodically organizing physical stores, a domain in which research has also delved, notably in respect to the Dewey-Decimal System (Satija, 2013), based on subject matter and widely employed in libraries, or the Hierarchical Location System.

2.2 Blockchain technology for inventory management

2.2.1 Blockchain – leveraging a centralized computer application with the decentralized system of blockchain.

Decentralized applications, making use of blockchain technology, have also been considered for inventory management purposes across industries.

The term Blockchain was popularized by Satoshi Nakamoto from his whitepaper “Bitcoin: A Peer-to-Peer Electronic Cash System”. This was the premise for a decentralized digital currency. The paper highlighted problems with traditional online payments and the need to overcome a trusted third party when transacting. It highlights a system called proof of work, where the validity of transactions and the creation of new Bitcoins are ensured by miners performing computational work. These miners are in turn rewarded with new bitcoins, incentivizing them to conduct computational work to keep the system going. The system uses the basis of cryptography, by computationally time-stamping digital transactions by hashing them into a continuous chain of hash-based proof-of-work, forming an immutable record that cannot be changed. The proof-of-work which validates the transactions is grouped together into blocks by miners. Independent miners are responsible for processing and verifying the transactions. The processing of the transactions is done by solving complex mathematical problems, this is the proof of work concept. When the mathematical problem is solved, a consensus is made between the miners. As the miners are independent, they create a decentralized network it negates the need for a trusted third party (Nakamoto, 2008).

The decentralized blockchain network run on the Ethereum platform enhances this concept further by allowing prerequisite, smart contracts. Because Ethereum incorporates smart contracts, it is a distributed state machine, as represented in Figure 2 (Pettinari, 2023). The smart contracts are coded using the Solidity programming language (Solidity, 2024). Solidity is based off other commonly used programming languages such as C++, Python and JavaScript. With Solidity you can create smart contracts that transact automatically with the Ethereum blockchain.

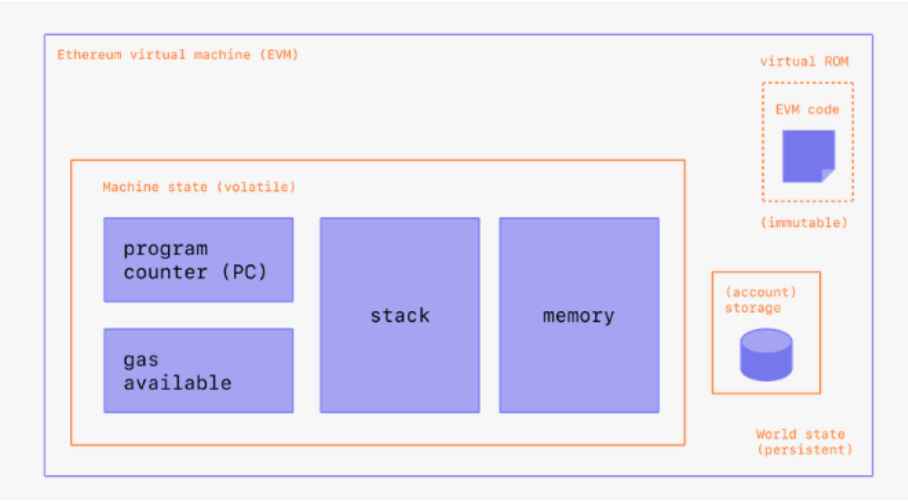


Figure 2 - Diagram of the Ethereum Virtual Machine (EVM), reproduced from Pettinari, 2023.

Ethereum smart contracts are complemented by decentralized applications (dApps), which are programs based on the blockchain. This is fundamentally different to normal applications that run on a centralized system. The dApps run on a peer to peer (P2P) network that is Blockchain based (Cai et al., 2018). Decentralized applications (dApps) offer advantages like fault tolerance, privacy, data integrity, a flexible development platform, and verifiable behavior through smart contracts (Antal et al., 2021). However, they face challenges such as performance overhead, maintenance difficulties, scalability issues, complex user experience, potential centralization, and network congestion (Pop et al., 2021). These pros and cons highlight the balance between the innovative features of dApps and the practical considerations in their implementation and use.

2.2.2 Use cases in industries other than aviation.

Much has been published and speculated on benefits of blockchain technology for inventory management, generalizing across industries. In fact, blockchain technology is believed to have potential to revolutionize inventory management by providing a secure and transparent way to track and manage inventory (Henry and Kathawate, 2023). By leveraging blockchain, it has been claimed that companies can increase transparency, accuracy, and security of inventory management operations (Bhushan et al., 2023).

Some direct use cases of blockchain technology for inventory management have been further detailed. For instance, blockchain can be used to enable environmental, social, and governance (ESG) tracking through supply chain traceability (Henry and Kathawate, 2023). Additionally, blockchain can help companies proactively detect and mitigate supply chain risks before any severe impact occurs.

A paper titled "A System for Storing Anonymous Patient Data in Healthcare applications" introduces a solution aimed at enhancing the privacy and security of healthcare data through a blockchain application (Oksuz, 2024). The significance of anonymizing patient data to ensure secure access and sharing of health records was emphasized. The method of employing pseudo-random identities for patients to mitigate the risk of health record leaks was discussed, highlighting its potential to substantially improve data privacy. This approach was compared with existing data storage practices, illustrating its potential impact on the confidentiality and integrity of patient health information in the digital healthcare landscape. The paper's methodology and proposed solution were critically examined within the context of current trends in healthcare data privacy, underscoring its relevance and possible implications for future research and technological advancements in healthcare information systems. Using the blockchain to encrypt data, adding an extra layer of security would be beneficial to the aviation industry who has struggled with forged parts.

Additionally other industries such as Agriculture have been using the IOT framework integrated with blockchain to enhance transparency and trust in the agricultural food supply chain. A paper called "A Framework for Agricultural Food Supply Chain using Blockchain,", discusses how blockchain technology can overcome challenges related to tampering, traceability, and the complexity of global trade, thus ensuring food safety and integrity. The framework aims to leverage blockchain for secure, transparent transactions and IoT for real-time data collection, enabling a comprehensive view from farm to consumer. The paper covers related works, blockchain's benefits and limitations, and outlines a specific proposed framework for applying blockchain in the food supply chain (Sudarssan, 2024). The relevant data being collected is similar as aircraft item batch numbers are mirrored in the industry and storage being an important factor.

2.2.3 Upcoming solutions in the aviation ecosystem incorporating blockchain technology.

Ho et al., 2021, built a system on Hyperledger Fabric and Hyperledger Composer, which is an Ethereum backed system. The system introduced generates a digital twin following the validation and endorsement of transactions. It serves as a practical example for the entire aviation industry, establishing connectivity between Original Equipment Manufacturers (OEMs), airlines, and Maintenance, Repair, and Overhaul (MRO) entities. This integration is designed to support comprehensive 'back to birth' records, streamlining the traceability and management of aircraft parts. By linking these key segments, the system facilitates enhanced coordination and efficiency across the aviation supply chain. The proposed system integrates Radio Frequency Identification (RFID) technology into aircraft parts during their manufacture by OEMs. These integrated RFIDs serve multiple roles, primarily storing essential data about the part, providing essentially a birth certificate. They also facilitate logistics providers in managing inventory and enable airlines and MROs to access and augment traceability records with specific information. A performance evaluation was done to demonstrate the blockchain based system’s ability to scale using simulations. The results indicate a highly predictable

performance behavior under varying demand loads. The study mentions several limiting factors including the integration of the Digital Twin, data privacy and security concerns, performance during real world scenarios, interoperability with existing systems as well as compliance and approval across multiple aviation authorities.

In turn, SkyThread is a consortium-driven initiative aiming to improve data flow in aviation transactions by leveraging blockchain (SkyThread, 2022). SkyThread aims to create a Utility Layer that can provide the full account of aircrafts part history and maintenance, supporting over twenty applications to streamline aviation processes. The company is working with regulatory agencies to create a comprehensive industry standard. It anticipates creating revenues of approximately $11 billion, reducing operating costs by $6 billion, inventory costs by $14 billion, and increasing residual asset values by $2 billion.

The multinational information technology company Society of International Telecommunications in Aviation (SITA), tested a "Flight Chain" project which aimed at using blockchain to facilitate a single, reliable source of truth for flight status data, addressing the issue of multiple and often conflicting data sources in the industry (SITA, 2017). Currently, SITA claims to service about 90% of the world's airline business and a revenue of $1.3 billion a year, underscoring the potential impact of this novel solution. For the company, however, the benefits were mitigated by the complexities of the invention. In fact, with every new customer airline and airport, additional nodes would need to be created in a manual and ineffective process, preventing scalability. Since publications of these results in 2017, no further communications on the matter or advances on the project have been released by SITA.

2.2.4 Cybersecurity risks and regulations within the Civil Aviation Industry

All inventory management solutions and new technologies implemented in aviation must be zealous of potential dangers. In fact, civil aviation has been prone to cyber-attacks. The 5 top recent occurrences include a BA breach of 400,000 records in 2018, a SITA breach of 2 million records in 2021, an EasyJet breach of 9 million records in 2020, a Cathay Pacific Airways of 9.4 million breaches in 2018 and finally an Air Canada breach of 20'000 records in 2018 (cnsight, 2021).

A report from the Civil Aviation Cybersecurity Subcommittee provides further recommendations for improving cybersecurity in the aviation supply chain (Schwindt, 2023). It addresses the complexity and risks associated with the global and interconnected nature of the supply chain, emphasizing the increasing threats from hardware and software vulnerabilities, insufficient supplier vetting, and global access to aviation components. The report outlines objectives to improve supply chain safety, security, and resilience, including developing guidance for cyber supply chain risk management, establishing supplier trust, and implementing secure configuration management. It highlights the need for harmonized regulatory guidance and industry standards. Emphasis is placed on how the aviation industry

organizations must develop and implement effective anti-counterfeit policies and procedures. Counterfeit components include components that have been produced by an unauthorized party and not to specifications, components that are illegitimately sold as a higher specification component (e.g., with a higher environment rating, or components that are not permitted to be resold as they are from an aircraft that has crashed. Brokers may be considered the highest risk of sources of components, especially when these are not subject to aviation requirements for tracing component lots. Among others, the Commonwealth Nations are now required to submit attestation documentation related to protecting against counterfeit components.

Smart technologies can counter the risk of counterfeit components entering the supply chain. Reports of cyber-security incidents across the aviation sector between 2001 and 2021 have been systematically analyzed in a study by Ukwandy and colleagues, 2022. The study finds that the most prominent threats have been from groups that, paired with a state actor, have attempted to gain unauthorized access to information technology infrastructures, with the goal of using the intellectual property thus obtain for their partner nation's advancement. Authors point to the risk of multiplying potential attack surfaces with increasing computer-based IT systems and the practice of Bring Your Own Device (BYOD), valid for both customers and employees. The increased exposure that comes with automation can be mitigated, at least in part, by implementing cyber-hygiene measures, which include recurrently updating of system and anti-virus software, training personnel, and enforcing policies regarding data uploads and passwords. A sense of urgency is necessary as AI technologies develop and with the automated attacks take center stage. (Ukwandu et al., 2022). In developing a smart application for use within the airport environment and on aircraft during maintenance, these questions are essential to avoid creating additional cyber-security vulnerability in an era of relentless and potentially automated threats. As such, literature regarding cyber security risks and management in civil aviation showcases the dichotomy between how automation, IoT technologies and the increase in the use of data can optimize part management and significantly reduce the threat posed by counterfeit components, whilst also potentially providing additional attack surfaces for cyber-attacks, creating new cyber-security risks.

2.3 Conclusion of the Review

Publications regarding mobile applications for inventory management across industries highlight the importance of using a systematic approach in developing and testing prototypes. Key aspects to focus on are streamlining workflows, creating a collaborative interface, and providing real-time accessibility to the data. To provide such accessibility, a QR system approach, coordinated with a reorganization of the stores can be valuable. Decentralized applications appear as a valuable interface allowing for mobile, user-friendly, front ends to have their data secured by a blockchain, as summarized in Figure 3. Recommendations extracted from this literature review are detailed throughout the results section as they guide the project forward.

3 Methodological Approach

The aim of this project is to build, deploy and assess the outcomes of an inventory management solution that interacts with the block-chain and is custom-built for BA's engineering operation at JFK Airport (see 1.2). The approach taken is inspired by the literature review and represented in Figure 4.

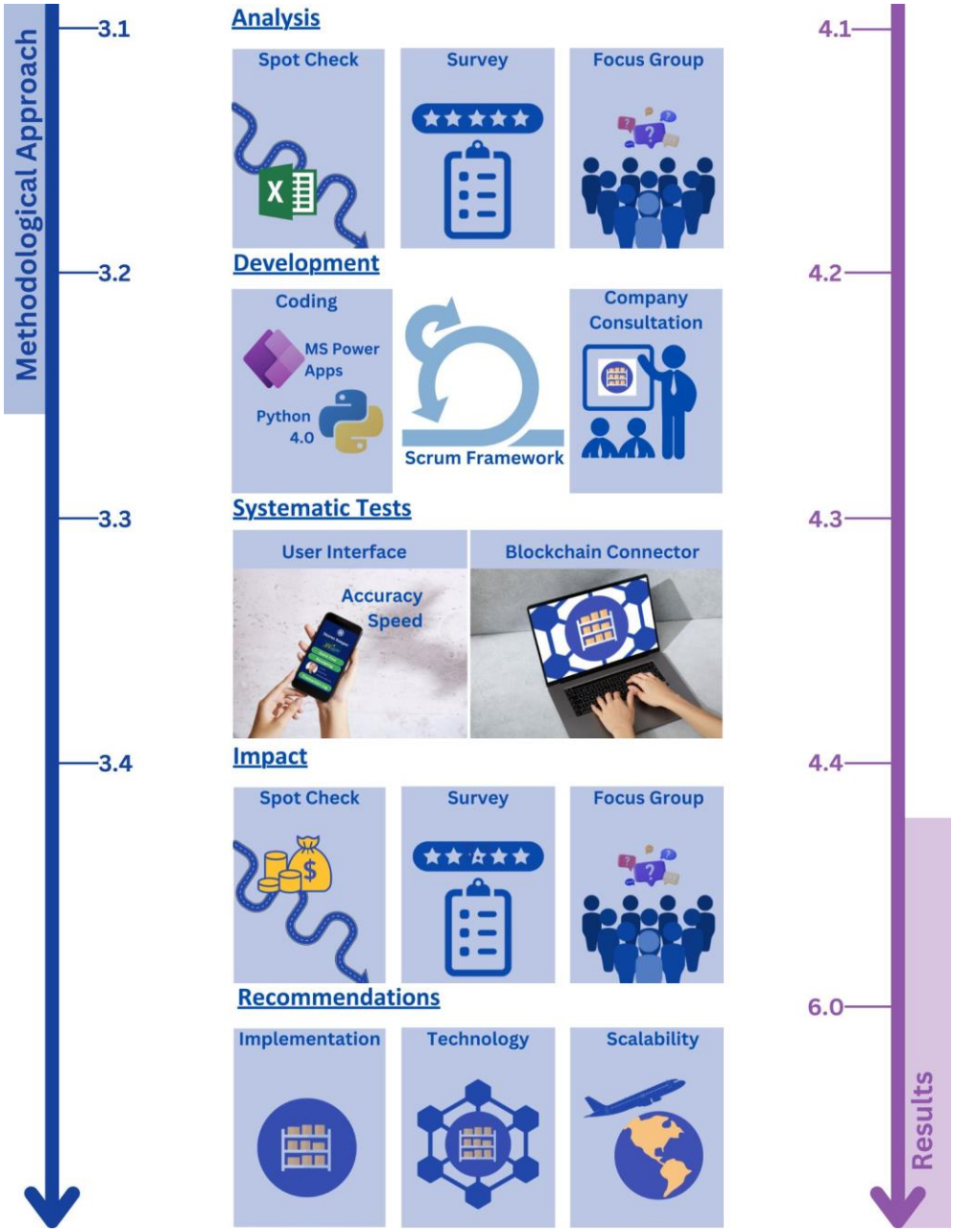


Figure 3 - Methodological Approach Including Analysis of the Stores at Time 0, Development of a Novel Solution, Systematic Testing of the Prototype and Estimation of its Impact, with the Aim of Extracting Useful Recommendations for Future Implementation and Scalability.

3.1 Analyzing the current inventory management spreadsheet system at JFK

Prior to building a custom system, the system initially present was analyzed (Figure 5).
The analysis includes:

1. A detailed description of the system's flows and limitations.
2. A spot check to estimate location errors. The cost of such location errors was extrapolated from tabulated parts pricing.
3. A JFK Stores Inventory Management Feedback Survey deployed to all Engineering team members anonymously.
4. A focus group with all team members and semi-structured interviews with targeted senior unit staff to discuss the results of the survey and delineate a new solution.

The aim of the survey and focus group was two-fold, to engage key stakeholders early on thus fostering a sense of involvement and investment in the project that would favor its later adoption and consulting with key stakeholders from the onset to ensure the project aligned with the organization's goals, addressing real operational needs. The interview sessions aimed at further building trust and open lines of communication to demonstrate the commitment to transparency and collaboration. This is key for a station such as JFK, where recurrent friction arises between unionized and non-unionized teams, working on BA aircraft and customer aircraft, respectively.

Figure 4 - First Step of the Methodological Approach: Analyzing the Stores at Time 0.

3.2 Developing the solution

A mobile inventory management application was developed using Microsoft Power Apps. In turn, a blockchain component was created with Python 4.0. and its Web3.py library.

The SCRUM framework was employed for app development. With its focus on transparency, communication, and accountability, it can enhance aviation projects by enabling early identification and immediate resolution of safety-related issues through cross-functional collaboration (Rademan, 2023).

Furthermore, consultation with relevant BA personnel in operational maintenance management, in the quality department and in the information technology department was regularly conducted during development. Once the solution was prototyped it was discussed with the Chief Engineering Technology Architect and the Business Improvement and Innovation Manager, directly under the Chief Technology Officer of the airline. See Figure 6.

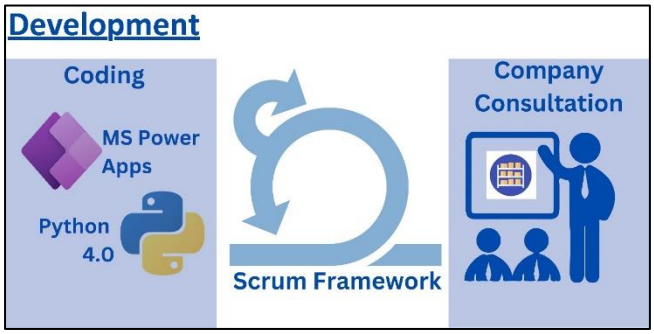


Figure 5 - Second Step of the Methodological Approach: Developing a Stores-Keeping Mobile Application Using Ms Power Apps and a Blockchain Interface, Called Decentralized Application (sApp), using Python 4.0.

3.3 Systematic tests of the App and the Blockchain system

Once a prototype of the mobile app was ready, an experiment was conducted to systematically assess the proposed application. Analysis on the quantitative output of the experiment has as its main outcome (1) the accuracy of the transactions performed and (2) the time taken to complete the transactions. Both outcomes were subject to a repeated measures ANOVA of between-subjects factor role (mechanic v. maintenance reps v. managers), and within subject's factors system employed (app v. analog) and type of transaction (in v. out v. transfer), with a significance threshold of $p<0.05$.

To test the efficacy of the blockchain twin in preventing inventory errors, a set of simulations, scripted in Python, was conducted. See Figure 7.

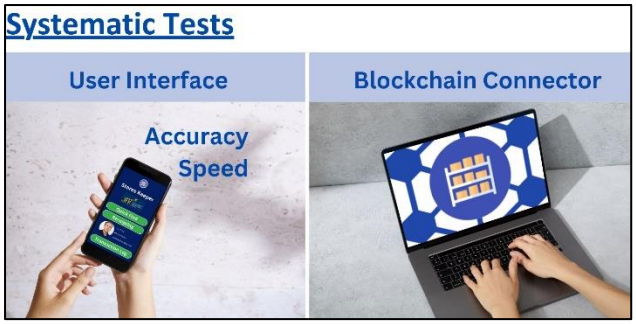


Figure 6 - Third Step of the Methodological Approach: Systematic Testing of the Mobile Application and of the Blockchain Interface.

3.4 Evaluating the impact of the application on stores keeping accuracy

A new spot-check, two weeks after deployment of the prototype quantified location errors and their estimated cost. Transfer transactions performed via the app, linked to items being misallocated, were also quantified.

Additional targeted semi-structures interviews with senior unit staff, as performed at the onset of the projected were then repeated. A feedback meeting was ultimately conducted in the same format as the initial focus group, to gather additional information and conclude the project. See Figure 8.

Figure 7 - Fourth Step of the Methodological Approach: Deploying the Prototype and Measuring its Impact with Quantitative and Qualitative Metrics.

4 Results

Here we present the findings from an evaluation of the newly implemented blockchain-enabled management system, as compared to the system previously employed at JFK.

4.1 Analysis of the store's keeping system at the onset of the project

4.1.1 Description of the system at the onset of the project

Inventory management processes are summarized in the diagram below (see also Annex 1).

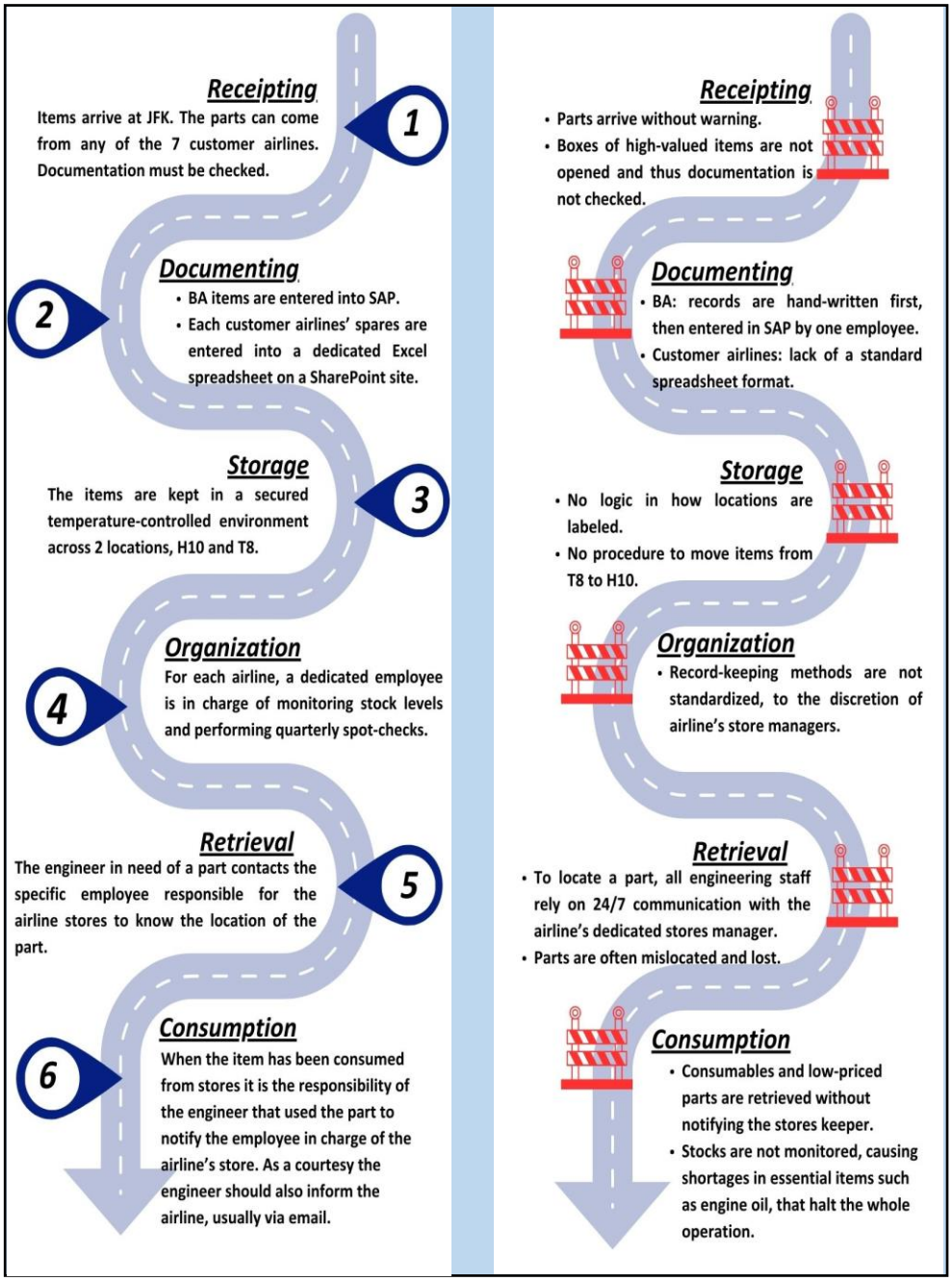


Figure 8 - Diagram of the Information Flows (left panel) and their Main Limitations (right panel) in Stores at Time 0.

4.1.1.1 Strengths and Limitations of the system in place at the onset of the project

For customer airlines, the system affords independence to JFK engineers in charge of each customer airline to best organize their spreadsheets to their liking and understanding, which can favor time-effective completion and tailor to professionals with different digital literacy skills. Furthermore, the record-keeping system is hierarchical, with the local spreadsheet being for internal use at JFK only, while the central record is held by the airline's inventory management system, updated from tech logs and direct reports of maintenance. Such a structure can prevent errors in one record system to impact the other. Lastly, there is no legal obligation as to what specific form of local control over stores must be implemented. As such, spreadsheets on the BA MS platform are compliant with national aviation authority regulations across the world all the while conforming to BA cyber-security standard. For BA's stores, the system is integrated with SAP and thus there is additional transparency for airlines and station engineers. With this, SAP is a system familiar to most engineers since it has been implemented by BA in 2004 and is used for a range of functions.

Several roadblocks can be identified in the current system for customer airlines, summarized above in Figure 9 (right panel). For a complete description see Annex 1.

4.1.2 Staff's views on the system in place at the onset of the project

4.1.2.1 Survey Results and Analysis

The organizational chart of the station is described in Figure 11. The station has 24 employees, of which 19 responded to the survey, or 80%. At least one employee in each level responded to the survey and distribution of responses across levels is representative of the distribution of employees, see Figure 12.

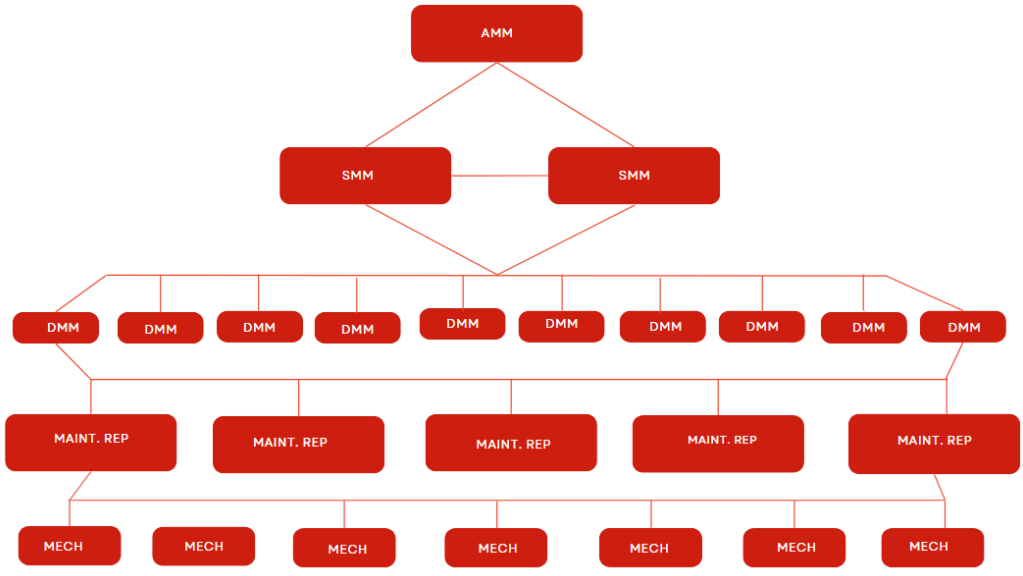


Figure 9 - JFK Engineering Organization Chart.

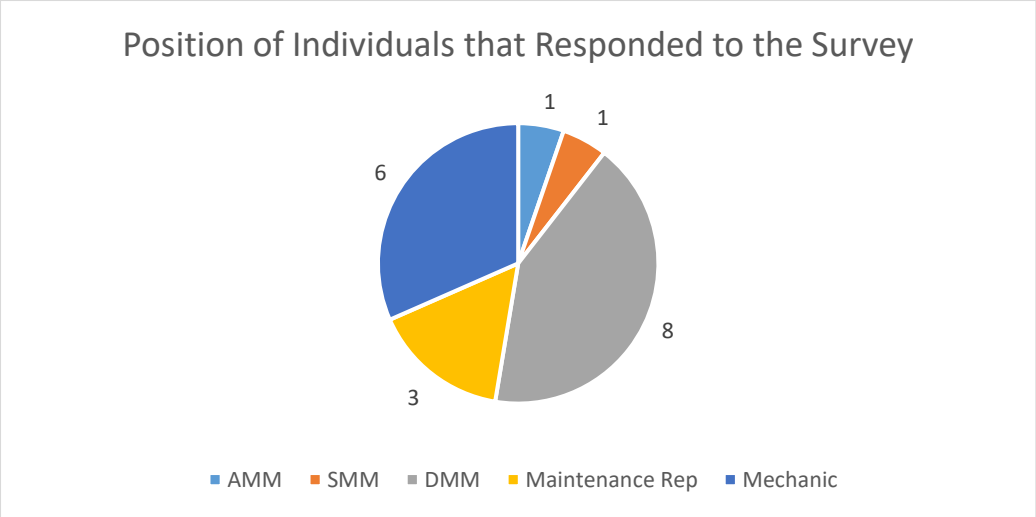


Figure 10 - Distribution of Employees Having Responded to the Survey at Time 0.

The results provide valuable insights , summarized below.

1. Moderate Satisfaction with Current System:
   - The average ratings for questions related to the current state of inventory management (Questions 3 to 18) is 3 out of 5.
2. Challenges in Notification and Documentation:
   - Lower scores on being promptly notified when new items are received (2.53 for Q4) and the ease of locating items (2.89 for Q11) point to challenges in communication and item retrieval processes. The average score of 3 for the accuracy and consistency of documentation (Q5) also suggests room for improvement in data management.
3. General Contentment with Store Maintenance:
   - Higher scores for the physical organization, cleanliness, and layout of the stores (Q8-Q10, Q13) indicate general contentment with the store maintenance and physical arrangement.
4. Operational Efficiency Needs Attention:
   - The efficiency of the check-out process (3.16 for Q17) and experiences of delays or issues in retrieving/checking out items (2.95 for Q18) have average to slightly below-average scores, highlighting areas where operational efficiency could be enhanced.
5. Strong Support for Digital Transformation:
   - Questions related to the implementation of a digital inventory management system (Q19 to Q29) receive high average ratings (above 4), suggesting that employees see significant benefits in a technological upgrade.

It's also worth noting the different roles and their perception between the unionized v. non-unionized managers (DMM/SMM/AMM) – see Figure 13.

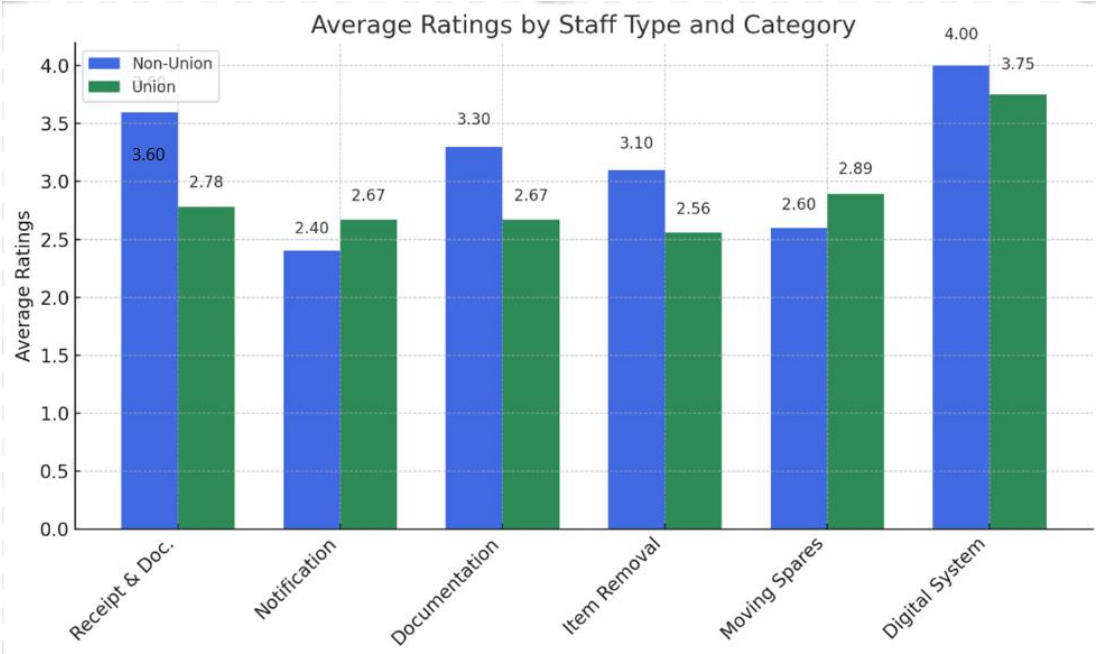


Figure 11 - Bar Graph of Union v. Non-Union Staff Views on the System at Time 0.

In turn, DMM's answered high 100% of the time for Q9,22,23 and 27. This indicates that the DMMs are highly aware of the operational aspects of the inventory system and are supportive of incorporating technological advancements.

In conclusion, these results indicate a recognition of the limitations of the current inventory management system and a strong inclination towards embracing a digital solution that promises enhanced efficiency, accuracy, and convenience.

4.1.2.2 Results of the Focus group

The main takeaway was the team's willingness to digital transformation. A recurring talking point was the importance of reinstating a part-time dedicated stores manager, as previously existed prior to the COVID pandemic, elaborating on how expenditure on part-time minimal wages in exchange for reliable stores-keeping would be a better use of resources than using mechanic's and engineer's time.

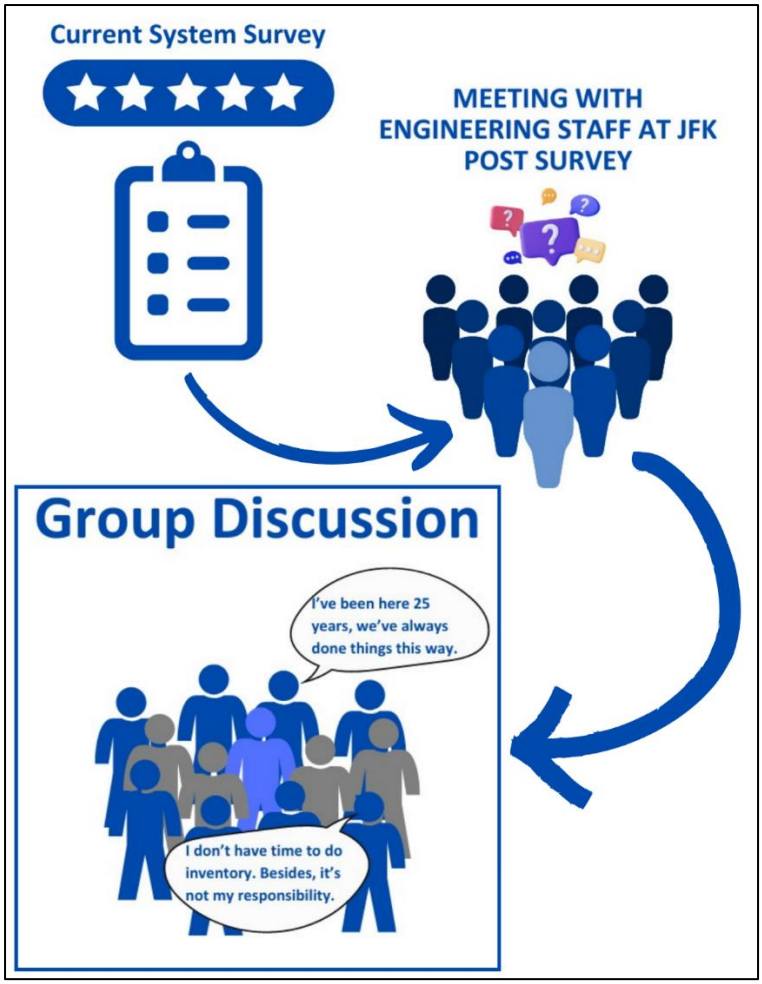


Figure 12 - Illustration of the Qualitative Data Collected Regarding Employee's View on the System at Time 0.

4.1.2.3 Results of targeted Semi-Structured Interviews

Two interviews were conducted, one with the unionized Maintenance Representative in charge of BA stores and the other with a non-unionized Station Maintenance Manager in charge of customer airline services.

The first interview brought to light the burden caused by the current inventory system, expressing frustration regarding misuse of resources, as time should be spent on aircraft maintenance rather than store keeping. When questioned about potential solutions envisioned, the Maintenance Representative cited reinstating a dedicated stores keeper as well as training all employees to complete SAP entries, removing the analog hand-written ledger intermediary and the bottleneck of one employee performing all data entry. In the absence of training for all employees to perform SAP data entries, the Representative mentioned the importance of enforcing compliance to the existing registry system, holding

employees that either failed to complete or inaccurately completed the hand-written book, accountable. The Representative was highly skeptical of a new solution, highlighting the prevalence of a systemic underlying issue, where employees disregarded the importance of storekeeping, independent of the method employed. Furthermore, the Representative questioned the usefulness of a new solution that would not automatically integrate with BA's SAP system, thus diminishing transparency for BA's and failing to address the SAP data entry bottleneck.

The Station Maintenance Manager interviewed had the opposite perspective, showing great expectation as to a potential digital tool. They highlighted the importance of control over essential consumable stocks such as oil and nitrogen, because of instances where the station has run out, having to borrow from other maintenance stations at JFK to continue operating. To overcome this issue, they suggested an automated stock monitoring system, where falling of stock quantity below a certain threshold would trigger an email to the Maintenance Managers in charge, suggesting that a purchase be made. Another issue brought out was the absence of a secure wi-fi internet connection in Hangar 10, rendering use of connected mobile applications difficult. They stated that fiber connectivity would be implemented by February 2024, at the cost of $50'000, in partnership with American Airlines. Lastly, the Maintenance Manager stated internal budgeting conflicts were delaying the delivery of company tablets to station engineers, forcing many employees to continue using their personal devices on a BYOD fashion. In turn, this has brought about a conflict within the team, since SIM cards and phone plans on personal devices are only paid by the company for non-unionized member with a management role, leaving all other employees to rely on their own phone plans. Awareness of these issues are crucial to building a usable application and to adequately presenting it to team, preempting and appropriately addressing conflictual topics.

4.2 Overhaul of the Stores Organization and Signage

At the incipience of the project, it was observed that the stores did not present a current hierarchical location system, signage was inconsistent and systems distinct between the Hangar 10 and Terminal 8 bonded stores. Storage of items, equipment, and tools at BA JFK stores lacked a standardized approach, leading to inefficiencies in locating and retrieving items. Thus, it was first necessary to overhaul the organization of the physical stores.

A hierarchical locating system at British Airways (BA) Engineering JFK was thus introduced at storage facilities at Hangar 10 (H10) and Terminal 8 (T8). The system defined the location by Store (H10 or T8) > Row (A to Z) > Rack (1 to 10) > Shelf (1 to 100) > Bin (1 to 1000), yielding unique identifiers such as H10 – C – 10 – 1 – 0, see Figure 15.A. Corresponding signage was then created and hung throughout the stores, over all Rows, Racks, Shelfs and on Bins, see Figure 15.B. Lastly, a Local Working Procedure (Annex 2) was created and submitted to approval by Quality, being subsequently published in the company's internal documentation repository, called DocuNet, as a mandatory read & sign for all BA Engineering employees at JFK.

Once the stores were reorganized and signs adequately placed, over 350 codes were created for each shelf or bin and attached at their referring location codes were thus printed, laminated, and placed (Figure 15.C)

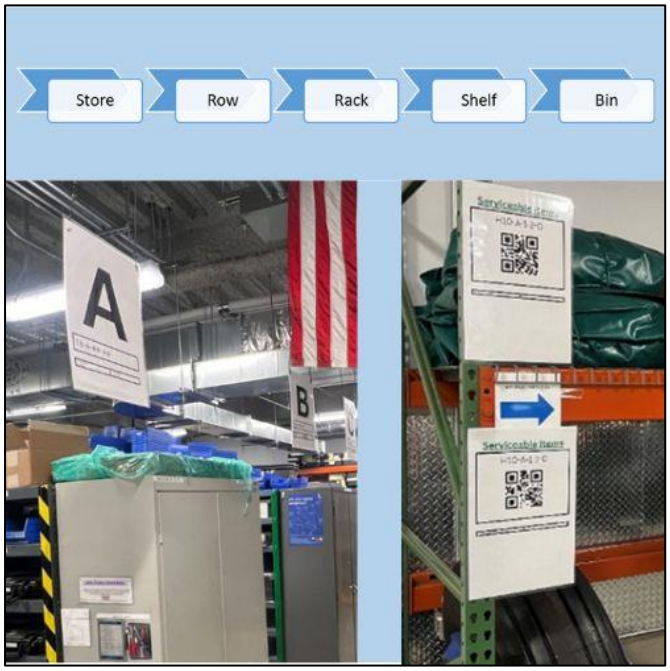


Figure 13 - Depiction of the Overhaul of the Store's Organization. Top panel: hierarchical location system implemented. Bottom right panel: hierarchical signage throughout stores facilitates navigation and retrieval. Bottom left panel: QR codes on each bin allow for integration with an app.

4.3 Description of the Mobile Application and Blockchain Implementation

The application harnesses BA's Microsoft (MS) network through MS Power Apps. There is a dedicated BA environment to roll out apps across the company.

4.3.1 Front-end

4.3.1.1 Rationale for the app's front-end

From the literature review and the analysis of the system in place, key messages appeared. Firstly, traceability emerged as a main priority in the front end, allowing all users to access the data before and after each transaction as well as search the database across airlines and fields. In doing so, the front-end would also become a single point of oversight for all users. In addition, traceability could be reinforced by implementation of a QR code signage system within the stores and QR code readers in the front-end. Secondly, increasing accountability was also a priority, transparently stating by which user changes in the databases were made. Thirdly, a user-friendly interface, with an intuitive feeling and a pleasant design, personalized to JFK's station, appeared essential for user satisfaction, potentially boosting adoption.

4.3.1.2 Description of the app's front end

To harness BA's secure MS environment, seamlessly integrate with existing systems and use procedures familiar to the team, the front-end was built using MS Power Apps low-code platform, assisted by its Artificial Intelligence Copilot function.

The structure of the app front-end follows the logic described in the subsequent Figure 16, a user guide (Annex 3), a technical guide (Annex 4) and the full code are also provided (Annex 5).

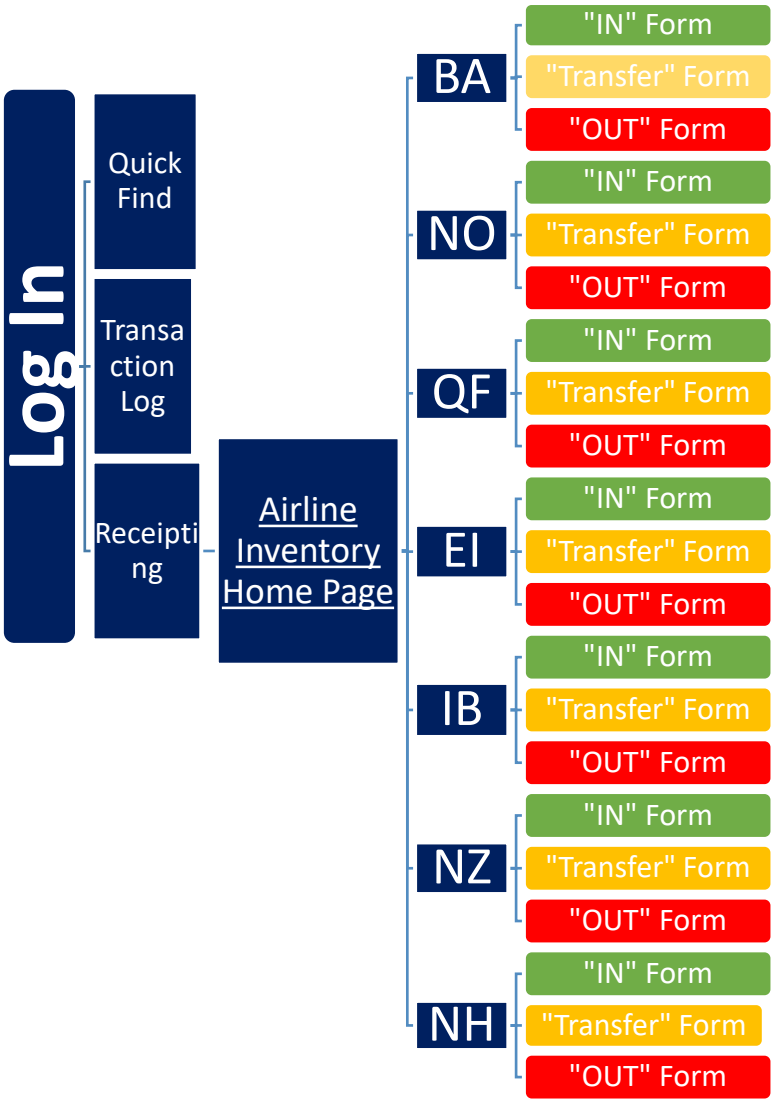


Figure 14 - Diagram of the App Front-End Pages, from Log In to Transaction Forms for Each Airline. Client Airlines Include Norse (NO), Qantas (QF), Aer Lingus (EI). Iberia (IB), Air New Zealand (NZ) and All Nippon Airlines (NH).

4.3.2 Back-end

4.3.2.1 Rationale for the app's backend

Research and surveys highlighted the importance of (1) guaranteeing cyber-security, especially in a BYOD context, (2) building upon existing systems to match end-users' digital literacy skills and ease the transition. Using the MS BA environment SharePoint lists emerged as the logical solution, since their functioning is very similar to the Excel Spreadsheets employed by engineers to keep an internal record of each customer's airlines stores. The choice to move from Excel to SharePoint was due to the storage capacity of the latter being much larger and processing faster than with the former (Guzowski, 2019).

4.3.2.2 Description of the app's backend

The backend relies on individual airlines' SharePoint lists, that are updated as transactions go through, see Figure 17. The lists are still directly accessible on the MS BA environment, but now in a read-only format for everyone except the engineer in charge of each airline's stores keeping. All columns and formats are harmonized across airlines SharePoint lists. Rules are set so that fields left blank when manually adding an entry to the SharePoint lists are automatically set to N/A or 01/01/2099 when it comes to Expiry Date.

In addition, the back end includes a master SharePoint list of all possible locations for each airline, that is consulted by all Location fields across IN, OUT and TRANSFER pages in the front-end, to ensure a part is not input, output, or transferred to or from the wrong airline's inventory. Importantly, transfers of parts between airlines are not directly allowed by the application since it requires specific authorization from both airlines, unless part of a pool agreement.

Lastly, it also includes a SharePoint list for the transaction log. Any new entry in the transaction log triggers an automated email comprising the entry's content in the body of the message, sent to the researcher. The transaction Log is available to view to all members of the JFK BA team, within the MS BA environment, only the researcher who created it has editing capabilities.

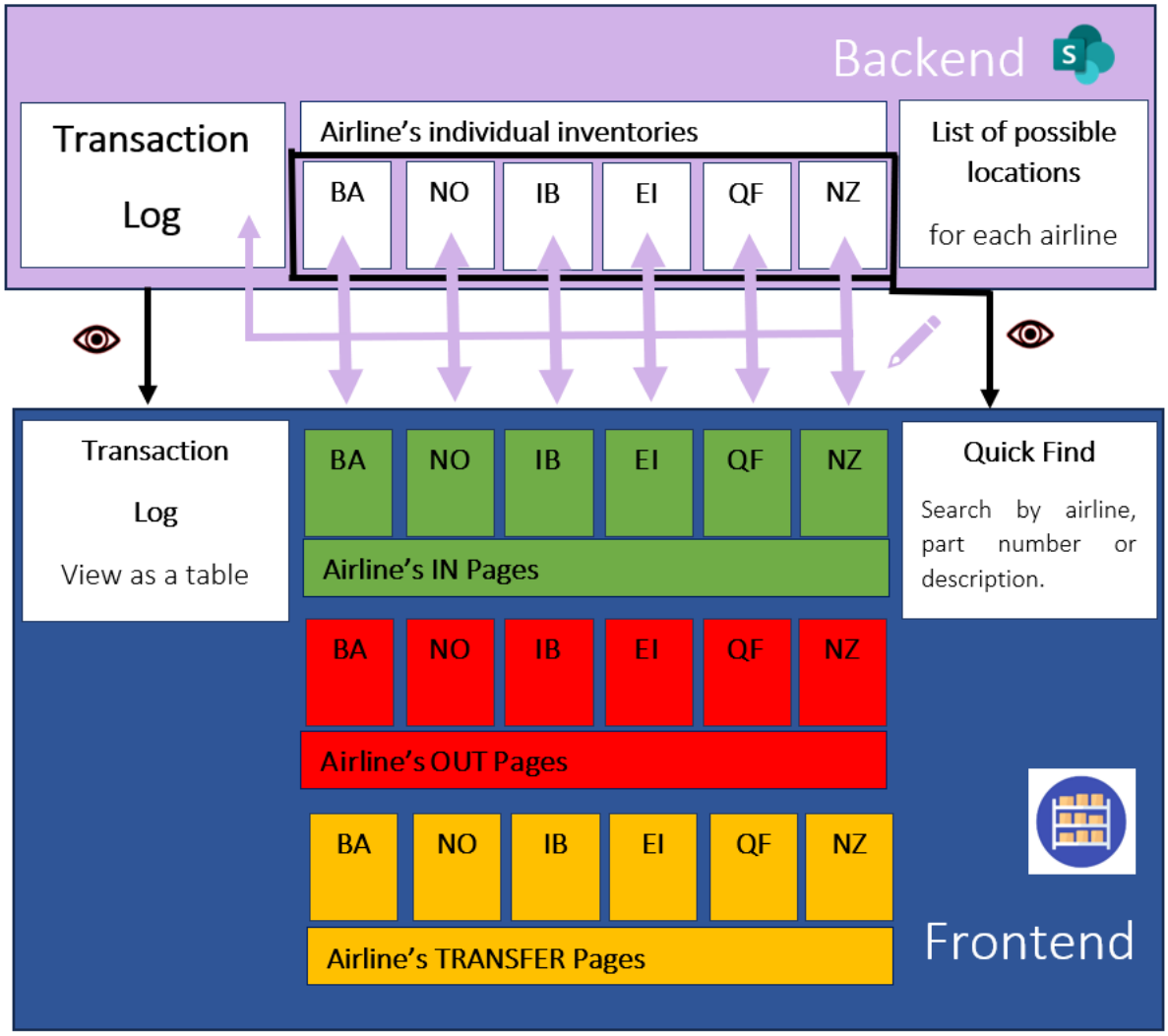


Figure 15 - Diagram of the Mobile Application's Frontend (Blue Panel) and Backend (Purple Panel). Black arrows indicate view-only information, while purple arrows indicate information can be edited. Airlines are represented by their tail numbers, see figure 16 for a list of abbreviations.

4.3.1 Interface with the blockchain

4.3.1.1 Rationale behind the blockchain interface

According to the literature search, blockchain is best harnessed for secure inventory management by (1) creating a digital twin of records on a distributed ledger and (2) tokenizing items to be transacted and (3) specifying transaction mechanisms and rules in a carefully crafted smart-contract based on a buyer-seller relationship. In this project, each item was tokenized by associating it with a unique identifier, created by sequentially combining the strings that make each record distinct in the application's back-end databases. Transactions of these tokenized assets are made between each airline's catalog, the distributed-ledger

twin of the airline's SharePoint list in the application's back end, and an additional catalog that represents all the application's users and is the mirror image of the log. A smart contract is deployed, restricting which airlines the universal user is allowed to transact with, which specifications regarding the part are mandatory for a transaction to go through and how quantities and entries in the ledger are updated for each type of transaction.

Because of security concerns, no low-cost option was available to interact with BA's MS environment. A rest API was envisioned, and a template coded, but its implementation was ultimately limited by paywalls within MS Azure. As such, to allow the prototype to go forward, full integration between the application and the interface with the blockchain was not achieved at the project's conclusion.

4.3.1.2 Wallet Creation and Management

For each airline, including British Airways, as well as for a unique virtual user called Stores Keeper, a unique digital wallet is created on MetaMask, a popular Ethereum wallet platform. Each wallet is secured with a private key, which is critical for authorizing transactions (Figure 18). The security of these keys is paramount and is typically managed by a small group of authorized personnel within each airline's inventory management team.

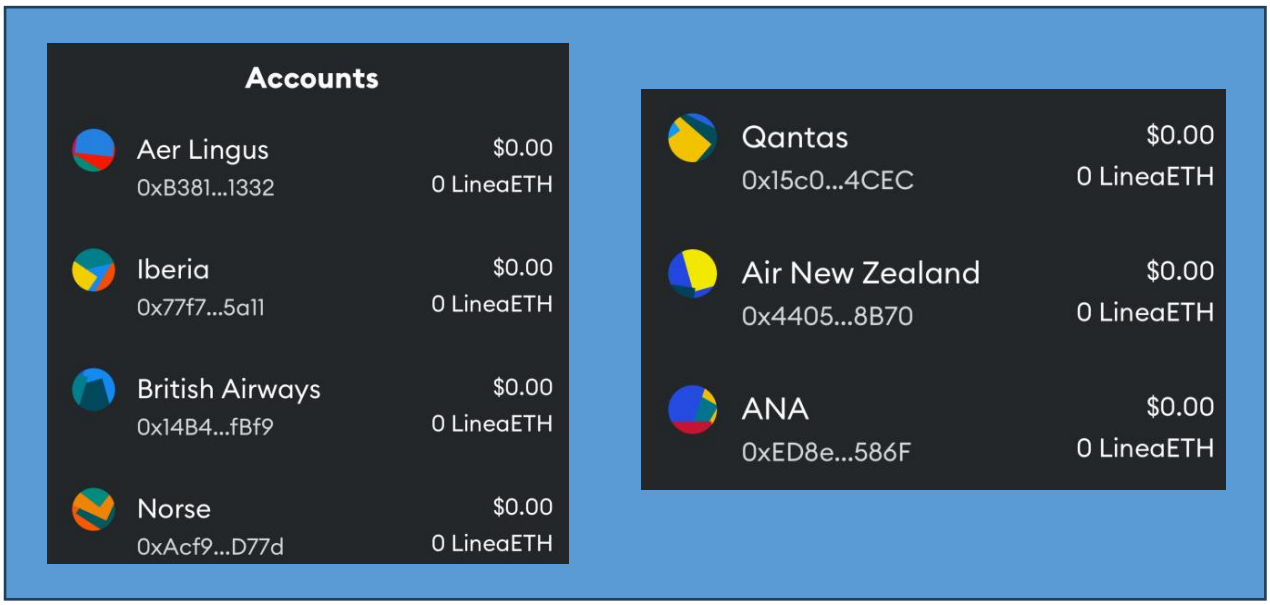


Figure 16 - MetaMask Wallets for Each Airline.

Users of the mobile application are not attributed to a unique individual digital wallet since such structure would require automated mechanisms for wallet creation. This would engage third party web services and be associated with security risks, potentially becoming a gateway for attacks to BA's MS environment.

Each blockchain transaction has its twin in the log file, a SharePoint list containing the identity and email of the individual responsible for the transaction. Thus, authorized BA personnel, having access to both the SharePoint log file and the blockchain ledger, can easily trace back from each transaction's unique identifier who initially elicited it via the application front-end. Accountability is ensured without compromising security.

4.3.1.3 Smart Contract Integration

The smart contract's architecture is delineated in the diagram below. The full Solidity code and its explanation are available in Annex 6.

The smart contract is coded in Solidity, tested, compiled, and deployed using Remix IDE, on the Sepolia Testnet under ownership of the store's wallet. It includes functions to authorize a new airline's digital wallets for transactions, to receipt items in and out, as well as to consult items present in the distributed ledger.

4.3.1.4 Inventory Tokenization

Each part in the inventory is represented as a digital token on the blockchain with a unique identifier. These tokens carry as metadata part unique identifier, part number, part description, serial number, batch number, expiry date, location, and quantity.

When a new part is added to the inventory, a corresponding token is created and assigned to the catalog associated with the airline's wallet address. This token can then be tracked across the blockchain, providing a transparent history of its movement and usage.

4.3.1.5 Transaction Process on a Custom Decentralized Application (dApp)

Transactions are run using a proprietary script coded in Python 4.0, using the Web3.py library, constituting a dApp (Annex 7).

A node in the blockchain net is connected to using a http link provided by Infura. The script automates reading airline's SharePoint lists and SharePoint log file line by line and performing the corresponding transactions as defined by the smart contract. For every transaction authorized, the information for the part transacted is retrieved and saved in a list. Likewise, every transaction hash is saved in a separate list. In addition, the script includes timers to prevent delays in transactions to disrupt the sequences of transactions to be performed, which could lead to errors in records, for instance causing an attempt at outing a part before the inputting of the part is complete. This feature is of the essence given transactions can take variable times, depending on the demand placed on the net when the transaction is signed. See Figure 19 for a full diagram.

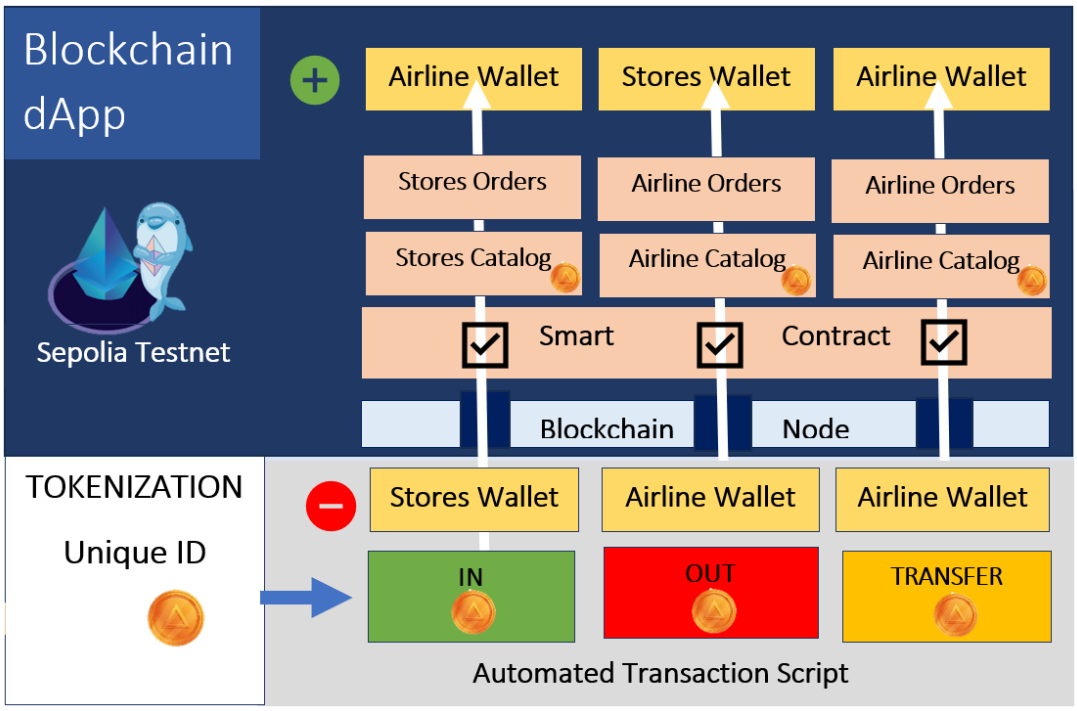


Figure 17 - Diagram of the Decentralized Application.

In the first initiation of the script, or any time a new customer joins the JFK's BA Engineering client roster, the airline's catalogs must be saved in the blockchain. Once the catalogs are set up, transactions can be performed. For this, there is not, at this stage, a direct integration with the mobile application. The log file must be uploaded in the script's environment. For this study, the log file was manually uploaded once at the conclusion of the experiment conducted to verify the usability of the mobile application (see methods 4.5 and results 5.5). The file is then read by the script, and each line triggers a specific transaction, thus updating the airlines' catalogs on the blockchain. For each of these transactions, Ether is debited and credited to the buyer's and seller's wallets, respectively. See Figure 20 for a diagram of the complete solution developed, including mobile application and decentralized application .

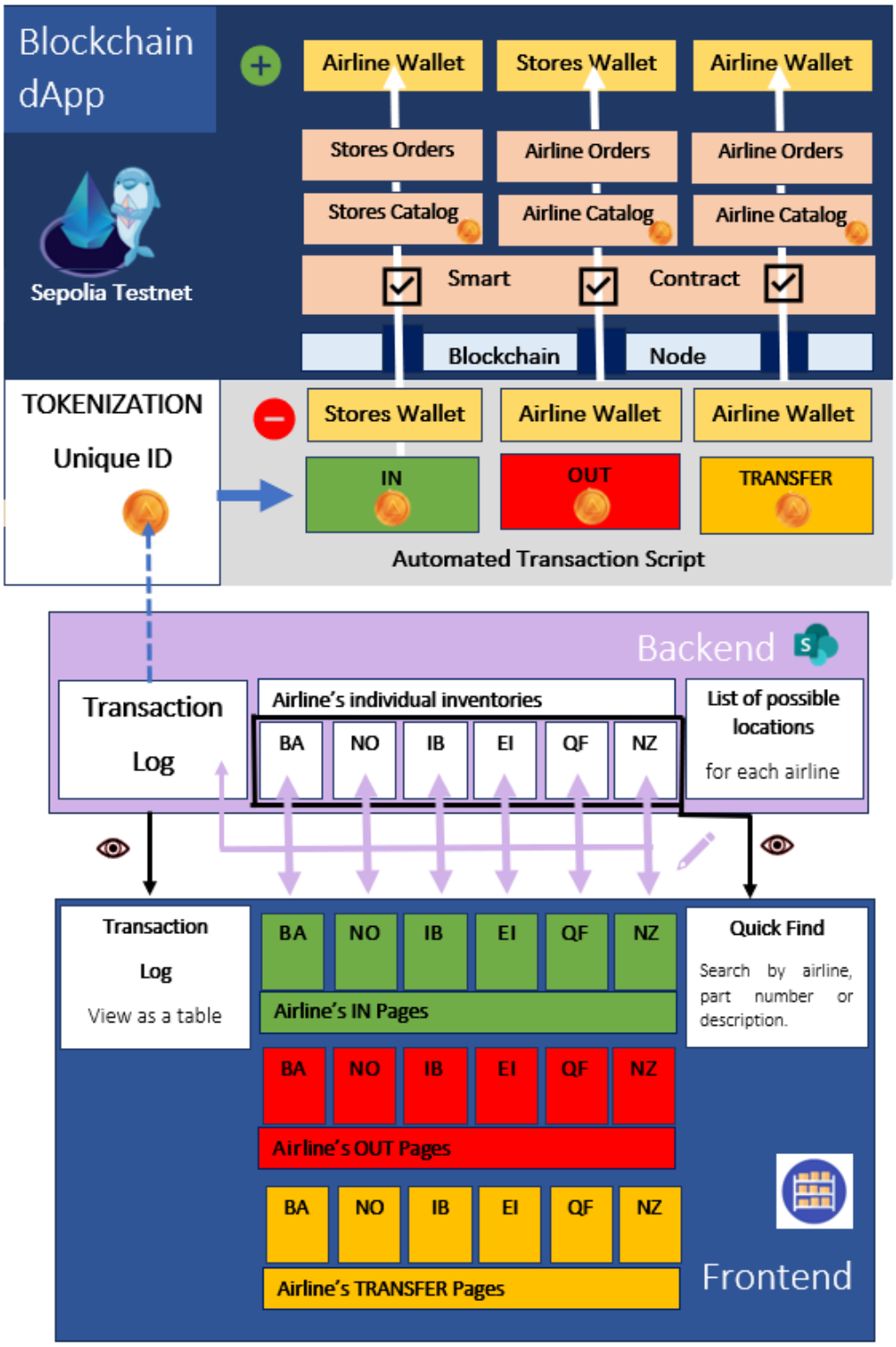


Figure 18 - Diagram of the Stores Keeper Solution. Includes a front end (Stores Keeper Mobile App), a back end (MS Share Point Lists) and a decentralized application coded in Python and running on the blockchain (Sepolia Testnet).

4.4 Evaluating the Usability of the Mobile Application

4.4.1 Materials Created to Introduce the Application to Users

Once the application was ready for deployment, it was introduced to users. A complete user guide was provided (Annex 4). Video demos were recorded (screenshots in Annex 7) and shared on relevant company communication channels to users. A technical guide (Annex 5) was also available for consultation. One-to-one training was additionally available upon request.

4.4.2 Systematic testing of the applications

A structured test of the application was conducted, recording the screens of the mobile devices used. Statistical analysis of anonymized data was performed using IBM SPSS Statistics for Windows.

Eleven (13) users completed the test, performing a total of 18 transactions each. Distribution of subjects is described in the table below, according to relevant categories for the analysis. One subject was rejected from the analysis, due to failure of the screen recording. Full results are available in Annex 8.

Table 1 - Distribution of Employees Having Performed the Experiment to Test the Mobile Application's Prototype.

Category	Mechanics (Unionized)	Maintenance Reps (Unionized)	Managers (DMM/SMM/AMM – Non Union)
N of Subjects	4	1	7

4.4.2.1 Accuracy of Transactions

The primary outcome considered was the accuracy of the completed transactions. The log entries in the back end of the mobile application to verify if transactions were correctly performed either the mobile application, rating them 1 if correct and 0 if incorrect. Similarly, the entries in the analog book were photographed and read by the researcher, giving equivalent rating. When handwriting was difficult to make out, creating ambiguity as to whether entry was correct or wrong, a 0.5 rating was given. Scores were summed for each transaction type, yielding a total of *min* = 0 to *max* = 3 for each of the 4 transaction types.

The mobile application (m=1.00 , $s.d.$=0.00) was significantly ($F(1, 33)$= 12.50, p=0.00, *p. eta sq.* =.28) less prone to errors than the analog system (m=.85, $s.d.$=.04) . Furthermore, the app was especially beneficial ($F(2, 33)$= 4.8, p=0.01, *p. eta sq.* =.23) for mechanics, more so than for maintenance reps and managers (see Figure 21). Interestingly, such divide bridged across the union v. non-union separation of roles. Because only 1 subject was a maintenance rep, we repeated the analysis by adding said subject to the Mechanics group. With this, the

interaction between method used and role, confounded with the effect of union status, became marginal ($F(1, 34)= 3.43$, $p=0.07$, *p. eta sq.* =.09). No other within-subject's factors, between-subjects factors or interactions reached significance in either analysis (all p's > 0.4).

Accuracy of Transactions with the Analog Method v. Mobile App.

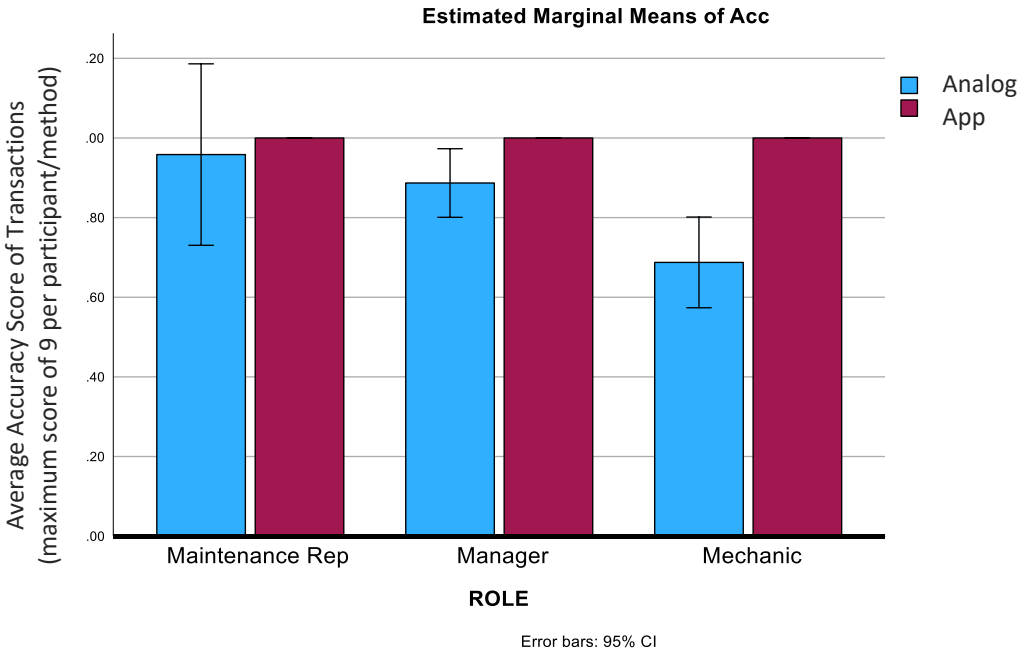


Figure 19 - Accuracy Score of Transactions by Employee Category for the Analog Method as Used at Time 0 (Blue) and the Mobile Application's Prototype (Magenta).

4.4.2.2 Time taken to Complete Transactions

The secondary outcome considered was the time taken to complete each transaction. Screen recordings were analyzed by the researcher to extract the duration of the transactions, in both the app and the analog system. Here as well, the within-subjects factor of analog (*m=64.95 , s.d.=2.24*) v. mobile app (*m=43.92 , s.d.= 1.17*) reached significance ($F(33 ,1) =75.30$, $p=0.00$, eta sq. =0.70 . There was an interaction with analog v. mobile app ($F(33 ,1)=17.08$, $p=0.00$, *p. eta sq.* =0.34), showing that the app increased the speed of transactions

for “IN”, “OUT” and “Quick Find” (*all p’s*=0.00), but the not for “Transfer” (*p*=32). In addition, a main effect of transaction type appeared (*F*(33 ,1)=52.48 , *p*=0.00, *p. eta sq.* =0.61), with Quick Find transactions being significantly faster (*p*=0.00) than the 3 others (by *m*= 17 ms - 24 ms, *s.d.*=1.58 - 2.30). No other within or between factors reached significance (all *p*’s>0.53). This pattern of results attests to a time-saving capability of the app across transaction types and employees, with an increased benefit for certain types of stores transactions.

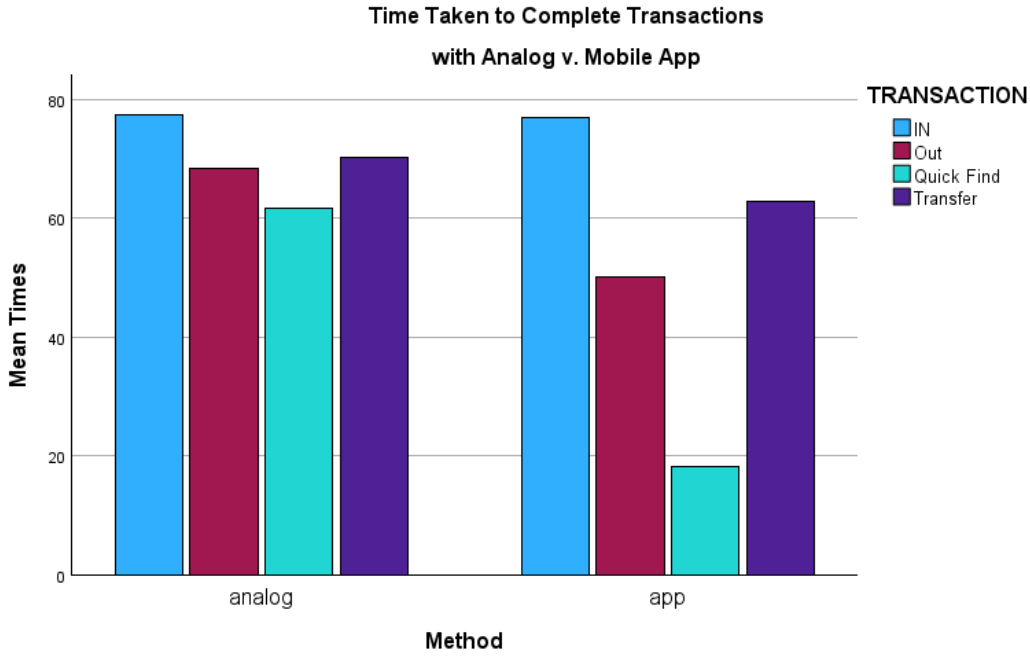


Figure 20 - Box Plot of the Mean Time Taken to Complete a Transaction by Transaction Type, Using the Analog System as Implemented at Time 0 or the Mobile Application Developed.

4.4.2.1 Observed Difficulties Using the Mobile Application

The tertiary outcome considered were patterns of difficulties using the mobile application. Screen recordings were viewed and subjectively analyzed by the researcher. Some common issues were internet connectivity problems, failure to understand the logic of the quick find page and the part location not being typable, having to be input only by scanning a QR code. Of note, these difficulties did not result in terminal errors since all participants completed the 18 transactions required without assistance. See Annex 8 for screenshots of the experiment and a table of results.

4.4.3 Systematic testing of the Blockchain component

4.4.3.1 Simulations

The full simulation script is provided in Annex 9.

For all simulations tested, the blockchain component to transact on the blockchain scored 1, reaching a perfect score of 5. In other words, the blockchain implementation was successful in identifying errors and limiting transactions accordingly for the 3 scenarios listed below:

- Retrieving a wrongly identified part from storage.
- Retrieving a part in a stated location that does not match its physical location.
- Creating a duplicate record when parts with identical identifiers are added to stores, as opposed to increasing quantity of said part in stores.

A Tamper-proof twin of records was thus maintained in the testnet's distributed ledger. However, more rules were implemented in the mobile application's front end than in the Smart Contract, the latter being thus more permissive. This is because both methods were designed to be used in conjunction, the smart contract transacting only from the SharePoint log, as updated by transactions on the mobile application's frontend. A balance was achieved between the smart contract functionality and its complexity.

4.4.3.2 Daily Transactions from the Log to the Blockchain Component

In addition, for the 3 weeks that the application was deployed, new entries in the log were input to the blockchain transacting script once. Because there were 3 or less transactions per day in the log, this frequency was sufficient for the blockchain to detect potential errors. However, no errors occurred in the test period. The average cost of the transactions was $ 0.03 and the time taken to complete each transaction varying from *min* = 0.4 seconds to *max* = 45 seconds, depending on the testnet's demands at the time of each transaction.

It is of note, however, that the blockchain implementation was not fully integrated with the mobile application, requiring the log SharePoint list to be downloaded and placed in the blokchain component's environment. Likewise, there were no automated mechanisms to notify users of the mobile application when transactions on the blokchain component failed.

4.4.4 Evaluating the impact of the new solution

4.4.4.1 Qualitative Survey of the Experiment's Subject and Focus Group

After having completed the experiment, all 20 subjects filled a survey comprising 5 items. To the question "Did you find what you were looking for?", 100% of responses were "Yes". The transaction experience, an open question received responses that ranged from "Great" to "Excellent". All subjects responded "Yes" to "Did it Save you Time?" and all ratings, from 0 to 10, were 10. No suggestions for improvement were received.

However, of note, this survey was taken in presence of the researcher and with users logged in to the mobile app, potentially biasing answers. Thus, a different and anonymous survey was conducted via MS Forms at the conclusion of the project. Responses collected were n=19. Overall app rating on a scale of 0 to 10 painted a mitigated picture (*min*=7, *max*=10, *mean*=8, *s.d.*=0.25). Responders did not consistently perceive the app as capable of preventing parts from being misplaced, only 70% responded positively. Furthermore, only 65% of responders agreed that the application saved time when transacting parts. All responders cited Quick Find as being the most useful feature of the application. Interestingly, only 50% of responders

mentioned having used the application on their own at least once. For these responders only ratings were higher than in all responses (*min*=9, *max*=10, *mean*=10, *s.d.*=0.5), and 100% viewed the app as being effective to both prevent errors and save time. In turn, for the same 50% of responses, the most useful feature was, as for the larger cohort, the Quick Find.

4.4.4.2 Spot-Check Results

A first spot-check before the onset of this project (time 0) found 40 parts mislocated and additionally 100 serviceable parts of which the location was absent in records. The value of these parts ranged from *min*=$50 to *max*=$10'000.00, with an average of *m*=$1'000.00 and *s.d*.=$750, with 75% of the misplaced parts having a value between $1'000 to $ 2'000. The total potential losses estimated were $ 200'500.

A spot check conducted 3 weeks after the mobile application was deployed (Time 1), across BA stores and all other customer airline stores found 20 parts mislocated, an ~~85~~% reduction in parts misplaced when compared to Time 0. In addition, revising of all SharePoint lists did not reveal any entries timestamped at after the deployment of the mobile application and conducted via the mobile application, since said transactions were also present in the log SharePoint list, that had errors in their entries in either format or content. The value of these parts ranged from *min*=$50 to *max*=$5'000.00, with an average of *m*=$1'000.00 and *s.d.*=$25, with 75% of the misplaced parts having a value between $1'000 to $ 2'000. The total potential losses estimated were $ 48'000, representing a %76 reduction from T0 to T1. See a breakdown by part value in Figure 21.

50%

SharePoint list may have errors but none came from mobile app.

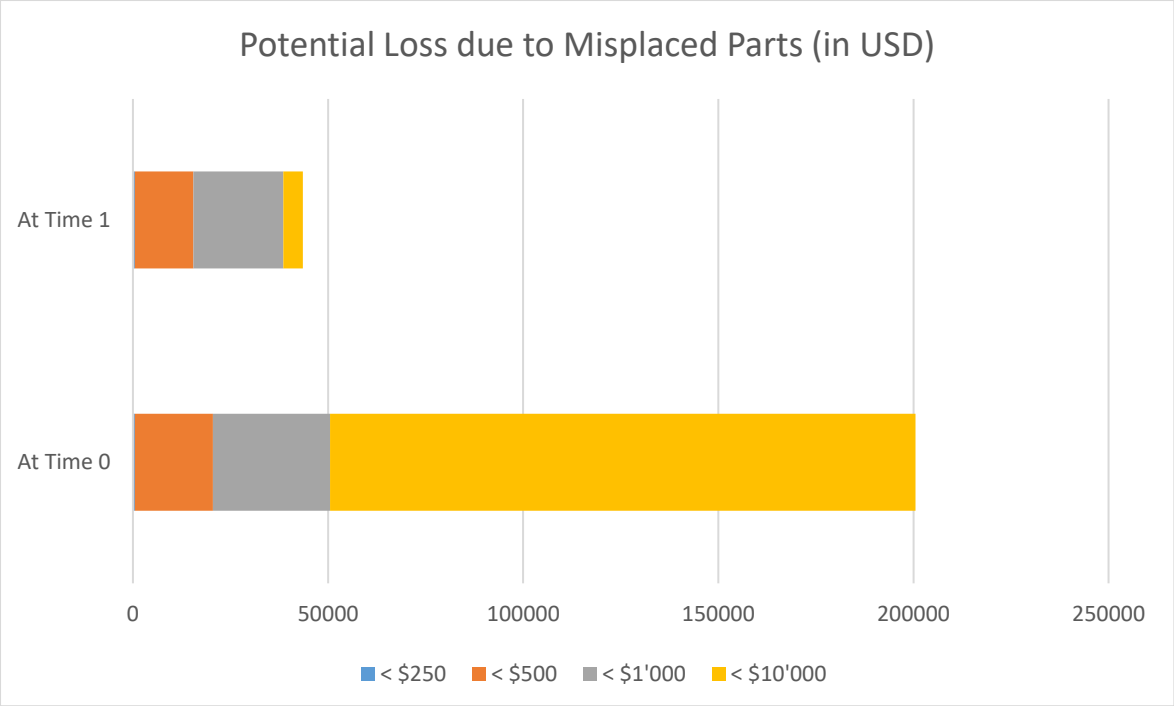


Figure 21 - Bar Graph of Potential Losses Due to Misplaced Parts at Time 0 and Time 1. Summed costs attributed to misplaced parts are color-coded by part value.

4.4.4.3 Markers of Spontaneous Adoption by Personnel

User acquisition was determined from entries in the log that did not stem from the experiment conducted (detailed in methods 4.4) during the first 3 weeks of deployment. In total 3 engineer users were acquired, each conducting one or more "IN" transactions for the airline's inventory that was under their responsibility. Of note, all the users acquired started using the application spontaneously after taking part in the experiment (see methods 4.4 and results 5.3).

In contrast, by analyzing the time stamps of modifications made on the SharePoint lists in comparison to the timestamps in the Log, modifications that were made directly on the back end, as opposed to through the app's front-end, were detected. In total, during the 3 weeks of deployment 2 duty maintenance managers systematically used the back end, rather than the front-end, to modify inventory records. These 2 duty maintenance managers were not part of the ones that adopted the mobile application.

4.4.4.4 Feedback from a Conclusion Meeting

Lastly, a focus group was held at the project's conclusion, as an online Teams meeting, with 15 employees in attendance. Overall feedback over the whole project was positive, from the stores location's reorganization and relabeling to the adoption of the mobile application as

well as the channel of communications opened, and the conversations sparked. Subjects brought out the importance of keeping such communication channels open, especially between union and non-union members. When presented with the results regarding prevention of errors, time saved and potential financial losses prevented, the group expressed what was best summarized as being “shocked but not surprised”. Such group sentiment was different from the average individual responses to the conclusion survey (see results 5.3.1.2) where only 65%-70% agreed with these statements. A presentation of the blockchain transaction portion of the project, never mentioned before, did not trigger any further comments or suggestions. It was further discussed how few users had adopted the tool since its deployment despite its advantages and how it should be formalized as a mandatory working procedure.

5 Discussion

Our findings are discussed in the context of their significance to our research questions, providing a clear picture of the blockchain-based system's effectiveness in revolutionizing inventory management processes.

The project's primary aim was to tokenize spares in a distributed ledger to improve security and efficiency. To achieve this, ~~it~~ I had to first overhaul 2 stores physical organization, place over 300 signs, homogenize 7 inventory databases and create a complex mobile application, all the while keeping all stakeholders satisfied. Only after these steps were complete, was centralized and coherent data available for a blockchain implementation.

Reorganizing the stores with hierarchical unique identifiers, presented in text and QR codes decreased time to retrieve and locate items (Morales-Hernández et al., 2021). The mobile application developed, called Stores Keeper, is a versatile app that can easily scale to fit any airline's needs. A similar app has been recently made available by AMOS , 2024, however it cannot be personalized and ~~does~~ comprise a blockchain component. Extensive testing in quantitative and qualitative terms, using methods similar to Ubald et al., 2022 and Yankah et al., 2024, demonstrated Stores Keeper accuracy-boosting, timesaving, and cost-saving potential, even only after 3 weeks of deployment.

The Stores Keeper blockchain component, albeit not fully integrated with the mobile application, was successful at detecting errors in stores storekeeping during simulations. In addition, it proved to be effective in the 3 weeks of its deployment, creating and updating daily ~~an~~ AND immutable twin of the Stores Keeper back end. The existence of such an immutable record adds a layer of security that physical books, spreadsheets in virtual clouds or SAP-like software's cannot provide (Henry and Kathawate, 2023; Oksuz, 2024; Sudarssan, 2024). In an industry with high employee turnover, low investment in stores management and yet very high safety stakes and costs associated with spares, achieving the utmost security standards in storekeeping is crucial (FAA, 2019).

Furthermore, the blockchain portion of the project, as it was conceived, overcame limitations described in literature. Firstly, one of the lessons of the FlightChain project were that scaling such a network to an ever-growing number of participants , as new airlines were onboarded required extensive work from developers and thus rendered the whole project difficult to justify (SITA, 2017). Here, thanks to the smart contract and the blokchain component created, little time or technical knowledge is required when adding a new airline to the roster, other than setting up a MetaMask wallet and providing the airline's inventory spreadsheet to the blokchain component. Secondly, because transactions through the blockchain and through the Stores Keeper app were not synced in time, delays caused by high demands on the blockchain net did not affect users of the Stores Keeper app, an essential contention point in an industry where time is of the essence (see limitations in (Ho et al., 2021)). This dissociation between the timelines of the Stores Keeper and the blockchain script was not problematic for accuracy of inventory keeping since rules in the smart contract were redundant with those in the Stores Keeper front end. Furthermore, since most rules were implemented in the Stores

Keeper front end the complexity of the smart contract was alleviated, avoiding errors. Thirdly, the solutions was cost-free, as it was deployed on a testnet, which is a blockchain designed to mimic the operating environment of a "mainnet" but is cost-less since it runs on a completely separated ledger (Axen, 2022). Beyond the current prototype stage, deployment in the Ethereum "mainnet" would ensure the record's perennation, with costs remaining under $10 / year at the current rate of use (Tambe, 2024).

Importantly, because the Stores-Keeper application was fully created and hosted within the company's pre-existing cloud environment, cyber-security automatically conformed to company standards, as confirmed by consultations with relevant departments. However, because the deployment of a mobile application enticed a BYOD practice, it is recommended that cyber hygiene training and communications to raise awareness are put in place as the solution is rolled out (Ukwandu et al., 2022). Regarding the blockchain transacting portion of the solution, information stored in the blockchain is made secret through hashes, a form of cryptography (Peng, 2021). In this implementation, a security vulnerability exists in the blokchain component, in the manual step of uploading data from the company's cloud environment to the transaction script.

To address such vulnerability, it is ideally envisioned that the blokchain component (D'Oro et al., 2022) is moved to the airline's cloud environment and integrated with the mobile application, via a restAPI (Massé and Massé, 2012) and custom connector (Jopanchal, 2023). Because such system would require additional development cost, for instance with a paid subscription to a platform such as MS Azure (Divyaswarnkar, 2022), prototyping to thoroughly assess the efficacy and the real benefits of tokenizing spares on the blockchain is essential before incurring additional costs. In addition to integrating blockchain and mobile application, an automated parallel path can be imagined for the integration of the mobile application and the SAP software used throughout the company. In fact, according to a consultation with the company's IT department, when the most recent SAP system update is completed, third-party integration will be allowed, and SAP GUI scripting (SAP, 2019) could be used for the Stores Keeper application to transact directly with the BA SAP system. Here, as for direct integration with a blockchain system, thorough proofing of the prototype is recommended before establishing data-altering automation.

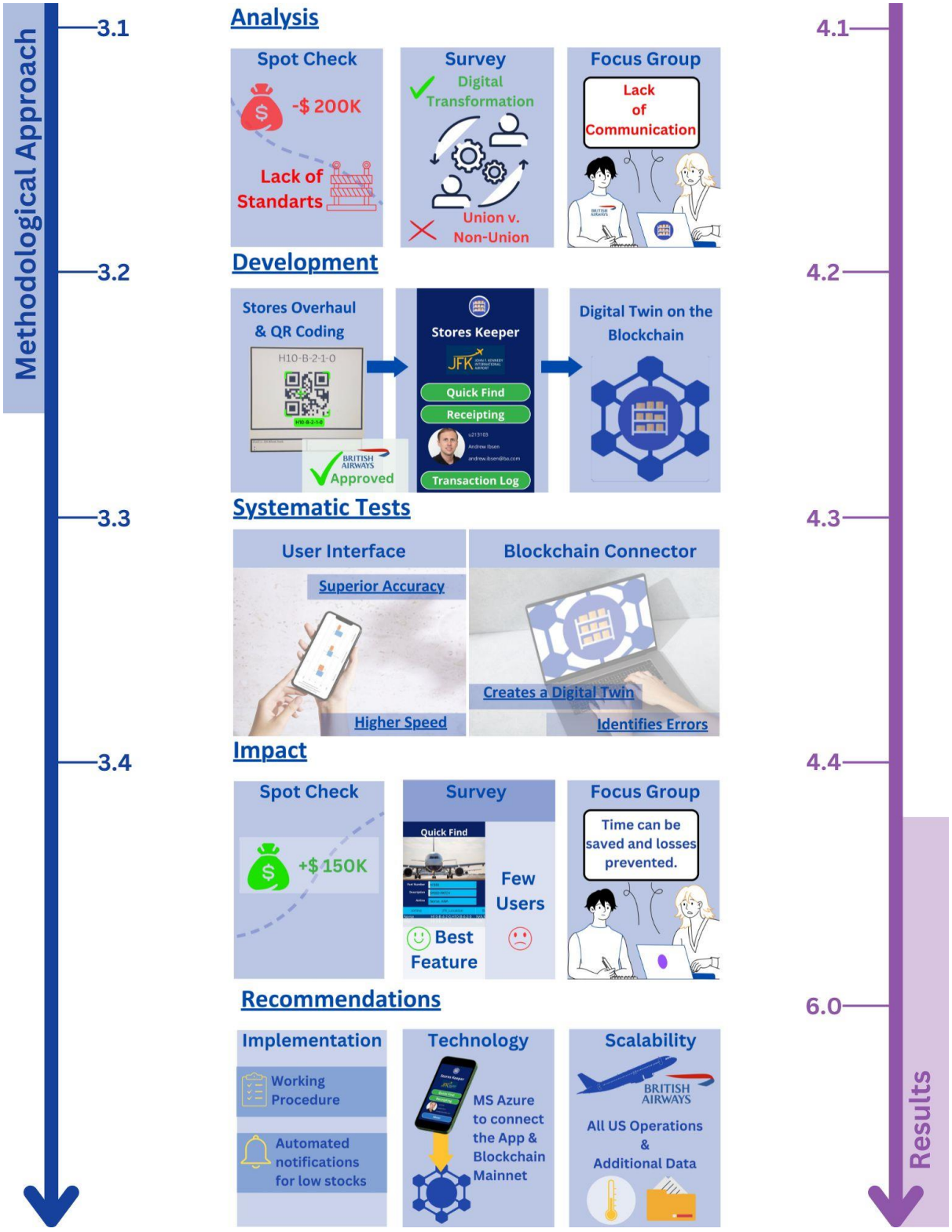
Overall, the project has successfully implemented and tested a tokenization of parts for inventory management in line maintenance. It has done so by harnessing publicly available tools, exploiting complex technology with little programming and software development expertise, at minute cost and in compliance with cyber-security standards. Thus, the much-discussed trend of blockchain for inventory management in aviation was put to test. Its value was proven, provided it is built upon real-life, bottom-up meticulous organization of stores and their information systems, tailored to the operation's capabilities and personnel.

6 Conclusion

6.1 Achieving the study's objectives

The results outlined offer a compelling narrative on the transformative potential of blockchain technology in aviation maintenance. They
provide a roadmap for future enhancements and broader implications for the aviation industry. All 4 main objectives were met as listed and illustrated below, in Figure 24 .

1. Main strengths and limitations of spares management at BA's JFK station at the onset of the project included:
    a. A spot-check detected $200K in misplaced parts, while a systematic analysis of the system found a lack of standards in inventory-keeping.
    b. A survey of users uncovered a rift between unionized and non-unionized employees as to best practices in inventory-keeping, but an overall willingness of all for digital transformation.
    c. A focus group and semi-structured interviews with key stakeholders brought out a lack of effective communication regarding stores.
2. Building of a user-friendly mobile application to tokenize spares comprised:
    a. A hierarchical location system with QR codes was implemented in the physical stores.
    b. A QR-scanning MS Power inventory management mobile application, called Stores Keeper App, was built.
    c. A decentralized application that interacts with a blockchain, tokenizing spares, was developed. Together with the mobile application, it constitutes the Stores Keeper Solution.
3. Test the viability of tokenizing spare parts in BA's hangars included:
    a. An experiment with the mobile application found it to improve accuracy and time taken to complete transactions in and out of stores.
    b. A set of simulations with the decentralized application found it to be success in preventing errors in stock keeping and in keeping an immutable digital twin of stores record.
4. Qualify and quantify the impact of the proposed solution to tokenize spare parts.
    a. A new spot-check found 50K in misplaced parts, or a 150K cost-saving relative to the spot-check at the onset of the project (1.a).
    b. A survey of users rated the application positively overall, despite few new users having been acquired.
    c. A focus group and targeted interviews highlighted the potential time and cost saved thanks to the solution.


Methodological Approach
3.1
Analysis
Spot Check
-$ 200K
Lack of Standarts
Survey
Digital Transformation
Union v. Non-Union
Focus Group
Lack of Communication
3.2
Development
Stores Overhaul & QR Coding
H10-B-2-1-0
Approved
Stores Keeper
JFK
Quick Find
Receipting
Transaction Log
Digital Twin on the Blockchain
3.3
Systematic Tests
User Interface
Superior Accuracy
Higher Speed
Blockchain Connector
Creates a Digital Twin
Identifies Errors
3.4
Impact
Spot Check
+$ 150K
Survey
Quick Find
Few Users
Best Feature
Focus Group
Time can be saved and losses prevented.
Recommendations
Implementation
Working Procedure
Automated notifications for low stocks
Technology
MS Azure to connect the App & Blockchain Mainnet
Scalability
BRITISH AIRWAYS
All US Operations & Additional Data
4.1
4.2
4.3
4.4
6.0
Results

6.2 Additional take-aways

In the dynamic environment of line maintenance, having centralized access to part availability and the capabilities, in both hardware and software, to transact components quickly and efficiently encourages the user to not cut corners – the path of least resistance. Here, implementing complex technology such as a distributed ledger proves to be effective only when stores are re-organized from the bottom-up. Furthermore, the Stores Keeper creation process highlights the importance of tailor-made solutions, based on in-depth analysis of the systems and information flows in place. In addition, involving future users from the project's onset proved to be extremely valuable for insight, feedback and, ultimately, adoption of the tool created. Since efficient technology is available in low-code and AI assisted platforms such as MS Power Apps, effort can move away from programming challenges, instead going further into refining understanding of needs, translating into better design. Lastly, even efficient, and well though-out solutions do not exist in a void, and must be integrated with larger, possibly outdated, systems. Thus, it is of the essence to also work closely with IT departments, to create customized and secure integration solutions as legacy systems are gradually upgraded.

Next steps to further enhance the Stores Keeper solution include deploying the smart contract on a "mainnet" ($70.00) and running blockchain transactions on it (projected $10.00/year) as well as integrating the mobile and decentralized applications creating using a custom connector on MS Azure (estimated at $ 100.00/year). It also includes integrating the system to SAP records using GUI Scripting once the SAP system is updated, in April 2024. Furthermore, scaling to all stations in the US is envisioned and supported by BA. With this, it would be conceivable to move away from external blockchain and implement a proprietary blockchain, distributed across BA stations. Another potential add-on currently being considered is using computer vision tools to digitize part's documentation when transacting them into the stores. This would prevent storage of parts without proper document checks (see diagram in 4.1.1) and allow for more comprehensive audits to be conducted. In the same spirit, a temperature monitoring system is being installed in JFK stores, using a miniature Raspberry Pi computer ($ 70), with temperature records stored in a blockchain. This allows for the retrieval of continuous and tamper-proof graphs of the store's conditions for audits and, most importantly, to trigger immediate email alerts to personnel in charge when regulations are breached. Successful implementation in the scope of this project has paved the way to building custom application, bringing BA stores to the forefront of technology and, most importantly, making much-needed strides in efficiency and safety.

6.3 Avenues for research

Thanks to a system such as the Stores Keeper, data is systematically collected and made tamper-proof. This data can now be analyzed to increase efficiency of inventory management, procurement, and overall strategy. It has been shown across industries that inefficient inventory management can lead to financial loss and delays, with the main risks to the companies being financial, i.e. loss of capital, operational, i.e. poor forecasting and

planning, and strategic, i.e. inappropriate decision-making on pricing and diversification (Disk, 2023). For airline's operational cost, stores that are not optimized lead to reactive rather than proactive maintenance strategies. This approach tends to be more costly due to the need for urgent repairs and procurement of items at short notice. Inefficient inventory management can strain relationships with suppliers and disrupt the supply chain, leading to challenges in obtaining parts in the future. It is thus of the essence not only to correctly monitor levels in stores but also to predict future needs before conflicts arise. A growing research topic in this domain is thus predictive forecasting (Daily and Peterson, 2017; Gilges, 2023), powered by machine-learning technology, neural models and, more recently, large language models (LLMs), which are by definition predictive models for sequential data. The promise of LLMs is further enhanced by the very recent development of long-term memory mechanisms such as MemoryBank, that would allow them to further tailor responses to users (Zhong et al., 2023). Beyond theoretical research, such models must now be applied and tested in the field.

In addition, the data collection could be expanded to include personnel time on aircraft, personnel present at the station across teams, use of tools and vehicles, flight turnaround and other crucial variables, which can all be thought of as, essentially, transactions. Such large data sets could then be explored with predictive modeling tools. The company Moonware, for instance, inspired by Uber, explores vehicle tracking and ground traffic control to streamline turnaround performance. It has received $ 2.5 million in pre-seed financing (Alamalhodaei, 2023). Preventing inefficiencies in labor and resource allocations is the overarching theme. In addition to their potential benefit for the airline's balance sheet, such methods would enhance safety. In fact, models could be geared towards detecting the provenance of undocumented parts, the cause for repeated technical errors and other pitfalls, allowing for prevention to be implemented. Furthermore, digitizing documentation upon receipt and systematically recording counterfeits would allow for AI models to be trained in recognizing such counterfeits and, ultimately, emitting warnings in real-time. Technology is available to accurately collect and process large data sets continuously, building and training machine learning models in-house for many of airline's everyday engineering operations and challenges.

When it comes to strategic planning, if costs of maintenance are poorly accounted for, because the turnover of parts is not routinely monitored, decisions of fleet updates and routes are ill-informed. Strategic risks are also relevant in complex maintenance operations, such as in airlines selling engineering as a service as a secondary business such as at BA's JFK operation. This fairly common practice allows carriers to have access to engineering outside of their main hubs with fewer overhead. However, it requires that the provider and the customer airline work together in keeping optimal stores outside of their territory, although paradoxically under the remote national regulations of the foreign customer airline. In such settings, the decision of taking on new customer airlines by the service provider needs to be carefully weighed against personnel, equipment, and processes available, and priced in accordance. Overall, poor part control is linked to significant financial, operational, and

strategic mistakes that impact a carrier's balance sheet and survival in a competitive market. Having access to continuous quality data through a solution like Stores Keeper allows the creation of models and implementation of simulations that can effectively guide strategic solutions. As for inventory keeping, accurate and large data collection is the essential first step when building and training machine learning models.

Overall, this project and its future developments showcase that powerful in-house technological solutions can be created organically, by personnel on the ground with only average digital literacy. Having such bottom-up digital creation leads to continuous data collection that is well integrated with operations, meaningful and secure, as it is both encrypted and made tamper-proof by the blockchain. This is all the most important considering that predictive AI tools, the natural next step to guide airlines' management, strategy, and safety procedures, will only ever be as good as the data they are trained and tested on. Taken together, these elements bode a strategic question for airlines, one that warrants future investigations. That is, should airlines operate with a technology-forward mindset, giving training, tools, and technical support to employees at all levels to independently automate day-to-day operations? A structured system, allowing for pilot-testing of applications thus created and cyber security risk assessments, coupled with rewards for employees taking initiative could be imagined. Such a logic could safely launch an airline to the forefront of the technological transition we are living in all the while solving long-standing vulnerabilities. Furthermore, in the engineering department, recruiting geared towards individuals willing to engage with digital tools will build a workforce that is better suited to learn and care for ever-more digitally intensive aircraft. A bottom-up data-driven approach can result in a gain in efficiency and reliability, ultimately moving the whole industry towards higher safety standards.

References

Alamalhodaei, A., 2023. Moonware's AI lets airfield ground crews ditch the walkies. TechCrunch. URL https://techcrunch.com/2023/08/03/moonwares-ai-lets-airfield-ground-crews-ditch-the-walkies/ (accessed 2.25.24).

Alfredo Leonidas Vasquez Ubald, et al., 2022. (PDF) Mobile Application: A Proposal for the Inventory Management of Pharmaceutical Industry Companies [WWW Document]. URL https://www.researchgate.net/publication/363304871_Mobile_Application_A_Proposal_for_the_Inventory_Management_of_Pharmaceutical_Industry_Companies (accessed 2.5.24).

AMOS, 2024. AMOSmobile/STORES | Swiss-AS [WWW Document]. URL https://www.swiss-as.com/node/974 (accessed 1.16.24).

Antal, C., Cioara, T., Anghel, I., Antal, M., Salomie, I., 2021. Distributed Ledger Technology Review and Decentralized Applications Development Guidelines. Future Internet 13, 62. https://doi.org/10.3390/fi13030062

Axen, D., 2022. Sepolia Testnet Guide - What is the Sepolia Testnet? [WWW Document]. Moralis Web3 Enterp.-Grade Web3 APIs. URL https://moralis.io/sepolia-testnet-guide-what-is-the-sepolia-testnet/ (accessed 2.25.24).

Bhushan, S., Sharma, M.K., Gehlot, A., Pandey, S., Yamsani, N., Balyan, R., 2023. Inventory Management Using Blockchain and Big Data, in: 2023 4th International Conference on Smart Electronics and Communication (ICOSEC). Presented at the 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), IEEE, Trichy, India, pp. 1759–1763. https://doi.org/10.1109/ICOSEC58147.2023.10276234

Brock, T., 2022. How Much Airline Revenue Comes From Business Travelers? [WWW Document]. Investopedia. URL https://www.investopedia.com/ask/answers/041315/how-much-revenue-airline-industry-comes-business-travelers-compared-leisure-travelers.asp (accessed 1.27.24).

Cai, W., Wang, Z., Ernst, J.B., Hong, Z., Feng, C., Leung, V.C.M., 2018. Decentralized Applications: The Blockchain-Empowered Software System. IEEE Access 6, 53019–53033. https://doi.org/10.1109/ACCESS.2018.2870644

cnsight, 2021. Top 5 Cyber Attacks in the Aviation Industry. CnSightio - Cybersecurity Perform. Improv. URL https://cnsight.io/2021/04/16/top-5-cyber-attacks-in-the-aviation-industry/ (accessed 1.28.24).

Daily, J., Peterson, J., 2017. Predictive Maintenance: How Big Data Analysis Can Improve Maintenance, in: Richter, K., Walther, J. (Eds.), Supply Chain Integration Challenges in Commercial Aerospace: A Comprehensive Perspective on the Aviation Value Chain. Springer International Publishing, Cham, pp. 267–278. https://doi.org/10.1007/978-3-319-46155-7_18

Disk, 2023. Understanding the Risks of Inventory Management. Disk.com. URL https://disk.com/resources/understanding-the-risks-of-inventory-management/ (accessed 1.28.24).
divyaswarnkar, 2022. Create a custom connector in Azure Logic Apps [WWW Document]. URL https://learn.microsoft.com/en-us/connectors/custom-connectors/create-logic-apps-connector (accessed 1.5.24).
D'Oro, S., Polese, M., Bonati, L., Cheng, H., Melodia, T., 2022. dApps: Distributed Applications for Real-time Inference and Control in O-RAN. IEEE Commun. Mag. 60, 52–58. https://doi.org/10.1109/MCOM.002.2200079
FAA, 2019. FAA Proposes $3.9 Million Civil Penalty Against The Boeing Co. | Federal Aviation Administration [WWW Document]. URL https://www.faa.gov/newsroom/faa-proposes-39-million-civil-penalty-against-boeing-co?newsId=24456 (accessed 1.14.24).
Gilges, M., 2023. The Impact of Predictive Maintenance on Passenger Experience. Connect. Aviat. Today. URL https://connectedaviationtoday.com/the-impact-of-predictive-maintenance-on-the-passenger-experience/ (accessed 1.5.24).
Guzowski, M., 2019. How to overcome 500 items limit in PowerApps. CEO-Refleksje. URL https://michalguzowski.pl/how-to-overcome-500-items-limit-in-powerapps/ (accessed 12.2.23).
Henry, W., Kathawate, R., 2023. Using Blockchain to Drive Supply Chain Transparency and Innovation [WWW Document]. Deloitte U. S. URL https://www2.deloitte.com/us/en/pages/operations/articles/blockchain-supply-chain-innovation.html (accessed 1.20.24).
Ho, G.T.S., Tang, Y.M., Tsang, K.Y., Tang, V., Chau, K.Y., 2021. A blockchain-based system to enhance aircraft parts traceability and trackability for inventory management. Expert Syst. Appl. 179, 115101. https://doi.org/10.1016/j.eswa.2021.115101
iagcargo, 2018. Aircraft on Ground: How a technical fault can cost millions [WWW Document]. IAG Cargo Mag. URL https://iagcargomagazine.com/2018/07/03/aircraft-on-ground-how-a-technical-fault-can-cost-millions/ (accessed 1.16.24).
IFS, 2023. The role of modern MRO in performance excellence at China Airlines.
Jacobs, F.R., Chase, R.B., 2018. Operations and supply chain management. McGraw-Hill.
jopanchal, 2023. Create a custom connector from scratch [WWW Document]. URL https://learn.microsoft.com/en-us/connectors/custom-connectors/define-blank (accessed 2.25.24).
Kotzé, J., Antonopoulos, G., 2022. Con Air: exploring the trade in counterfeit and unapproved aircraft parts. Br. J. Criminol. 63. https://doi.org/10.1093/bjc/azac089
Massé, M.H., Massé, M., 2012. REST API design rulebook: designing consistent RESTful Web Service Interfaces. O'Reilly, Beijing Köln.
McCue, A., 2006. Best of CES 2024: ZDNET editors' top picks [WWW Document]. URL https://www.zdnet.com/article/ba-completes-six-year-sap-rollout/ (accessed 1.15.24).

Morales-Hernández, M., Morales-Jiménez, I., Osorio-Hernández, L., Díaz Sarmiento, B., 2021. Prototype of a web and mobile application for inventory management of a parts store using QR code. J. Comput. Syst. ICTs 9–20. https://doi.org/10.35429/JCSI.2021.19.7.9.20

Nakamoto, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System [WWW Document]. URL https://bitcoin.org/en/bitcoin-paper (accessed 1.29.24).

Nast, C., 2023. 5 Business-Class Seats That Are Actually Worth the Price Tag [WWW Document]. Condé Nast Travel. URL https://www.cntraveler.com/story/business-class-seats-worth-splurging-on (accessed 1.5.24).

Oksuz, O., 2024. A System For Storing Anonymous Patient Healthcare Data Using Blockchain And Its Applications. Comput. J. 67, 18–30. https://doi.org/10.1093/comjnl/bxac155

Peng, Y., 2021. Application of Blockchain in Civil Aviation, in: 2021 2nd International Conference on Computing and Data Science (CDS). Presented at the 2021 2nd International Conference on Computing and Data Science (CDS), IEEE, Stanford, CA, USA, pp. 198–202. https://doi.org/10.1109/CDS52072.2021.00042

Pettinari, P., 2023. Ethereum Virtual Machine (EVM) [WWW Document]. ethereum.org. URL https://ethereum.org/developers/docs/evm (accessed 1.29.24).

Pop, C., Cioara, T., Anghel, I., Antal, M., Salomie, I., 2021. Blockchain based Decentralized Applications: Technology Review and Development Guidelines. Future Internet 13, 62. https://doi.org/10.3390/fi13030062

Rademan, R., 2023. (4) Navigating Aviation Safety with Agile and Scrum: A Journey to Excellence | LinkedIn [WWW Document]. URL https://www.linkedin.com/pulse/navigating-aviation-safety-agile-scrum-journey-rudi-rademan/ (accessed 1.16.24).

SAP GUI Scripting API, n.d.

Satija, M.P., 2013. The Theory and Practice of the Dewey Decimal Classification System. Elsevier.

Sean Broderick, 2023. AOG Technics Ordered To Provide Details Of Engine Parts Transactions.

SITA, 2017. FlightChain - RESEARCH INTO THE USABILITY AND PRACTICALITIES OF BLOCKCHAIN TECHNOLOGY FOR THE AIR TRANSPORT INDUSTRY.

SkyThread, 2022. SkyThread Introductory Whitepaper [WWW Document]. URL https://mailchi.mp/d48ad70b3549/skythread-introductory-whitepaper (accessed 1.28.24).

Solidity, 2024. About [WWW Document]. Solidity Program. Lang. URL https://soliditylang.org/about/ (accessed 1.22.24).

Sudarssan, N., 2024. A Framework for Agricultural Food Supply Chain using Blockchain.

Tambe, N., 2024. Ethereum (ETH) Price Prediction For 2024, 2025 And 2030 [WWW Document]. Forbes Advis. INDIA. URL https://www.forbes.com/advisor/in/investing/cryptocurrency/ethereum-price-prediction/ (accessed 2.25.24).

Ukwandu, E., Ben-Farah, M.A., Hindy, H., Bures, M., Atkinson, R., Tachtatzis, C., Andonovic, I., Bellekens, X., 2022. Cyber-Security Challenges in Aviation Industry: A Review of Current and Future Trends. Information 13, 146. https://doi.org/10.3390/info13030146

Yankah, J., Kotey, S., Coffie, G., Herbertabeka, Tieru, C., 2024. Enhancing Construction Project Document Management with Mobile Apps VII, 2454–6186. https://doi.org/10.47772/IJRISS.2023.7012087

Zhong, W., Guo, L., Gao, Q., Ye, H., Wang, Y., 2023. MemoryBank: Enhancing Large Language Models with Long-Term Memory. https://doi.org/10.48550/arXiv.2305.10250

Annexes

1. Complete description of the system in place at the onset of the project

a. System's description at time 0

British Airways JFK holds a minimal store for the airline's operational needs. BA operates the B777-2/3 and B787-10 from LHR and LGW daily with up 9 flights. At the main store location in Terminal 8, JFK, most of BA's tooling and spares are held. An additional stores facility at "Hanger 10", houses customer airline stock holding. Customers include Norse Atlantic Airways, JAL, ANA, Qantas, Air New Zealand, Finnair, Aer Lingus, and Iberia, to which BA provides engineering support which includes stock holding, see Figure 1.

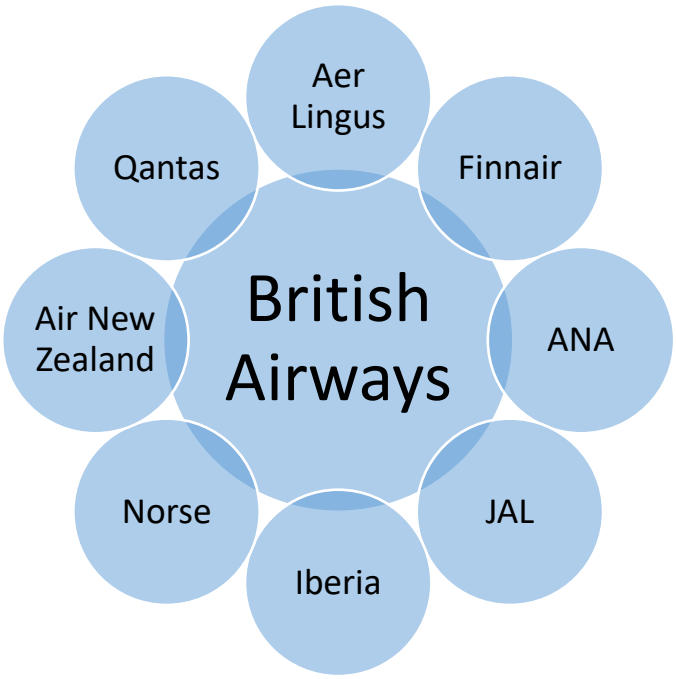


Figure 22 - British Airway's Customer Airlines at JFK Engineering.

A different designated engineer is responsible for keeping an inventory for each BA and customer airlines, using an excel spreadsheet loaded on the company Microsoft SharePoint environments. The spreadsheets do not include any rules, anyone with access to BA's MS environment is free to view and edit. Furthermore, there is no uniformization regarding the format of the spreadsheet, columns, or content, leaving it completely to the responsible engineer's criteria. When new parts are received by the stores, the same engineer is expected to enter them in the spreadsheet without, however, a time-constraint between reception of the part and its registration in the dedicated spreadsheet. Furthermore, when a part is used by any engineer in the team while doing maintenance on an aircraft, the engineer in charge must subtract it from the spreadsheet. See Figure 2.

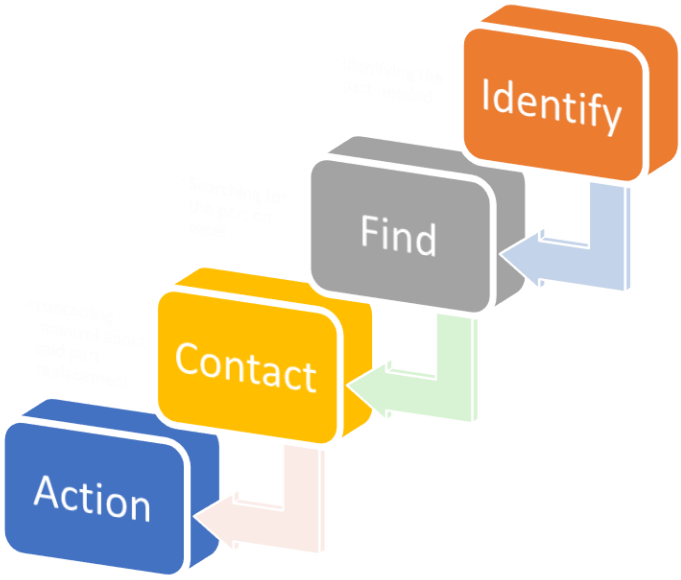


Figure 23 - Diagram of the Steps to Retrieve a Part in the System at Time 0.

Furthermore, the engineer in charge of the airline's specific inventory is required to conduct quarterly spot checks. Spot checks focus on the transactions that occurred in that same quarter, i.e., if new items received gave way to correct input on the dedicated spreadsheet and if items used gave way to correct subtraction or deletion from the spreadsheet. Upon completion of the spot check, the engineer in charge of those stores must log and retain the information incase called upon during an audit. There is no oversight regarding the spreadsheet's contents by either BA or the customer airlines' quality department, the spreadsheet itself is not shared with the airlines quality, who hold their own inventory management systems.

When conducting maintenance, installing a part is relatively rare, on average an estimate of 1 to 3 parts / month are changed for each customer airline. This is because there are only a few turnovers per day for each customer airline, 1-3/day, and the operation is focused on sending the aircraft back to its main hub on time. Repairs that involve installing parts are only done if essential for airworthiness. For instance, a seat cushion replacement due to stains will not be performed due to the time necessary to retrieve the cushion in stores and complete the necessary paperwork. Thus, the airline's operation will be faced with the choice of blocking the stained seat or selling it with poor quality, reducing revenue or decreasing customer satisfaction, respectively.

When airworthiness requires a new part to be fitted, the engineer conducting aircraft maintenance most often calls the engineer in charge of the airline's inventory spreadsheet to ask whether a part needed for maintenance exists in storage and where it is located. If the local engineer in charge of the airline's spreadsheet is not available, maintrol at the airline's hub is called (e.g. in Dublin for Aer Lingus) to determine if the part needed should be held in storage at JFK, according to the last quarterly spot check. In some exceptional instances, when the engineer conducting the maintenance is familiar with the specific spreadsheet's format and its location on BA's MS environment, they will consult it directly. After the part is installed, the engineer performing maintenance completes the airline's tech log, which returns to main hub with the aircraft, and informs of the use of the part via a detailed report. With this information, the airline will update its central stores inventory system, subtracting the use

part from their record of inventory at JFK. Additionally, the engineer conducting the maintenance informs the engineer in charge of the specific airline's store spreadsheet that a part has been used, usually orally, via a note left on the desk, an e-mail, or a text message.
Decisions on which parts and consumables will be made at JFK is entirely the responsibility of each airline's central hub. All spares from customer airlines are delivered to Terminal 8 (T8) in a dedicated location. However, if engineers at JFK find low stocks of certain items, usually upon conducting maintenance since no monitoring system is in place, a request can be placed to the airline's main operations. Conversely, BA will be notified by the airline when an item that is expected to be held at the customer airline's stores in JFK is nearing expiry and must be placed in the unserviceable rack.
For BA's dedicated inventory at JFK, processes are also slightly different to those for customer airlines, as they are managed by unionized employees with direct access to BA's inventory systems. A unionized BA engineer has, among other duties, the role of stores manager. This role used to be fulfilled by a dedicated stores employee, however that position was cut during the COVID pandemic along with many others. The engineer now in charge is self-taught in SAP and in charge of input and output of spares.
When another engineer uses a spare in stores, they register the use in a physical book in the storage facility, handwriting it with a pen. In addition, the aircraft's tech log is completed and sent back to the hub with the aircraft, and a report of BA's maintenance software system is directly sent to the airline. During their work hours the engineer in charge of the stores then does the corresponding data entry in BA's SAP. Part changing in BA is more frequent than for customer airlines, with 9 flights a day and approximately 10 parts / month installed. Summer additionally sees increased schedules and part use.
For BA as for customer airlines, in this case headquarters in London, are responsible for deciding which parts will be kept at JFK stores and ordering these spares to be delivered at JFK. Contrary to customer airlines', BA has full and direct access to the SAP system maintained by the JFK operation.
When a new part, sent by a customer airline or BA, is delivered to the dedicated shelf in the T8 stores, according to a unique shipping address. The received part must then be placed on their definitive storage location at Terminal 8 (T8) or Hangar 10 (H10) bonded stores by the engineer in charge of the inventory for the airline. If the part needs to be transported to H10, because transport is occurring between two bonded stores, it is legally required to document the transport. There is no procedure in place for documenting this transport, which was highlighted in a self-audit concluded in August 2023.

b. System's limitations at time 0

At the onset of the project, several limitations were found in the system in place. Lack of oversight and absence of procedures in how inventory spreadsheets are filled by the one employee in charge makes them of little use without said employee's guidance. In a 24/7 and time constrained operation, relying on calls to a specific engineer followed by calls to a distant Maintrol is highly ineffective. It has a direct impact on turnover time, having caused AOGs as the crew is timed out. In addition, engineers in charge of maintenance rely on their experience to estimate the time needed to install a part and, accordingly, to decide on what repairs to carry out during the turnover, given the priority is sending the aircraft back to its main hub on time. As such, when it comes to repairs that do not affect airworthiness, if the engineer's experience includes long processes to locate and retrieve a part, decisions will be made to forgo any non-essential repairs, causing seats to be blocked or customer comfort to be affected. Furthermore, the same reasoning could also push engineers towards foregoing

part replacements that are mandatory for airworthiness, especially under high-stress and staff-shortage circumstances, when susceptibility to unconscious biases increases, causing security concerns.

In addition, the absence of continuous harmonization between an airline's stores records and local JFK records, especially when it comes to the exact location of each part, creates openings for parts to be lost and misplaced. Furthermore, discrepancies between the two records make accurate prediction of needs and placing of orders difficult. The quarterly spot checks, if they miss errors in input and outputs in the period, compound to increasingly incorrect local registries over the years.

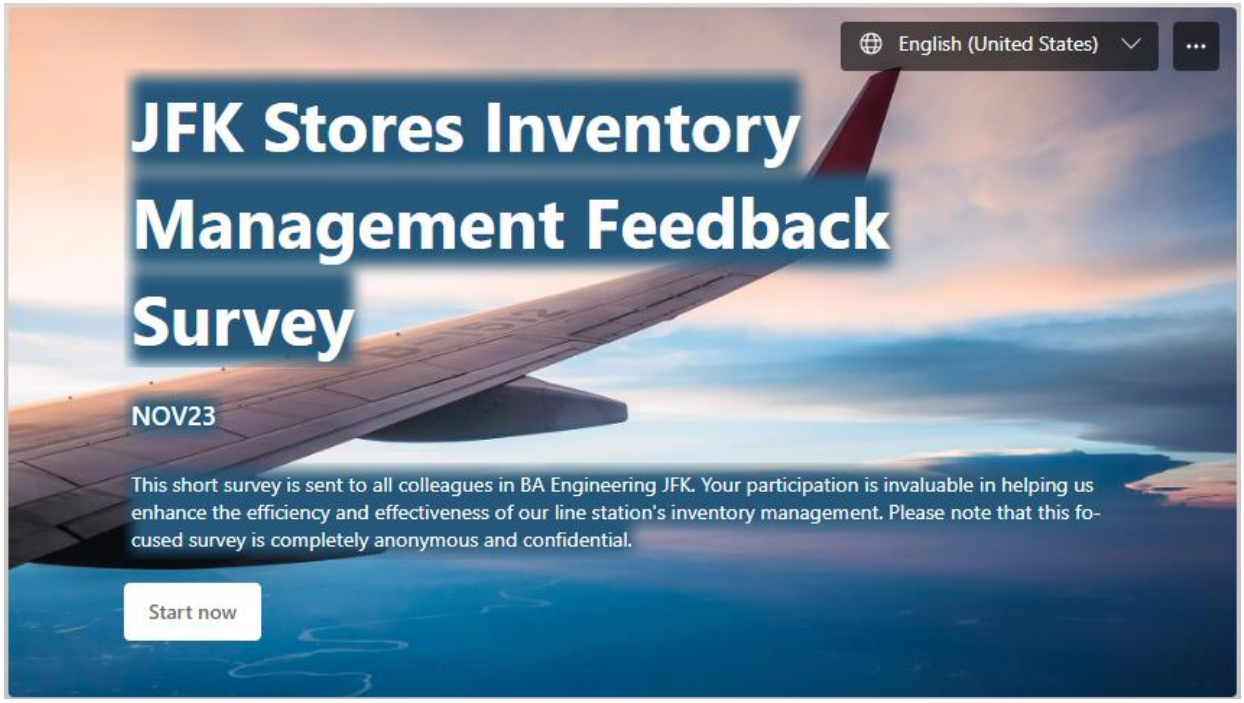
For BA stores, the main difficulty resides in an employee having to interpret and convert hand-written entries by over 25 different employees into SAP entries a posteriori. Time is of the essence and frequently employees simply forgo registering retrieving parts from stores, leaving used parts to be completely unaccounted for possibly until the next quarterly spot check or, realistically, until the part is needed again and found to be missing. Furthermore, even when time is taken to inscribe the removal of the part in the book, speedy handwriting is prone to errors. All these fields are crucial to uniquely identify parts, trace them, and determine if they can be used in an aircraft. As such, errors in their SAP input can lead to expired parts being mistakenly installed in an aircraft. Furthermore, if parts are not correctly identified, several entries for identical parts will be created in the SAP, instead of updating the quantity in one unique entry. Such errors will have repercussions in monitoring stores' activity and predicting what parts need to be replenished. A poorly updated SAP system increases operational costs for BA in two ways. On the one hand it causes unnecessary parts to be purchased and, on the other hand, it causes the stores to lack necessary parts for maintenance.

Importantly, because each airline, be their customer or BA, relies on one dedicated employee's knowledge of the store's management system and manual updates, difficulties arise upon employee turnover. This is a frequent occurrence in an international and highly mobile industry facing severe staff shortages in recent years. The rate of turnover is only predicted to worsen in the coming years as demand increases without sufficient relative uptake in aircraft type-specific training of new generations of engineers. With this, there is no consistent organizational and signage systems across both H10 and T8 bonded stores, creating additional difficulties in oversight and when employee turnover occurs. There is also no process in place to document transport of parts between bonded stores, directly infringing legal requirements.

Lastly, in urgent situations, parts belonging to one airline could be of use for repairing aircraft belonging to another airline, provided authorization from both airlines are obtained, unless airlines have previously entered into a parts pooling agreement. By having no centralized record of parts in stores at JFK for all customer airlines and BA, this additional flexibility becomes impossible. If a central ledger existed, one could even envision part pooling between customer airlines, further optimizing inventory, reducing operating costs and increasing revenue.

2. Complete survey results at Time 0

a. Survey Questions



Role-Specific Insights

Different roles have varying interactions with the inventory management system. Understanding the role of each respondent can help tailor improvements to meet the specific needs of different users.

1. Select your position. *

- ◯ AMM
- ◯ SMM
- ◯ DMM
- ◯ Maintenance Rep
- ◯ Mechanic

2. Years of Service at BA Engineering JFK? *

- ◯ 0-5
- ◯ 5-10
- ◯ 10-15
- ◯ 15-20
- ◯ 20+

Receipting of Spares/Tools

1 'Strongly Disagree' and 5 'Strongly Agree'.

3. The process for receipting new spares/tools is clear and straightforward. *

1	2	3	4	5

4. I am promptly notified when new spares/tools are received in the store. *

1	2	3	4	5

5. The documentation and data entry for new spares/tools are consistent and accurate. *

1	2	3	4	5

6. It is easy to remove un-serviceable items from our stores. *

1	2	3	4	5

7. I understand the process of moving spares from T8 to H10. *

1	2	3	4	5

Organization of the Stores Facility

1 'Strongly Disagree' and 5 'Strongly Agree'.

8. The physical arrangement of the store's facility is well-organized. *

1	2	3	4	5

9. Spares/tools are consistently stored in their designated locations. *

1	2	3	4	5

10. The stores facility is maintained in a clean and orderly manner. *

1	2	3	4	5

Comfort in Navigating the Stores to Find Items

1 'Strongly Disagree' and 5 'Strongly Agree'.

11. It is easy to locate the items I need in stores. *

1	2	3	4	5

12. I feel confident navigating around the stores facility to find what I need. *

1	2	3	4	5

13. The layout of the stores facility makes it easy to access different sections. *

1	2	3	4	5

14. Signage and directions within the store's facility are clear and helpful. *

1	2	3	4	5

15. The labeling and categorization of spares/tools in stores are clear and logical. *

1	2	3	4	5

Efficiency in Retrieving and Checking Out Parts/Tools

1 'Strongly Disagree' and 5 'Strongly Agree'.

16. Retrieving a part/tool from the inventory is usually a quick process. *

1	2	3	4	5

17. The check-out process for tools and parts is efficient and time saving. *

1	2	3	4	5

18. I rarely experience delays or issues when retrieving or checking out items. *

1	2	3	4	5

Adoption of Digital Solutions

1 'Strongly Disagree' and 5 'Strongly Agree'.

19. I believe that digitizing inventory management would enhance accuracy and efficiency. *

1	2	3	4	5

20. A digital inventory management system would improve the ease of tracking and managing stock levels. *

1	2	3	4	5

21. Implementing a digital system would make the process of receiving and cataloging new spares/tools more streamlined. *

1	2	3	4	5

Usability and Convenience of an App

1 'Strongly Disagree' and 5 'Strongly Agree'.

22. I feel confident in my ability to adapt to a digital/app-based inventory management system. *

1	2	3	4	5

23. Having an app-based solution for inventory management would make it more convenient to access information on-the-go. *

1	2	3	4	5

24. An app for inventory management should have user-friendly features such as bar-code scanning and real-time updates. *

1	2	3	4	5

Perceived Benefits and Time Management

1 'Strongly Disagree' and 5 'Strongly Agree'.

25. I think an app could provide more detailed and up-to-date information about inventory status than the current system. *

1	2	3	4	5

26. An app-based inventory management system would likely reduce the time spent in locating and checking out items. *

1	2	3	4	5

27. Using an app could potentially reduce errors and misplacement of inventory. *

1	2	3	4	5

Training and Support

1 'Strongly Disagree' and 5 'Strongly Agree'.

28. I would be willing to undergo training to use a new digital inventory management system effectively. *

1	2	3	4	5

29. Adequate support and resources should be provided for a smooth transition to a digital inventory management system. *

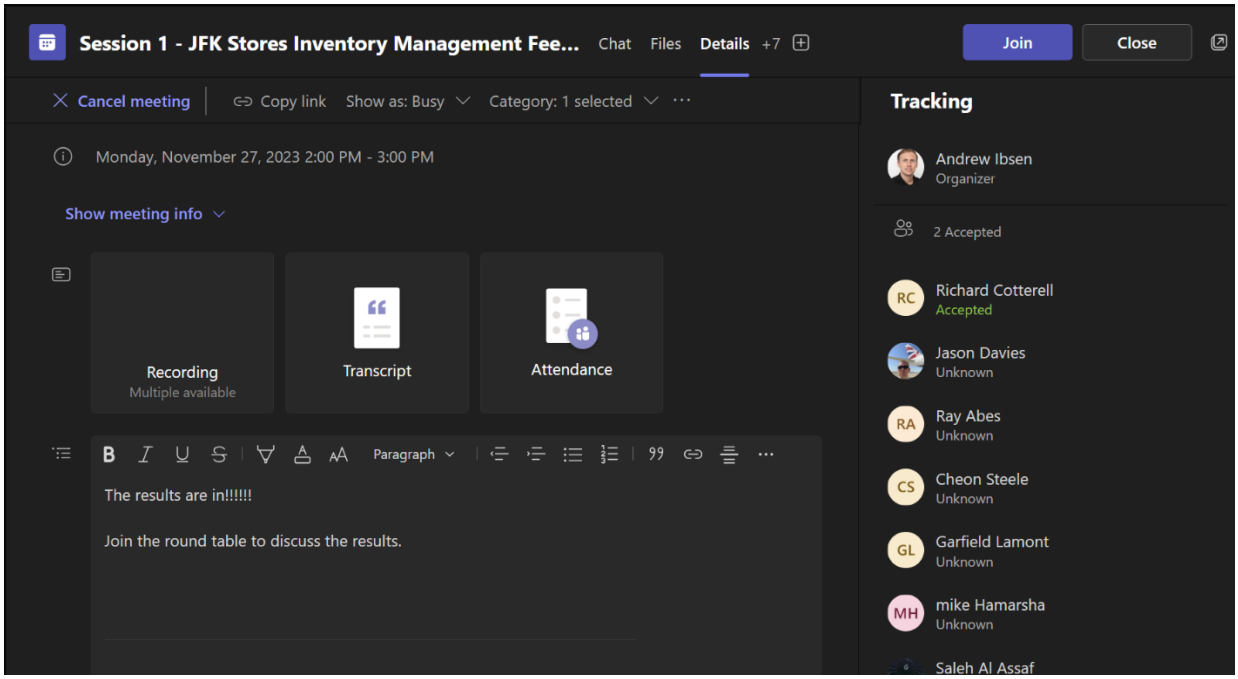
1	2	3	4	5

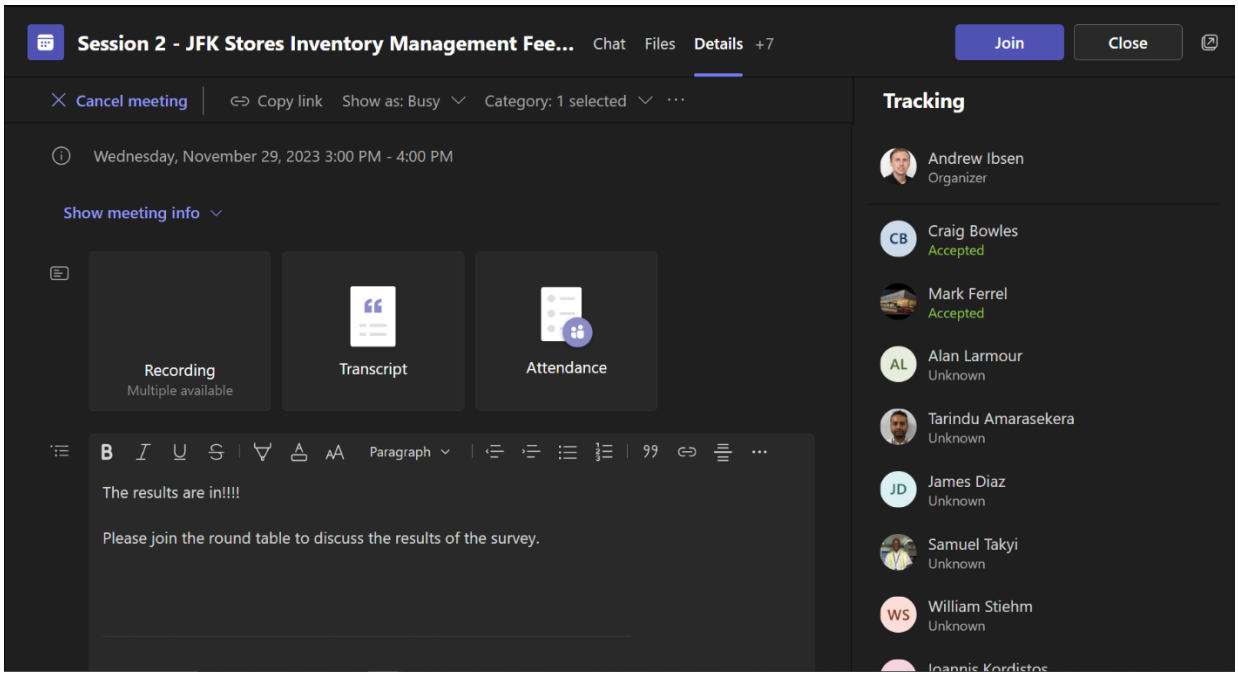
b. Survey Results

ID	Start time	Completion time	Email	Role	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	11/12/23 21:47:19	11/12/23 21:47:22	anonymous	DMM	3	2	3	5	4	4	4	4
2	11/13/23 1:36:03	11/13/23 1:41:13	anonymous	DMM	3	2	3	1	4	3	4	3
3	11/13/23 10:35:25	11/13/23 10:43:00	anonymous	DMM	5	1	4	4	1	4	4	4
4	11/13/23 11:46:44	11/13/23 11:49:01	anonymous	DMM	2	2	2	2	1	4	4	5
5	11/13/23 21:00:22	11/13/23 21:04:15	anonymous	Maintenance Rep	1	4	1	1	1	5	4	5
6	11/14/23 16:11:40	11/14/23 16:14:26	anonymous	DMM	3	2	2	2	1	5	5	5
7	11/15/23 11:20:22	11/15/23 11:24:46	anonymous	DMM	4	3	5	4	4	4	4	4
8	11/15/23 13:17:40	11/15/23 13:19:41	anonymous	Mechanic	3	3	3	3	3	3	3	3
9	11/15/23 14:17:12	11/15/23 14:19:54	anonymous	Mechanic	4	3	3	3	3	4	4	4
10	11/15/23 15:06:59	11/15/23 15:11:42	anonymous	Maintenance Rep	3	3	3	3	3	3	3	2
11	11/15/23 17:24:27	11/15/23 17:27:42	anonymous	DMM	2	4	4	2	3	4	4	4
12	11/15/23 21:24:21	11/15/23 21:31:16	anonymous	Mechanic	2	1	2	1	1	3	4	4
13	11/16/23 6:27:03	11/16/23 6:31:16	anonymous	Maintenance Rep	1	1	1	1	1	4	4	4
14	11/17/23 13:21:21	11/17/23 13:26:51	anonymous	DMM	4	4	4	4	3	4	4	4
15	11/17/23 23:04:26	11/17/23 23:07:06	anonymous	Mechanic	5	5	5	5	5	5	5	5
16	11/18/23 16:07:26	11/18/23 16:11:01	anonymous	Mechanic	3	1	3	3	4	3	3	3
17	11/18/23 18:31:22	11/18/23 18:33:26	anonymous	Mechanic	3	3	3	3	3	3	3	3
18	11/20/23 22:39:18	11/20/23 22:44:40	anonymous	SMM	5	3	3	4	3	1	1	3
19	11/26/23 20:49:06	11/26/23 20:53:27	anonymous	AMM	5	1	3	3	2	4	4	3

ID	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27
1	4	3	5	3	4	4	4	3	5	5	5	5	5	5	5	5	5	5	5
2	2	2	4	2	3	5	2	2	4	5	5	5	5	5	5	2	4	5	5
3	4	4	4	4	3	3	2	2	5	4	4	5	5	5	4	5	5	5	4
4	3	5	5	4	4	4	2	2	3	3	3	5	5	5	5	5	5	5	5
5	3	3	4	3	3	2	1	1	4	4	4	2	1	1	1	1	1	2	2
6	1	1	5	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	4	4	4	4	4	4	5	4	4	5	5	4	5	5	5	5	5	3	4
8	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
9	3	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
10	2	2	2	2	2	4	4	3	4	4	4	3	3	3	4	4	4	4	4
11	2	2	3	2	2	3	1	2	4	4	4	4	4	4	5	4	4	5	4
12	2	4	4	4	4	1	1	1	1	1	1	3	2	5	1	1	3	3	5
13	4	3	4	4	4	3	4	3	4	4	4	4	3	3	3	3	3	4	4
14	3	3	4	4	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4
15	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
16	3	4	4	4	3	2	2	2	5	5	5	5	5	5	5	5	5	5	5
17	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	5
18	1	3	4	2	1	4	4	4	5	5	5	5	5	5	5	5	5	5	5
19	3	3	4	5	4	4	5	4	4	4	4	5	5	5	5	5	5	5	4

3. Evidence of Focus Group Meeting at Time 0





Discussion Topics

- Feedback – They like the look and function of the app.
- Users wanted to be able to type instead of just use QR Code scanning.
- Have had trouble with the internet connection.
- Feels like it's a double entry if not connected to SAP.
- Who's going to keep oversite of this moving forward?
- All staff to be given iPads in the spring.

4. Meeting Notes from Semi-Structured Interview at Time 0

a. SMM Alan Larmour on 28NOV28

Q: How do current systems manage oil consumption tracking, and where do they fall short?
A: We try and get an order in when stocks run low.
Q: What specific features should the app include to ensure accurate accounting of oil consumption on station?
A: Needs to be quick to use and have the inventory separated between airlines.
Q: Can you describe the ideal workflow for using the app to enforce local working procedures?
A: X-Form to submit to QA to put as a local procedure. Read and Signs will then be sent out.
Q: What challenges do you anticipate in implementing and adhering to these new procedures through the app?
A: The Union will push back.
Q: How should the app facilitate compliance with local working procedures without disrupting existing workflows?
A: Well, the current system of manually entering consumed parts to then be updated into SAP is an informal solution the Union has come up with to manage oversite of stores.
A: Oil consumption on stations needs to be accounted for.

b. SMM Jason Davies on 02DEC23

Q: In ensuring that all items are accurately reflected in SAP, what role should the app play in synchronizing data?
A: It should be a bridge, if it can't transact directly with SAP at this time due to SAP limitations.
Q: For the process of transferring parts from T8 to H10, what specific features should the app have to streamline and document this procedure?
A: It needs a Log…
Q: How should the app facilitate compliance with local working procedures without disrupting existing workflows?
A: Well, the current system of manually entering consumed parts to then be updated into SAP is an informal solution the Union have come up with to manage oversite of stores.
Q: What feedback mechanisms can be integrated into the app to continuously improve the process of accounting for oil consumption?
A: The SMMs need to be notified automatically when stock minimums are met. Working with the union. Making sure items are also on SAP. Transferring parts from T8 to H10 – Procedure needs to be put in place.

7.1.1 c. Senior Maint Rep. Anthony D'erasmo on 05DEC24

Q: What kind of involvement would you like to see from the Union?
A: The physical work needs to be done by the union and managed by management.
Q: How difficult has it been managing the current inventory management system?
A: People aren't communicating what they are taking from stores. It's a bonded area, there needs to be better oversight.
Q: What issues do you have with the current system?
A:

- Lack of training… People need to be trained to use SAP.
- Consuming items from Stores need to be documented.
- Oil is consumed by the station not by SAP.
- Any app made needs to be integrated with SAP.

Union involvement is paramount.

5. Working Procedures Created and Implemented

a. Transfer Between Bonded Stores

BRITISH AIRWAYS

Amendment to "Line Maintenance Station Standards - MA-LM-0-1-WI.16"

Transporting Aircraft Spares between T8 and H10 bonded stores.

1. **Purpose:**

   This amendment outlines the procedure for receipting and handling of bonded spares/tools at British Airways Engineering JFK, specifically for transferring them from Terminal 8 to the storage facility in Hangar 10.

2. **Procedure:**

   2.1 Receipt of Parts and Tools at Terminal 8:

   2.1.1 Upon receipting of parts and tools at Terminal 8, the following "Receipt In" and Receipt Out" process shall be followed:

   2.1.2 A "Receipt In" and Receipt Out" table shall be used to document the transaction details, including the Part Number, Serial Number, Receipt In T8 and Receipt Out T8, Receipt In Hanger 10 and Receipt Out Hanger 10.

   2.1.3 The tables shall include the columns as outlined below:

2.2.3 The bonded spares/tools shall be securely packaged, with the applicable paperwork, ensuring compliance with all relevant regulations.

2.2.4 The responsible personnel shall transport the bonded spares/tools to the storage facility in Hangar 10.

2.2.5 Upon arrival at Hangar 10, the responsible personnel shall record the receipt in the "Receipt In (Hangar 10)" column adding their staff number, sign, and date.

2.2.6 The bonded spares/tools shall be stored in the designated area within Hangar 10, following appropriate storage guidelines.

3. Compliance and Documentation:

3.1 All personnel involved in the receipt and handling of bonded spares/tools shall comply with the procedures outlined in this amendment.

3.2 The completed "Receipt In" and Receipt Out" tables shall be maintained as part of the record-keeping process, ensuring traceability of the bonded spares/tools from Terminal 8 to Hangar 10. One table form is to be kept within the store's facility at Terminal 8, next to the already enforced Tooling check-in/out sheets. The other table form is to be kept at Hanger 10. Both forms will tie together to complement each other.

Please ensure that this amendment is implemented and communicated to all relevant personnel. It should be incorporated into the "Line Maintenance Station Standards - MA-LM-0-1-WI.16" document.

(See Tables – One for Spares – One for Tooling).

Part Number	Serial Number	Receipt In (Terminal 8)	Receipt Out (Terminal 8)	Receipt In (Hangar 10)	Receipt Out (Hangar 10)
- Add P/N	-Add S/N	-Staff Number - Sign - Date	-Staff Number - Sign - Date	-Staff Number - Sign - Date	-Staff Number - Sign - Date

2.1.4 The responsible personnel shall fill in the relevant information in the "Receipt In" and Receipt Out" table upon receipt and release of the bonded spares/tools.

2.2 Transfer of Bonded Spares/Tools to Hangar 10:

2.2.1 Once parts and tools are received at Terminal 8, and it is determined that they are bonded spares/tools, the following process shall be followed for their transfer to Hangar 10:

2.2.2 The responsible personnel shall record the receipt in the "Receipt In (Terminal 8)" adding their staff number, sign, and date.

British Airways Engineering
X FORM
TITLE: STATION LOCAL INSTRUCTION
X FORM ref. no.: MA-X667
Issue Number:3 Issue Date: 02/05/2019

BRITISH AIRWAYS ENGINEERING STATION LOCAL INSTRUCTION

Ser Nr: STA-YEAR-S/N JFK-2023-20213

Date : 19NOV2023

THIS STATION INSTRUCTION HAS BEEN RAISED IN ACCORDANCE WITH
TECHNICAL PROCEDURE MA-LM-0-1-WI.22 ENTITLED LOCAL STATION INSTRUCTIONS

TITLE:

Subject Matter.....

"Transporting Aircraft Spares between T8 and H10 bonded stores."

Please see attaching document for details.

Authorised By:

ASSOCIATED REFERENCES:

MA-LM-0-1-WI.22 **(Prev X1858)**

b. Store Organization and Signage

BRITISH AIRWAYS

Questions and comments, please contact:

Andrew Ibsen

+1(321) 831 9337

andrew.ibsen@ba.com

"Hierarchical Location Identification System" (HLIS) for JFK.

The term "Hierarchical" emphasizes the multi-tier structure of the location identification system. The acronym "HLIS" offers a concise, easy-to-reference name for internal discussions and documentation.

Define Location Hierarchy: Standardize hierarchy of locations (e.g., H10 → Row → Rack → Shelf → Bin.

Each row consists of a rack with shelves and sometimes bins.

- Location – H10 or T8
- Sub location – Row – A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z
- Sub, Sub location – Rack – 1/2/3/4/5/6/7/8/9/10/11/12/13 or A/B/C/D/E/F/G/H/I/J/K/L/M
- Sub, Sub, Sub location – Shelf – 0/1/2/3/4/5/6/7/8/9
- Sub, Sub, Sub, Sub location – Bin – 0/1/2/3/4/5/6/7/8/9/10/11/12/13 etc.

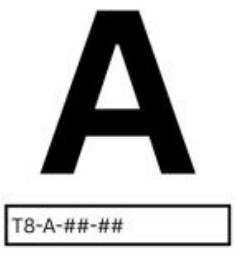
The 0 means that the location doesn't have that sub location. For example, the Oil rack has no bin, thus the bin number is 0. All zeros at the end means there is no bin location.

Unique Identifiers: Unique identifiers for each level of location. For example, H10-A-3-2-1 represents H10 (Location), A (Row), 3 (Rack), 2 (Shelf), 1 (Bin).

Other outliers and unique identifiers will follow the same concept.

Examples of outliers: Non aircraft equipment cabinet, GSE cabinet, PPE cabinet, vehicle related tooling lockers, ground equipment steps/platforms and axle jacks.

Unique Identifiers will be given to the Wheel Change Kit Trailers.



Additional location categories are integrated into the "HLIS" to accommodate these outlier storage types. For instance:

Locker – For individual lockers.

Closet – For storage closets.

Cabinet – For cabinets.

Cage – For the cages

Outlier Identifiers for example:

LKR for lockers,

SHD for shed,

CBNT for cabinets,

CG for cages.

Additional location categories are integrated into the "HLIS" to accommodate these outlier storage types. For instance:

Locker – For individual lockers.

Closet – For storage closets.

Cabinet – For cabinets.

Cage – For the cages

Outlier Identifiers for example:

LKR for lockers,

SHD for shed,

CBNT for cabinets,

CG for cages.

Location Coding: Integrate these identifiers into your existing location coding system. For instance:

- H10-CG-LKR-1 could represent the first locker in the H10 Cage.
- H10-CBNT-3 might be the third cabinet in location H10.
- OT8-SHD-001 represents the outside T8, first shed.
- T8-CG for the cage in location T8
- H10-CG for the cage in location H10.

Sub-locations for Outliers: If these storage units have internal divisions, like shelves in a closet or drawers in a cabinet, you can extend the coding. For example:

H10-CBNT-3-1 for the first shelf/drawer in the third cabinet at location H10.

T8-CG-1-2 could represent the cage in T8. There within rack 1, shelf 1.

H10-CG-3-4-1 could be for the cage in H10. There within rack 3, shelf D, bin 1.

Certain ground equipment like steps, platforms, jacks wheel change kits, will also need a location. However, they are located outside T8 stores. These will be labelled OT8-GE. For Outside T8 Ground Equipment and OT8-WCT for the Wheel Change Trailer Kit.

A unique identifier for each wheel change kit will be as follows: OT8-WCT-028 for the first wheel change trailer kit, OT8-WCT-029 for the second kit, and so forth.

Colour Coding Identifiers in Stores

- Red and white striped tape gives a visual representation of non-aircraft use equipment and materials. This gives an immediate visual representation of caution, that said equipment is not for aircraft us.

- Yellow and Black striped tape gives a visual representation of aircraft tooling. This has an immediate identifier for tools within our T8 and H10 stores.

- Red tape gives a visual identifier of unserviceable or condemned aircraft equipment and spares.

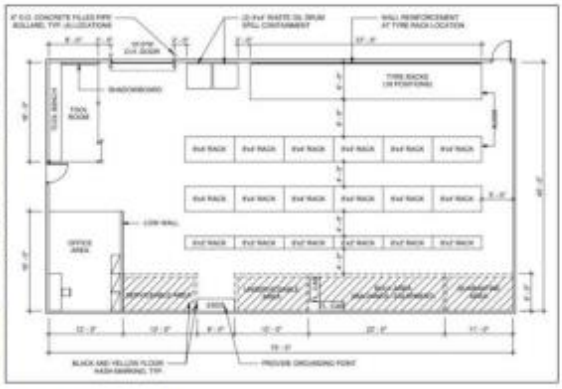
- Green tape visually identifies serviceable parts.

- Blue tape visually identifies items that are in quarantine, awaiting inspection, or under some form of evaluation. It can also mark parts that are being held for further decision-making.

Some examples:

Our Stores Locations

Hanger 10



Hanger 10 consists of 4 rows A/B/C/D/E/F, lockers, cabinets and cages

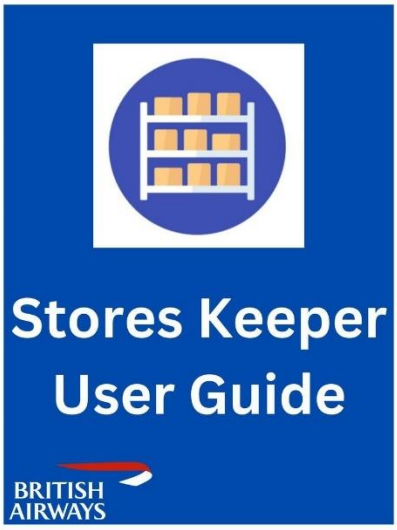
T8 Stores

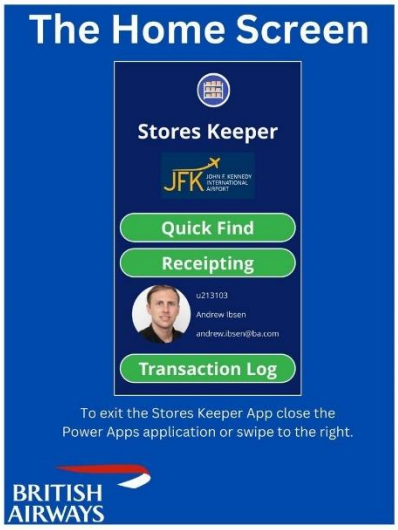
T8 consists of 4 rows A/B/C/D/E/F/G, lockers, cabinet and cages.

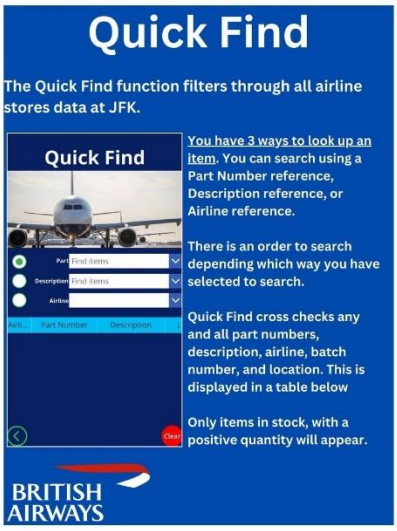
Stores Receipting – Customer and British Airways

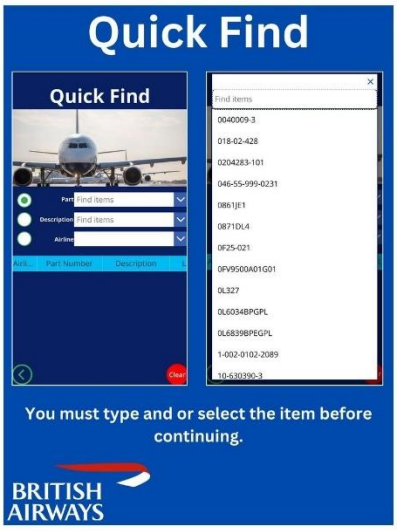
Receipting and stock picking practices should adhere to BA's quality standards policy First In, First Out (FIFO) and Last In, First Out (LIFO)

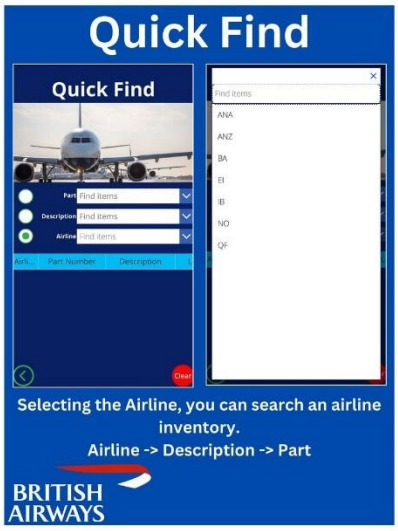
6. Stores Keeper Mobile App User Guide

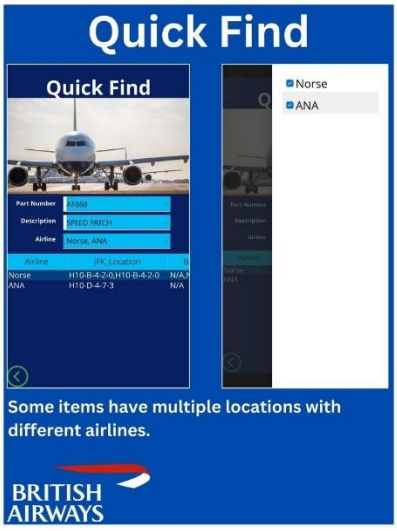


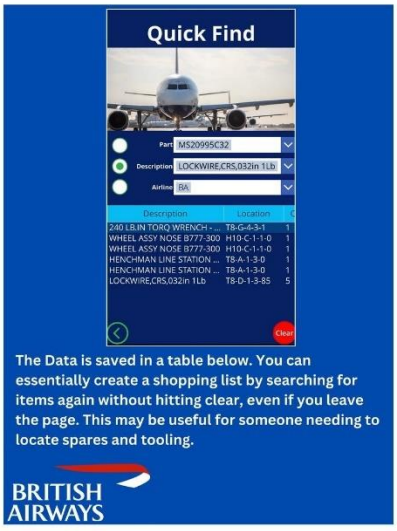


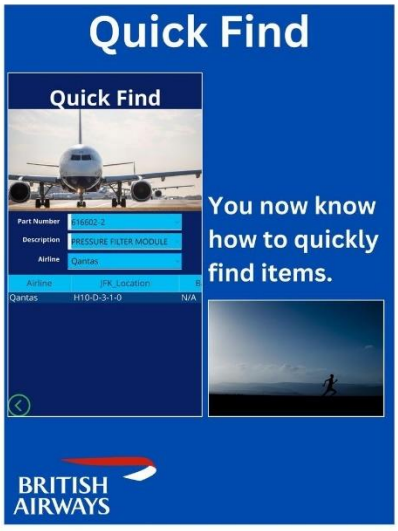












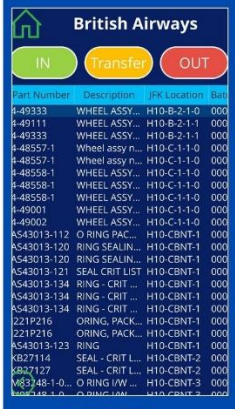
Receipting

The Airline Inventory page shows each airline afiliated with the station. Each airlines receipting process is the same.

Here you have a live temperature reading for each Stores Location.

BRITISH AIRWAYS

Receipting



IN - Items Received

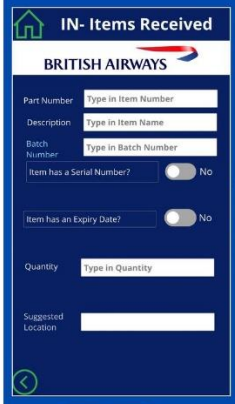
Transfer - Items moving from one location to another

Out - Items used.

Here you have a quick reference table for all the items associate with said airline.

BRITISH AIRWAYS

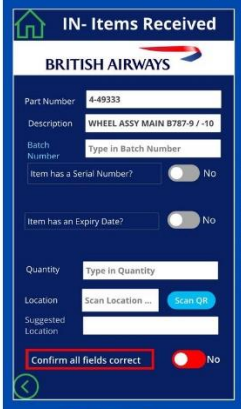
Receipting



IN - Items Received is the most important process in the receipting process.

BRITISH AIRWAYS

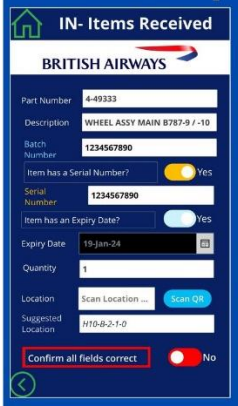
Receipting



When a recognized part number for the airline is entered, the description will autopopulate.

BRITISH AIRWAYS

Receipting

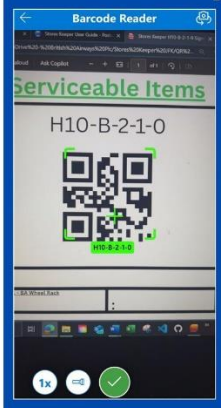


Notice the prepopulated suggested location. This shows where the item has been before and gives an idea where to put the item.

You wont be able to save the data unless you scan the location QR code.

BRITISH AIRWAYS

Receipting

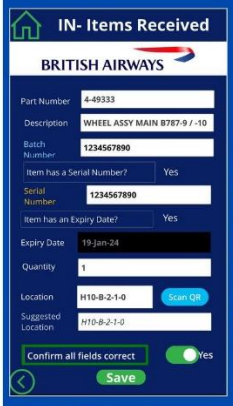


The QR Scanner tick mark goes green when it identifies the QR Code

There is a zoom and light function to aid scanning.

BRITISH AIRWAYS

Receipting



You must confirm all fields are correct in order to save. The purpose of this is to give the user a moment to double check the information is correct as there is no undo button.

BRITISH AIRWAYS

Receipting

Fantastic! That was the toughest part! You are now ready to receipt items into stores.

BRITISH AIRWAYS

Transfer

Transferring lets you move items from one location to another. However, the location must be associated with that airline.

Common Airline locations are the receivables rack in T8 and the U/S racks in T8 & H10.

Steps:

1. Scan Item Location
2. Select Item
3. Enter Quantity
4. Scan New Location

You must confirm the fields are correct before saving. If it doesn't let you confirm and reverts back to "No". This means you have an error and need to check you have scanned the right location.

BRITISH AIRWAYS

Items Out

After scanning the location, the part number will automatically appear. If there is more than one part number associated with that location. Select the appropriate part number. You must then input the quantity you are using.

BRITISH AIRWAYS

Transaction Log

The Transaction Log records each change in item location.

You must scroll to the bottom to see the latest transaction.

Transactions Registered:

- IN
- OUT
- TRANS-Out
- TRANS-IN

Each transfer registers the previous location and new location together.

The log also identifies the individual who has conducted the transaction.

BRITISH AIRWAYS

SharePoint

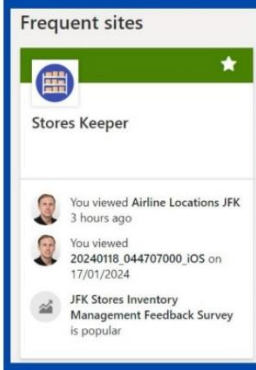
Access the apps navigation window.

Click Sharepoint.

To access the SharePoint site; type “Stores Keeper” in search.

BRITISH AIRWAYS

SharePoint



I recommend favoriting the site for quick access.

BRITISH AIRWAYS

SharePoint

The homepage has the App QR to quickly access or download the Stores Keeper App.

You will also find this user guide if you scroll down the SharePoint homepage.

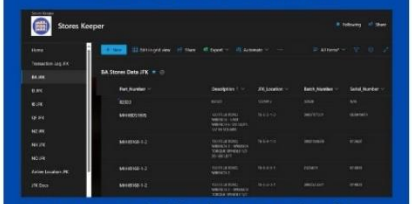
BRITISH AIRWAYS

SharePoint

You can access the Transaction Log Data in an easier to read/navigate formate than on the app via the SharePoint site.

BRITISH AIRWAYS

SharePoint



You can access each airlines inventory to view all information, such as.

- Part Number
- Description
- Store Location
- Seriel Number
- Quantity
- Time Expiration
- Comments
- Created - Date Receipted
- Created by - By Whom

BRITISH AIRWAYS

Thank You for Your Commitment!

To conclude the User Guide, I want to extend a heartfelt thanks to each member of our team accross British Airways for your dedication to integrating this innovative application into our daily operations. Your efforts in utilizing this app are not just a contribution to our inventory management system but a significant step towards enhancing our operational efficiency and accuracy.

BRITISH AIRWAYS

Why Your Participation is Vital

- **Keeping Our Data Current:** Your consistent use of the app ensures that our inventory data is always up-to-date. This real-time data accuracy is crucial for making informed decisions and maintaining optimal inventory levels.
- **Streamlined Inventory Management:** By using the app, you're contributing to a more organized and efficient inventory system. This streamlined process significantly reduces the time and effort required to locate and manage inventory items.
- **Accuracy at Your Fingertips:** With your help, the app provides precise and reliable data about the location and details of our inventory items. This accuracy is key to preventing errors and enhancing our overall productivity.

BRITISH AIRWAYS

7. Stores Keeper Mobile App Technical Guide

1. Login - User Authentication: The process begins with user authentication, using BA's pre-established security network.
2. Access Outcomes:
    - If the user is authenticated successfully, access to the main menu of the app is granted. Their profile can be seen on the main menu screen.
    - Access Denied: If authentication fails, access is denied, ensuring the security of the system.
3. Main Menu: Once access is granted, the user arrives at the main menu, which serves as the central hub for various functionalities of the app.
4. Function Choices:
    - Quick Find: This option allows users to quickly find inventory items.
        - The Quick Find allows a search across airline's databases according to the fields Part Number, Description, and Inventory Management. When all 3 fields are completed, the resulting parts in inventory are displayed in a table below that comprises all available information for the data entries selected (including batch/serial number, location, quantity available, expiry date, etc.). As the user types in any of the 3 search fields, options in the drop-down button are narrowed down. The search is sequential between fields, and a toggle allows to select with which field to start searching, allowing a combination of the 3 fields in any order desired by the user (i.e., Part Number → Description → Airline, Description → Part Number → Airline, Airline → Description → Part Number). Multiple items can be outputted to the table, allowing, for instance, comparison between several entries or displaying of the multiple parts needed to complete a task in a "shopping-list" fashion.
    - Transaction Processing and Inventory Management – Airline Inventory: Users first select the airline, viewing a table of complete entries for said airline. From this page, users can navigate to pages dedicated to transactions. Three transaction pages are available for each airline.
        - IN – dedicated to receiving new parts into stores.
        - OUT – dedicated to removing used parts from stores when parts are fitted on the aircraft.
        - TRANSFER – dedicated to placing an item in a new location in stores, including the unserviceable rack.
    - Data Analytics – Transaction Log: This section provides oversite of airline inventory transaction, displaying each transaction made through the application, in terms of the type of transaction, the airline concerned, the items identifier characteristics, the user responsible for the transaction and a timestamp. It is a key feature for transparency, accountability, analytical insights into inventory data, such as usage patterns, stock levels, and order histories. It also is the gateway to transacting with the blockchain.

Rules in the app front end are aimed to prevent errors when users perform transactions. As such, they are mainly implemented in the IN, OUT and TRANSFER pages within each Airline section.

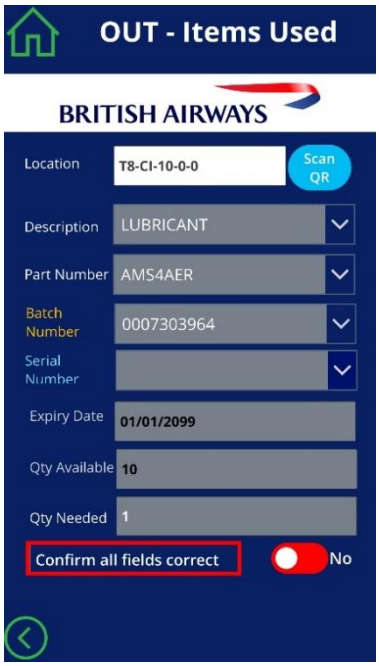
For the IN page, functions and rules are:

- Part Number must be typed for remaining fields to be displayed.
- If an entry with the same Part Number typed already exists in the airline's database, the Description field is autocompleted and cannot be manually altered.
- If the "Item has a Serial Number" toggle is set to yes, the Serial Number field must be completed, otherwise the field turns red when the user sets the "Confirm all fields correct" toggle to yes. Subsequently, when the user clicks on the red field, a specific notification is triggered, urging the user to complete the blank field.
- If the "Item has a Serial Number" toggle is set to no, the serial number field underneath is replaced by a Batch Number field and an equivalent behavior as described above occurs if the field is left blank.
- If the "Item has an Expiry Date" toggle is set to yes, the date picker field underneath appears, allowing for a date to be selected, not typed. If the date selected is prior to Today +1 day, when the user sets the "Confirm all fields correct" toggle to yes, the date picker turns red and, upon clicking sends a notification urging the user to pick an appropriate date. Conversely, if the Item has an Expiry Date" toggle is set to yes, but no date has been selected, the date picker field also turns red when the user sets the "Confirm all fields correct" toggle to yes and upon clicking a notification urging the user to pick a date is sent.
- Quantity must be typed as a number between 1 and 10'000, not starting with a 0. Failure to complete the field adequately triggers the field to turn red and sends specific notification messages when the "Confirm all fields correct" toggle is set to yes.
- Only if an item with identical specifications is present in the airline's database, a suggested location is displayed at the bottom of the page. Otherwise, this field is masked.
- A location can be either scanned, via a QR code signage present at the stores above each possible location or typed. Either way, the location must belong to the specific airline. If the location field is left blank or filled with a location that does not belong to the airline, the location field turns red, and a specific error notification is received when the "Confirm all fields correct" toggle is set to yes. If a location was suggested and the location provided is different, a notification will be sent enquiring if the user is sure. This rule is not, however, mandatory, as a distinct location can be saved if it belongs to the correct airline.
- If the "Confirm all fields correct" toggle is set to yes but any of the above-mentioned rules fail to be true, the "Confirm all fields correct" toggle automatically resets itself to no. When all fields are eventually correct, the toggle remains at yes and the fields can no longer be edited, taking a view only form. To edit any of the fields again, the user can reset the toggle to no, manually. Lastly, only after the "Confirm all fields correct" toggle is set to yes and remains at yes, does the save button at the bottom of the page become visible.

When the user presses Save the airline's database is updated, either adding the stated quantity to a previously existing record that matches all the information provided exactly or by creating a new record. In addition, an entry is created in the log database, stating the type of transaction, the parts specifications, the user responsible for the transaction, the user's company email and a timestamp.

- A slight variation in rules and forms exists for BA versus all other customer airlines. In fact, for BA a Batch Number is always mandatory, whilst a Serial Number is optional, thus toggles and fields in the IN page are not mutually exclusive. For customer airlines, Batch and Serial Number are mutually exclusive and as such both cannot be recorded when transacting IN.

For the OUT page, functions and rules are:

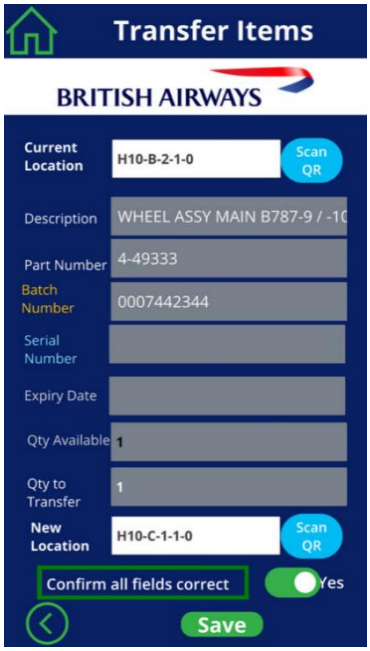


- A QR coded location for the part must be scanned in the stores or typed. If the location does not match the airline or is empty the field will turn red and send specific notifications when the "Confirm all fields correct" toggle is set to yes.
- All fields below, from Description to Quantity Available autocomplete according to the location selected. If multiple entries are present in the airline's database at the specified location, the Part Number, Serial Number and Batch Number fields work as sequential dropdowns, narrowing subsequent choices depending on the previous selections.
- Expiry date and Quantity Available are view only fields, reflecting the chosen item's characteristics as saved in the database. As Serial and Batch Number are not both required, an empty field will be displayed as N/A.
- The Quantity Needed field is only made visible after the Quantity Available field autocomplete. Furthermore, it only accepts numbers up to 5 digits that cannot start with 0 and cannot be superior to the Quantity Available. If any of these rules are broken, the field will turn red, and a specific notification sent when the "Confirm all fields correct" toggle is set to yes.
- The "Confirm all fields correct" toggle is only made visible after Quantity Needed is typed. If the "Confirm all fields correct" toggle is set to yes but any of the above-mentioned rules fail to be true, the "Confirm all fields correct" toggle automatically resets itself to no. When all fields are eventually correct, the toggle remains at yes and the fields can no longer be edited, taking a view only form. To edit any of the fields again, the user can reset the toggle to no, manually. Lastly, only after the "Confirm all fields correct" toggle is set to yes and remains at yes, does the save button at the bottom of the page become visible.
- When the user presses Save the airline's database is updated, by subtracting the Quantity Needed from the Quantity Available. A record is never deleted, even if quantities go to 0. In addition, an entry is created in the log database, stating the type

of transaction, the parts specifications, the user responsible for the transaction, the user's company email and a timestamp.

Regarding the TRANSFER page, functions and rules are:

- A QR coded location for the Current Location of the part to transfer must be scanned in the stores or typed. If the location does not match the airline or is empty the field will turn red and send specific notifications when the "Confirm all fields correct" toggle is set to yes.
- All fields below, from Description to Quantity Available autocomplete according to the location selected. If multiple entries are present in the airline's database at the specified location, the Part Number, Serial Number and Batch Number fields work as sequential dropdowns, narrowing subsequent choices depending on the previous selections.
- Expiry date and Quantity Available are view only fields, reflecting the chosen item's characteristics as saved in the database.
- The Quantity to Transfer field is only made visible after the Quantity Available field autocomplete. Furthermore, it only accepts numbers up to 5 digits that cannot start with 0 and cannot be superior to the Quantity Available. If any of these rules are broken, the field will turn red, and a specific notification sent when the "Confirm all fields correct" toggle is set to yes.
- Likewise, the New Location field is only made visible after the Quantity Available field autocompletes, signaling a unique part has been selected. A QR coded location for the New Location of the part to transfer must be scanned in the stores or typed. If the location does not match the airline the field will turn red and send a specific notification when the "Confirm all fields correct" toggle is set to yes.
- The "Confirm all fields correct" toggle is only made visible after New Location has been either scanned or typed. If the "Confirm all fields correct" toggle is set to yes but any of the above-mentioned rules fail to be true, the "Confirm all fields correct" toggle automatically resets itself to no. When all fields are eventually correct, the toggle remains at yes and the fields can no longer be edited, taking a view only form. To edit any of the fields again, the user can reset the toggle to no, manually. Lastly, only after the "Confirm all fields correct" toggle is set to yes and remains at yes, does the save button at the bottom of the page become visible.



- When the user presses Save the airline's database is updated, by subtracting the Quantity to Transfer from the Quantity Available at the Current Location, and by either adding the Quantity to Transfer to the Quantity of an entry matching the part specifications and the New Location if it already exists in the airline's database, or by creating a new entry if it doesn't already exist. A record is never deleted, even if quantities go to 0. In addition, two entries are created in the log database, each stating

the type of transaction (TRANSFER – OUT and TRANSFER – IN), the parts specifications (with Current Location for the TRANSFER – OUT log entry and with the New Location for the TRANSFER – IN log entry), the user responsible for the transaction, the user's company email and a timestamp.

In general, blank fields that are allowed to be left blank, are saved as N/A in the backend, and for Expiry Date, saved as 01/01/2099, to avoid formatting conflicts. No fields are ever completely blank in the backend.

8. Stores Keeper Solution Code Links

a. Mobile Application

The updated code is available below: [andrewibsen/Stores_Keeper at storeskeeperjfk (github.com)]

b. Smart Contract

The contract is deployed on the Sepolia testnet at the following address:
[Contract Address 0xf28e2c1c3ec439dc7ead7eddb4cc12c7a00fb20a | Etherscan]
The code of the deployed contract is as below, with explanations embedded in green.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SupplyChain {

    struct Part {
        string partName;
        string partNumber;
        string description;
        bool doesHaveSerialNumber;
        bool doesHaveExpiry;
        bool doesHaveBatchNumber;
        uint256 quantity;
        string location;
        string serialNumber;
        string expiryDate;
        string batchNumber;
    }

    struct AirlineData {
        mapping(string => Part) catalog;
        mapping(string => uint) orderRegistry;
        mapping(uint => string) ledger;
        string[] partNumbers;
        uint transactionCounter;  // Transaction counter for each airline
    }

    // Mapping to store airline-specific data
    mapping(address => AirlineData) airlines;
```

```
    mapping(address => bool) whitelisted;

    address public owner;

    // Events to log different stages of the supply chain
    event PartReceived(uint transactionId, string partNumber, uint quantity,
address buyer);
    event PartTransferred(uint transactionId, string partNumber, uint
quantity, address sender, address receiver);
    event PartDispatched(uint transactionId, string partNumber, uint quantity,
address sender);

    constructor () {
      owner = msg.sender;
      whitelisted[msg.sender] = true;
    }

    // Function to record an order in the order registry, catalog, and ledger
    function recordOrder(
        string memory partName,
        string memory partNumber,
        string memory description,
        bool doesHaveSerialNumber,
        bool doesHaveExpiry,
        bool doesHaveBatchNumber,
        uint256 quantity,
        string memory location,
        string memory serialNumber,
        string memory expiryDate,
        string memory batchNumber
    ) external {
        require(quantity > 0, "Quantity must be greater than zero");
        require(whitelisted[msg.sender], "The address is not authorized");

        AirlineData storage airline = airlines[msg.sender];

        if (doesHaveSerialNumber) {
            require(bytes(serialNumber).length > 0, "Serial number is required
for this part");
        }
        if(doesHaveExpiry) {
            require(bytes(expiryDate).length > 0, "Expiry date is required for
this part");
        }
        if(doesHaveBatchNumber) {
            require(bytes(batchNumber).length > 0, "Batch number is required
for this part");
        }

        // Increment the transaction counter and use it as the transaction ID
```

```
        uint transactionId = airline.transactionCounter++;

        // Add part to catalog if it doesn't exist
        if (bytes(airline.catalog[partNumber].partName).length == 0) {
            airline.catalog[partNumber] = Part(
                partName,
                partNumber,
                description,
                doesHaveSerialNumber,
                doesHaveExpiry,
                doesHaveBatchNumber,
                quantity,
                location,
                serialNumber,
                expiryDate,
                batchNumber
            );

            airline.partNumbers.push(partNumber);
        } else {
            airline.catalog[partNumber].quantity += quantity;
            airline.catalog[partNumber].location = location;
        }

        // Add order to order registry
        airline.orderRegistry[partNumber] = quantity;

        // Add entry to the ledger
        airline.ledger[transactionId] = "IN";

        emit PartReceived(transactionId, partNumber, quantity, msg.sender);
    }

    // Function to transfer a part from one airline to another
    function transferPart(
        address senderAirline,
        address receiverAirline,
        string memory partNumber,
        uint256 quantity,
        string memory newLocation
    ) external {
        require(whitelisted[msg.sender], "The address is not authorized");

        AirlineData storage sender = airlines[senderAirline];
        AirlineData storage receiver = airlines[receiverAirline];

        require(bytes(sender.catalog[partNumber].partName).length > 0, "Part
does not exist in sender's catalog");
        require(sender.orderRegistry[partNumber] >= quantity, "Insufficient
quantity to transfer");
```

```
        // Increment the transaction counter for the sender and use it as the
transaction ID
        uint transactionId = sender.transactionCounter++;

        // Update sender's catalog
        sender.catalog[partNumber].quantity -= quantity;

        // Update receiver's catalog
        if (bytes(receiver.catalog[partNumber].partName).length == 0) {
            receiver.catalog[partNumber] = Part(
                sender.catalog[partNumber].partName,
                sender.catalog[partNumber].partNumber,
                sender.catalog[partNumber].description,
                sender.catalog[partNumber].doesHaveSerialNumber,
                sender.catalog[partNumber].doesHaveExpiry,
                sender.catalog[partNumber].doesHaveBatchNumber,
                quantity,
                newLocation,
                sender.catalog[partNumber].serialNumber,
                sender.catalog[partNumber].expiryDate,
                sender.catalog[partNumber].batchNumber
            );

            receiver.partNumbers.push(partNumber);
        } else {
            receiver.catalog[partNumber].quantity += quantity;
            receiver.catalog[partNumber].location = newLocation;
        }

        // Update order registry for sender
        sender.orderRegistry[partNumber] -= quantity;

        // Add entry to the ledger for sender
        sender.ledger[transactionId] = "OUT";

        // Add entry to the ledger for receiver
        receiver.ledger[transactionId] = "IN";

        emit PartTransferred(transactionId, partNumber, quantity,
senderAirline, receiverAirline);
    }

    function authorizeAirline(address airlineAddress) external {
      require(msg.sender == owner, "Only owner can authorize");
      whitelisted[airlineAddress] = true;
    }

    // Function to check if data exists for a specific part number for an
airline
```

```
    function doesPartExist(address airline, string memory partNumber) external
view returns (bool) {
        Part memory part = airlines[airline].catalog[partNumber];
        return bytes(part.partName).length > 0;
    }

    // Function to get information about a specific part for an airline
    function getPartInfo(address airline, string memory partNumber) external
view returns (Part memory) {
        return airlines[airline].catalog[partNumber];
    }

    // Function to check if an order exists for a specific transaction ID for
an airline
    function doesOrderExist(address airline, string memory partNumber)
external view returns (bool) {
        uint quantity = airlines[airline].orderRegistry[partNumber];
        return quantity > 0;
    }

    // Function to get information about a specific order for an airline
    function getOrderInfo(address airline, string memory partNumber) external
view returns (uint) {
        return airlines[airline].orderRegistry[partNumber];
    }

    // Function to check if a ledger entry exists for a specific transaction
ID for an airline
    function doesLedgerEntryExist(address airline, uint transactionId)
external view returns (bool) {
        string memory entry = airlines[airline].ledger[transactionId];
        return bytes(entry).length > 0;
    }

    // Function to get information about a specific ledger entry for an
airline
    function getLedgerEntry(address airline, uint transactionId) external view
returns (string memory) {
        return airlines[airline].ledger[transactionId];
    }
}
```

c. Blockchain Component

The updated code is available below: [andrewibsen/Stores_Keeper at storeskeeperjfk (github.com)]

This is a Python 4.0. script designed for interacting with Ethereum blockchain via the Smart Contract described above. It is commented script divided into sections for readability.

Import Libraries

```
1. import numpy as np
```

```
2. import pandas as pd
3. import asyncio
4. import json
5. import requests
6. from web3 import Web3
7. from websockets import connect
8. import numpy as Np
```

Explanation: Libraries necessary for data manipulation, asynchronous operations, interacting with web services, and blockchain interactions are imported.

Connection Setup

```
1. #Connect to an Infura Node
2. infura_ws_url =
   "wss://sepolia.infura.io/ws/v3/616ee51a641b4517828ea542868adfea"
3. infura_http_url =
   "https://sepolia.infura.io/v3/616ee51a641b4517828ea542868adfea"
4. w3 = Web3(Web3.HTTPProvider(infura_http_url))
5. w3.is_connected()
```

Explanation: Establishes a connection to the Ethereum network via Infura. Checks if the connection is successful.

Smart Contract Interaction

```
1. #initialize the smart contract
2. my_contract=
   w3.eth.contract(address="0xf28E2c1c3ec439dC7EaD7Eddb4cC12C7a00FB20a", abi=abi)
3. #primt contract address if properly initialized
4. print(my_contract)
5. #print contract functions if properly initialized
6. my_contract.functions.abi
```

Explanation: Loads the for the smart contract and initializes it for interaction using a contract address. The ABI.txt file needs to be uploaded into the script's environment and the path provided.

Transaction Function

```
1. def transact_loglists( Log,Acc_Dict):
2.   Hxs=[]
3.   PartInfo=[]
4.   #Log: lists of lists where each column corresponds to a
   transaction in the sharepoint log, plus booleans and partNumber
```

```
   for smart contract functions, in order of the smartcontract
   function arguments
5.   #Acc_Dict: dictionary where keys are the airlines abbreviations
   and values are the airlines wallet accounts (!not their
   addressesm, there account rom key)
6.   for i in range(0, len(Log[1])):
7.       # Inputs items to the sender:airline catalog
8.       # Check that previous transaction was properly sent before
   moving on, otherwise nonce will be too low and gas misestimated
9.       if len(Hxs)>0:
10.              w3.eth.wait_for_transaction_receipt(Hxs[-1])
11.              print('proceeding to i = %s'%i)
12.           if Log[0][i] == "IN":
13.             acc = Acc_Dict[Log[1][i]]
14.             ins =
   my_contract.functions.recordOrder(str(Log[2][i]), str(Log[3][i]),
   str(Log[4][i]), Log[5][i], Log[6][i], Log[7][i],Log[8][i],
   str(Log[9][i]),str(Log[10][i]),str(Log[11][i]),str(Log[12][i])).b
   uild_transaction({
15.             "from": acc.address,
16.             "nonce": w3.eth.get_transaction_count(acc.address),
17.           })
18.             signed_tx = w3.eth.account.sign_transaction(ins,
   private_key=acc.key)
19.             #try:
20.             #send transaction and save Hx in list
21.             Hxs.append(w3.eth.send_raw_transaction(signed_tx.raw
   Transaction))
22.             #except Exception as e: print(e) -> removed this
   because the transactions need to be sequential otherwise
   conflicts arise
23.             #check transaction and save info in list
24.             PartInfo.append([x for x in
   my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
25.             if len(Hxs)>0:
26.               w3.eth.wait_for_transaction_receipt(Hxs[-1])
27.             #try:
28.             PartInfo.append([x for x in
   my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
29.
30.      # Send items transferred out from the airline to the
   receiver:stores
31.           if Log[0][i] == "TRANSFER-OUT":
32.             acc_st = Acc_Dict["ST"]
33.             acc = Acc_Dict[Log[1][i]]
34.             ins =
   my_contract.functions.transferPart(acc.address,acc_st.address,Log
   [3][i],Log[8][i],str(Log[9][i])).build_transaction({
35.             "from": acc.address,
36.             "nonce":  w3.eth.get_transaction_count(acc.address),
```

```
            })

              signed_tx = w3.eth.account.sign_transaction(ins,
  private_key=acc.key)
              #try:
              Hxs.append(w3.eth.send_raw_transaction(signed_tx.raw
  Transaction))
              #except Exception as e: print(e)
              #check transaction and save info in list
              PartInfo.append([x for x in
  my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
              if len(Hxs)>0:
                w3.eth.wait_for_transaction_receipt(Hxs[-1])
              #try:
              PartInfo.append([x for x in
  my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
            # Send items transferred in to the airline catalog
            if Log[0][i] == "TRANSFER-IN":
              acc = Acc_Dict[Log[1][i]]
              ins =
  my_contract.functions.recordOrder(str(Log[2][i]), str(Log[3][i]),
  str(Log[4][i]),Log[5][i],Log[6][i], Log[7][i],Log[8][i],
  str(Log[9][i]),str(Log[10][i]),str(Log[11][i]),str(Log[12][i])).b
  uild_transaction({
              "from": acc.address,
              "nonce": w3.eth.get_transaction_count(acc.address),
            })

              signed_tx = w3.eth.account.sign_transaction(ins,
  private_key=acc.key)

              #send transaction and save Hx in list
              #try:
              Hxs.append(w3.eth.send_raw_transaction(signed_tx.raw
  Transaction))
              #except Exception as e: print(e)
              #check transaction and save info in list
              PartInfo.append([x for x in
  my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
              if len(Hxs)>0:
                w3.eth.wait_for_transaction_receipt(Hxs[-1])
              #try:
              PartInfo.append([x for x in
  my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
      #Send items from the airline catalog to the receiver=stores

            if Log[0][i] == "OUT":
              acc_st = Acc_Dict["ST"]
              acc = Acc_Dict[Log[1][i]]
```

```
                ins = my_contract.functions.transferPart(acc.address,acc_st.address,str(Log[3][i]),Log[8][i],str(Log[9][i])).build_transaction({
                "from": acc.address,
                "nonce":  w3.eth.get_transaction_count(acc.address),
            })

                signed_tx = w3.eth.account.sign_transaction(ins, private_key=acc.key)

                #send transaction and save Hx in list
                if len(Hxs)>0:
                  w3.eth.wait_for_transaction_receipt(Hxs[-1])
                #try:
                Hxs.append(w3.eth.send_raw_transaction(signed_tx.rawTransaction))
                #except Exception as e: print(e)
                #check transaction and save info in list
                PartInfo.append([x for x in my_contract.functions.getPartInfo(acc.address,Log[3][i]).call()])
            print('%s completed'%Log[0][i])
        return Hxs, PartInfo
```

Explanation: Defines a function to process transactions based on log entries. This includes interacting with the smart contract functions.

Initialize the Solution

```
# Now automate function according to log entry
# Create a dictionary of Airline"s accounts
Acc_Dict= {
"EI":  w3.eth.account.from_key("private key"),
"ANZ":  w3.eth.account.from_key("private key"),
"BA":  w3.eth.account.from_key("private key"),
"ST": w3.eth.account.from_key("private key"),
"QF": w3.eth.account.from_key("private key"),
"ANZ":w3.eth.account.from_key("private key"),
     "ANA":w3.eth.account.from_key("private key")}
```

```
# The following steps only pertain to the first use of the code
#Authorize each airline to transact
list_airline=['EI','QF','BA','ANZ', 'IB', 'ANA','ST']
for airline in list_airline:
auth=my_contract.funct,ions.authorizeAirline(Acc_Dict["IB"].address).build_transaction({
"from": Acc_Dict["EI"].address,
"nonce": w3.eth.get_transaction_count(Acc_Dict["EI"].address),
"gasPrice": 35631095853
})
     signed_tx = w3.eth.account.sign_transaction(auth, private_key=Acc_Dict["EI"].key)
```

```
11.      w3.eth.send_raw_transaction(signed_tx.rawTransaction)
```

```
1. # The following steps only pertain to the first use of the code
2. # Read each airline's catalog and transact it
3. for idx, airline in enumerate(list_airline):
4. stores=pd.read_csv("/content/"+'%S'%airline+' Stores Data JFK
   .csv')
5. index_names=stores[stores['Quantity']<=0].index
6. stores.drop(index_names, inplace = True)
7. stores.reset_index(drop=True, inplace=True)
8. Transaction_type1=np.repeat("IN", len(BA_stores1))
9. Airline1=np.repeat(airline, len(stores))
10. Part_Name1=stores['Part_Number'].to_list()
11. Description1=stores['Description'].to_list()
12. JFK_Location1 = stores['JFK_Location'].to_list()
13. Quantity1=[int(x) for x in stores['Quantity']]
14. Serial_Number1=stores['Serial_Number'].to_list()
15. Batch_Number1=stores['Batch_Number'].to_list()
16. Time_Expiration1=stores['Time_Expiration'].to_list()
17. #Create a Unique Identifier for each part == tokenization
18. Part_Number=[Airline1[x]+str(Part_Name1[x])+str(Description1[x])+
   JFK_Location1[x]+str(Serial_Number1[x])+str(Batch_Number1[x])+str
   (Time_Expiration1[x]) for x in range(len(Transaction_type1))]
19. Create 3 boolean lists to match smart contract variables
20. doesHaveSerialNumber=[False if Serial_Number1[x] =="NaN" else
   True for x in range(len(Transaction_type1)) ]
21. doesHaveExpiry=[False if Time_Expiration1[x] =="1/1/2099" else
   True for x in range(len(Transaction_type1)) ]
22. doesHaveBatchNumber=[False if Batch_Number1[x] =="NaN" else True
   for x in range(len(Transaction_type1)) ]
23. Log1=[Transaction_type1,Airline1,Part_Name1,
   Part_Number,Description1,doesHaveSerialNumber,doesHaveExpiry,does
   HaveBatchNumber,Quantity1,JFK_Location1,Serial_Number1,Time_Expir
   ation1,Batch_Number1]
24. print("start transacting for %S"%airline)
25. # upload all of ba's data to the blockchain
26. Hxs, PartInfo=transact_loglists( Log1,Acc_Dict)
```

Explanation: Stores all the airline's wallet addresses and key to easily sign future transactions. Authorizes new airline(s) wallet(s) to interact with the custom smart contract. Updates the smart contract with the new airline(s) catalog(s). This section only needs to be read when adding new airline(s) to the solution.

Transact the Log

```
1. # Now read the log and transact the log
2. Log_file=pd.read_csv("/content/log_file.csv") #replace by the
   path to log file
3. Log_file['Quantity']=[0 if np.isnan(x) else int(x) for x in
   Log_file['Quantity']]
4. index_names=Log_file[Log_file['Quantity']<=0].index
```

```
Log_file.drop(index_names, inplace = True)
Log_file.reset_index(drop=True, inplace=True)
Log_file['JFK_Location']=["0" if x=="nan" else str(x) for x in Log_file['JFK_Location']]
index_names1=Log_file[Log_file['JFK_Location']=="0"].index
Log_file.drop(index_names1, inplace = True)
Log_file.reset_index(drop=True, inplace=True)
Transaction_type1=Log_file['Transaction_type'].to_list()
Airline1=Log_file['Airline'].to_list()
Part_Name1=Log_file['Part_Number'].to_list()
Description1=Log_file['Description'].to_list()
JFK_Location1 = Log_file['JFK_Location'].to_list()
Quantity1=[int(x) for x in Log_file['Quantity']]
Serial_Number1=Log_file['Serial_Number'].to_list()
Batch_Number1=Log_file['Batch_Number'].to_list()
Time_Expiration1=Log_file['Time_Expiration'].to_list()
Part_Number=[Airline1[x]+str(Part_Name1[x])+str(Description1[x])+str(JFK_Location1[x])+str(Serial_Number1[x])+str(Batch_Number1[x])+str(Time_Expiration1[x]) for x in range(len(Transaction_type1))]
# Create 3 boolean lists for
doesHaveSerialNumber=[False if Serial_Number1[x] =="NaN" else True for x in range(len(Transaction_type1)) ]
doesHaveExpiry=[False if Time_Expiration1[x] =="1/1/2099" else True for x in range(len(Transaction_type1)) ]
doesHaveBatchNumber=[False if Batch_Number1[x] =="NaN" else True for x in range(len(Transaction_type1)) ]
#Get the list of lists ready
Log1=[]
Log1=[Transaction_type1,Airline1,Part_Name1,Part_Number,Description1,doesHaveSerialNumber,doesHaveExpiry,doesHaveBatchNumber,Quantity1,JFK_Location1,Serial_Number1,Time_Expiration1,Batch_Number1]
```

Explanation: Reads the Log file and transacts it line by line. The Log.csv file must be uploaded to the code' environment and the path provided. The airline wallet keys must have been previously saved (see initializing function). The output is a list of Hexes and a list of Part Information, retrieved from the smart contract to verify the information was safely stored.

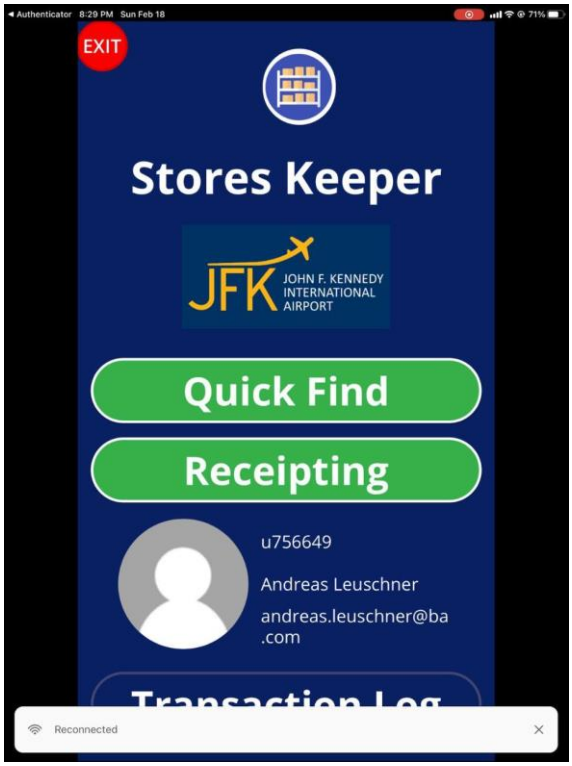
9. Mobile Application Experiment

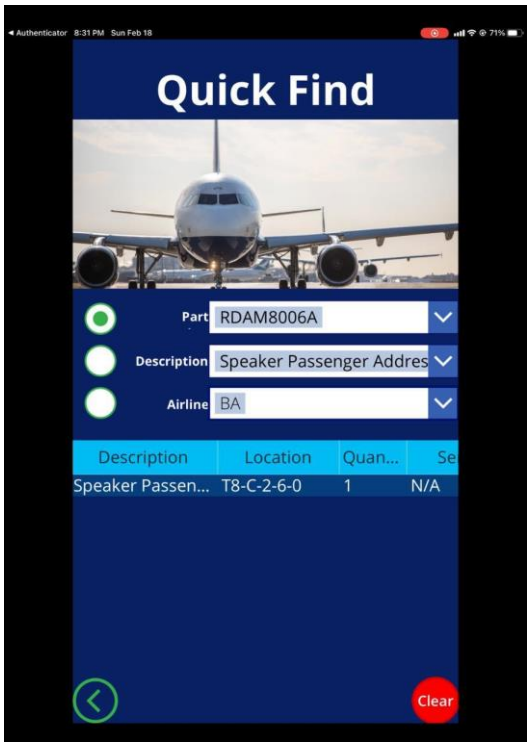
a. Experimental Protocol

Firstly, each participant undergoes a one-on-one training session in which (1) they are provided with the user guide and given 2-5 mins to explore it by themselves, (2) functionalities are demonstrated, (3) the user is invited to test the application's functionalities. The training is completed in around 20 minutes and participants are encouraged to ask questions throughout. The training is performed either on a company provided Apple Ipad, used in day-to-day operations, or with the employee's preferred mobile device, in BYOD situation. Training and testing are performed on the same device.

Secondly, immediately after the training, participants are required to employ the functionalities of the application for pre-determined parts previously selected at random, while the screen is recorded. This can be done on their own preferred device, provided screen recording is available, otherwise the company Apple Ipad is employed. As a comparison, participants then repeat the same inventory transactions on the analog BA system, by hand-writing all the relevant part information in the store's book. The accuracy of these entries and the time taken to complete them is measured. The test includes 4 randomly selected part for each user in a randomly selected airline's inventory, and for each of these parts, 3 transactions (1 in, 1 out, 1 transfer, 1 quick find), a total of 12 transactions, conducted via both systems, amounting to 24 transactions overall. The total duration of the test is circa 30 minutes. Participants are allowed breaks as needed. All communications between the examiner and the subject during the experiment follow a pre-determined script, as provided in annex.

Thirdly at the end of the experiment, qualitative feedback is collected within the application, via a survey presented when exiting the application. The survey includes 5 open and closed questions.





b. Results

Sbj	Accuracy		Time		Transaction	Role
	Analog	App	Analog	App		
S1	80	90	0.5	1	IN	Mechanic
S1	80	70	0.5	1	IN	Mechanic
S1	70	70	0.5	1	IN	Mechanic
S1	60	60	0.5	1	Transfer	Mechanic
S1	80	60	0.5	1	Transfer	Mechanic
S1	70	50	1	1	Transfer	Mechanic
S1	60	40	0.5	1	Out	Mechanic
S1	60	50	1	1	Out	Mechanic
S1	60	60	0	1	Out	Mechanic
S1	80	15	1	1	Quick Find	Mechanic
S1	70	15	1	1	Quick Find	Mechanic
S1	60	15	1	1	Quick Find	Mechanic
S2	60	80	1	1	IN	Mechanic
S2	90	90	1	1	IN	Mechanic
S2	80	70	1	1	IN	Mechanic
S2	90	60	1	1	Transfer	Mechanic
S2	60	60	1	1	Transfer	Mechanic
S2	70	60	0.5	1	Transfer	Mechanic
S2	80	60	0.5	1	Out	Mechanic
S2	60	40	1	1	Out	Mechanic
S2	90	50	1	1	Out	Mechanic
S2	60	20	1	1	Quick Find	Mechanic
S2	60	20	1	1	Quick Find	Mechanic
S2	70	20	1	1	Quick Find	Mechanic
S3	80	90	0.5	1	IN	Mechanic
S3	90	70	0.5	1	IN	Mechanic
S3	80	80	0.5	1	IN	Mechanic
S3	70	60	0.5	1	Transfer	Mechanic
S3	60	60	0.5	1	Transfer	Mechanic
S3	60	50	0	1	Transfer	Mechanic
S3	70	40	0.5	1	Out	Mechanic
S3	90	50	0.5	1	Out	Mechanic
S3	90	40	0.5	1	Out	Mechanic
S3	60	20	1	1	Quick Find	Mechanic
S3	60	20	0	1	Quick Find	Mechanic
S3	50	15	1	1	Quick Find	Mechanic
S4	60	60	1	1	IN	Mechanic
S4	70	80	1	1	IN	Mechanic
S4	80	70	0.5	1	IN	Mechanic
S4	60	40	0.5	1	Transfer	Mechanic
S4	60	60	0.5	1	Transfer	Mechanic
S4	60	50	0.5	1	Transfer	Mechanic
S4	60	60	0.5	1	Out	Mechanic
S4	60	60	1	1	Out	Mechanic

S4	60	50	1	1	Out	Mechanic
S4	50	15	0	1	Quick Find	Mechanic
S4	70	20	1	1	Quick Find	Mechanic
S4	80	15	1	1	Quick Find	Mechanic
S5	60	80	1	1	IN	Maintenance Rep
S5	70	80	1	1	IN	Maintenance Rep
S5	70	60	1	1	IN	Maintenance Rep
S5	50	70	0.5	1	Transfer	Maintenance Rep
S5	50	60	1	1	Transfer	Maintenance Rep
S5	70	60	1	1	Transfer	Maintenance Rep
S5	60	40	1	1	Out	Maintenance Rep
S5	60	40	1	1	Out	Maintenance Rep
S5	60	50	1	1	Out	Maintenance Rep
S5	60	15	1	1	Quick Find	Maintenance Rep
S5	60	15	1	1	Quick Find	Maintenance Rep
S5	60	15	1	1	Quick Find	Maintenance Rep
S6	80	70	1	1	IN	Manager
S6	90	90	1	1	IN	Manager
S6	80	60	0.5	1	IN	Manager
S6	80	60	0.5	1	Transfer	Manager
S6	80	60	1	1	Transfer	Manager
S6	80	50	1	1	Transfer	Manager
S6	80	40	1	1	Out	Manager
S6	80	50	1	1	Out	Manager
S6	80	40	1	1	Out	Manager
S6	60	20	1	1	Quick Find	Manager
S6	60	15	1	1	Quick Find	Manager
S6	60	20	1	1	Quick Find	Manager
S7	80	80	1	1	IN	Manager
S7	90	80	1	1	IN	Manager
S7	70	60	1	1	IN	Manager
S7	60	60	0	1	Transfer	Manager
S7	90	60	0.5	1	Transfer	Manager
S7	80	50	1	1	Transfer	Manager
S7	80	50	1	1	Out	Manager
S7	70	60	0.5	1	Out	Manager
S7	80	40	1	1	Out	Manager

S7	60	20	1	1	Quick Find	Manager
S7	60	15	1	1	Quick Find	Manager
S7	60	20	1	1	Quick Find	Manager
S8	80	60	1	1	IN	Manager
S8	80	80	1	1	IN	Manager
S8	80	70	1	1	IN	Manager
S8	60	60	1	1	Transfer	Manager
S8	60	60	1	1	Transfer	Manager
S8	80	60	1	1	Transfer	Manager
S8	60	60	1	1	Out	Manager
S8	70	60	1	1	Out	Manager
S8	80	60	1	1	Out	Manager
S8	60	15	1	1	Quick Find	Manager
S8	80	15	1	1	Quick Find	Manager
S8	80	30	1	1	Quick Find	Manager
S9	100	80	1	1	IN	Manager
S9	80	90	1	1	IN	Manager
S9	90	90	1	1	IN	Manager
S9	60	70	1	1	Transfer	Manager
S9	60	60	1	1	Transfer	Manager
S9	80	70	1	1	Transfer	Manager
S9	90	60	1	1	Out	Manager
S9	70	60	1	1	Out	Manager
S9	80	60	0.5	1	Out	Manager
S9	60	20	1	1	Quick Find	Manager
S9	80	30	1	1	Quick Find	Manager
S9	80	30	1	1	Quick Find	Manager
S10	60	90	0.5	1	IN	Manager
S10	60	70	1	1	IN	Manager
S10	70	70	1	1	IN	Manager
S10	100	80	0.5	1	Transfer	Manager
S10	100	60	1	1	Transfer	Manager
S10	90	80	1	1	Transfer	Manager
S10	50	40	0	1	Out	Manager
S10	60	50	1	1	Out	Manager
S10	40	60	0.5	1	Out	Manager
S10	50	15	1	1	Quick Find	Manager
S10	40	20	1	1	Quick Find	Manager
S10	60	15	1	1	Quick Find	Manager
S11	80	90	1	1	IN	Manager
S11	70	90	1	1	IN	Manager
S11	90	70	1	1	IN	Manager
S11	60	90	1	1	Transfer	Manager
S11	60	60	1	1	Transfer	Manager
S11	70	80	1	1	Transfer	Manager
S11	60	40	1	1	Out	Manager
S11	60	40	1	1	Out	Manager

S11	50	60	1	1	Out	Manager
S11	60	20	1	1	Quick Find	Manager
S11	60	20	1	1	Quick Find	Manager
S11	40	15	1	1	Quick Find	Manager
S12	80	90	1	1	IN	Manager
S12	70	80	1	1	IN	Manager
S12	90	70	1	1	IN	Manager
S12	60	90	1	1	Transfer	Manager
S12	90	60	1	1	Transfer	Manager
S12	60	80	1	1	Transfer	Manager
S12	100	60	0	1	Out	Manager
S12	50	40	1	1	Out	Manager
S12	50	50	1	1	Out	Manager
S12	60	15	1	1	Quick Find	Manager
S12	60	20	1	1	Quick Find	Manager
S12	40	15	1	1	Quick Find	Manager

General Linear Model

	20-FEB-2024 12:02:37
Data	C:\Users\Andrew\OneDrive - Andros Ventures\Desktop\Experiment_App_Data.sav
Active Dataset	DataSet6
Filter	<none>
Weight	<none>
Split File	<none>
N of Rows in Working Data File	144

10.

Within-Subjects Factors

Measure	App	Trans	Dependent Variable
Acc	1	1	ACC_OLD_IN
		2	ACC_OLD_OUT
		3	ACC_OLD_TRANS
		4	ACC_OLD_QF
	2	1	ACC_APP_IN
		2	ACC_APP_OUT
		3	ACC_APP_TRANS
		4	ACC_APP_QF
Time	1	1	TIME_OLD_IN
		2	TIME_OLD_OUT
		3	TIME_OLD_TRANS
		4	TIME_OLD_QF
	2	1	TIME_APP_IN
		2	TIME_APP_OUT
		3	TIME_APP_TRANS
		4	TIME_APP_QF

11.

Between-Subjects Factors

		N
ROLE	Maintenance Rep	3
	Manager	21
	Mechanic	12

12.

Descriptive Statistics

	ROLE	Mean	Std. Deviation	N
ACC_OLD_IN	Maintenance Rep	1.0000	.00000	3
	Manager	.8333	.32914	21
	Mechanic	.6667	.32567	12
	Total	.7917	.32459	36
ACC_OLD_OUT	Maintenance Rep	1.0000	.00000	3
	Manager	.8333	.32914	21
	Mechanic	.6667	.32567	12
	Total	.7917	.32459	36
ACC_OLD_TRANS	Maintenance Rep	.8333	.28868	3

	Manager	.8810	.26948	21
	Mechanic	.5833	.28868	12
	Total	.7778	.30342	36
ACC_OLD_QF	Maintenance Rep	1.0000	.00000	3
	Manager	1.0000	.00000	21
	Mechanic	.8333	.38925	12
	Total	.9444	.23231	36
ACC_APP_IN	Maintenance Rep	1.0000	.00000	3
	Manager	1.0000	.00000	21
	Mechanic	1.0000	.00000	12
	Total	1.0000	.00000	36
ACC_APP_OUT	Maintenance Rep	1.0000	.00000	3
	Manager	1.0000	.00000	21
	Mechanic	1.0000	.00000	12
	Total	1.0000	.00000	36
ACC_APP_TRANS	Maintenance Rep	1.0000	.00000	3
	Manager	1.0000	.00000	21
	Mechanic	1.0000	.00000	12
	Total	1.0000	.00000	36
ACC_APP_QF	Maintenance Rep	1.0000	.00000	3
	Manager	1.0000	.00000	21
	Mechanic	1.0000	.00000	12
	Total	1.0000	.00000	36
TIME_OLD_IN	Maintenance Rep	60.0000	.00000	3
	Manager	68.5714	15.58387	21
	Mechanic	70.0000	13.48400	12
	Total	68.3333	14.24279	36
TIME_OLD_OUT	Maintenance Rep	60.0000	.00000	3
	Manager	68.5714	15.58387	21
	Mechanic	70.0000	13.48400	12
	Total	68.3333	14.24279	36
TIME_OLD_TRANS	Maintenance Rep	56.6667	11.54701	3
	Manager	74.2857	14.34274	21
	Mechanic	66.6667	9.84732	12
	Total	70.2778	13.62479	36
TIME_OLD_QF	Maintenance Rep	60.0000	.00000	3
	Manager	60.4762	12.03170	21
	Mechanic	64.1667	9.96205	12
	Total	61.6667	10.82326	36
TIME_APP_IN	Maintenance Rep	43.3333	5.77350	3
	Manager	51.4286	9.10259	21
	Mechanic	50.0000	8.52803	12
	Total	50.2778	8.77858	36
TIME_APP_OUT	Maintenance Rep	43.3333	5.77350	3
	Manager	51.4286	9.10259	21
	Mechanic	50.0000	8.52803	12
	Total	50.2778	8.77858	36
TIME_APP_TRANS	Maintenance Rep	63.3333	5.77350	3
	Manager	66.6667	11.97219	21
	Mechanic	55.8333	6.68558	12
	Total	62.7778	11.11270	36
TIME_APP_QF	Maintenance Rep	15.0000	.00000	3
	Manager	19.2857	5.07093	21
	Mechanic	17.5000	2.61116	12
	Total	18.3333	4.30946	36

13.

Multivariate Tests[a]

Effect			Value	F	Hypothesis df	Error df
Between Subjects	Intercept	Pillai's Trace	.991	1830.370[b]	2.000	32.000
		Wilks' Lambda	.009	1830.370[b]	2.000	32.000
		Hotelling's Trace	114.398	1830.370[b]	2.000	32.000
		Roy's Largest Root	114.398	1830.370[b]	2.000	32.000
	ROLE	Pillai's Trace	.350	3.495	4.000	66.000
		Wilks' Lambda	.678	3.434[b]	4.000	64.000
		Hotelling's Trace	.435	3.370	4.000	62.000
		Roy's Largest Root	.300	4.942[c]	2.000	33.000
Within Subjects	App	Pillai's Trace	.718	40.669[b]	2.000	32.000
		Wilks' Lambda	.282	40.669[b]	2.000	32.000
		Hotelling's Trace	2.542	40.669[b]	2.000	32.000
		Roy's Largest Root	2.542	40.669[b]	2.000	32.000
	App * ROLE	Pillai's Trace	.241	2.259	4.000	66.000
		Wilks' Lambda	.759	2.360[b]	4.000	64.000
		Hotelling's Trace	.316	2.451	4.000	62.000
		Roy's Largest Root	.315	5.196[c]	2.000	33.000
	Trans	Pillai's Trace	.877	53.443[b]	4.000	30.000
		Wilks' Lambda	.123	53.443[b]	4.000	30.000
		Hotelling's Trace	7.126	53.443[b]	4.000	30.000
		Roy's Largest Root	7.126	53.443[b]	4.000	30.000
	Trans * ROLE	Pillai's Trace	.284	1.283	8.000	62.000
		Wilks' Lambda	.728	1.291[b]	8.000	60.000
		Hotelling's Trace	.358	1.296	8.000	58.000
		Roy's Largest Root	.304	2.357[c]	4.000	31.000
	App * Trans	Pillai's Trace	.736	20.926[b]	4.000	30.000
		Wilks' Lambda	.264	20.926[b]	4.000	30.000
		Hotelling's Trace	2.790	20.926[b]	4.000	30.000
		Roy's Largest Root	2.790	20.926[b]	4.000	30.000
	App * Trans * ROLE	Pillai's Trace	.119	.490	8.000	62.000
		Wilks' Lambda	.884	.476[b]	8.000	60.000
		Hotelling's Trace	.128	.462	8.000	58.000
		Roy's Largest Root	.088	.679[c]	4.000	31.000

14.

Multivariate Tests[a]

Effect			Sig.	Partial Eta Squared
Between Subjects	Intercept	Pillai's Trace	<.001	.991
		Wilks' Lambda	<.001	.991
		Hotelling's Trace	<.001	.991
		Roy's Largest Root	<.001	.991

	ROLE	Pillai's Trace	.012	.175
		Wilks' Lambda	.013	.177
		Hotelling's Trace	.015	.179
		Roy's Largest Root	.013	.230
Within Subjects	App	Pillai's Trace	<.001	.718
		Wilks' Lambda	<.001	.718
		Hotelling's Trace	<.001	.718
		Roy's Largest Root	<.001	.718
	App * ROLE	Pillai's Trace	.072	.120
		Wilks' Lambda	.063	.129
		Hotelling's Trace	.055	.137
		Roy's Largest Root	.011	.239
	Trans	Pillai's Trace	<.001	.877
		Wilks' Lambda	<.001	.877
		Hotelling's Trace	<.001	.877
		Roy's Largest Root	<.001	.877
	Trans * ROLE	Pillai's Trace	.269	.142
		Wilks' Lambda	.265	.147
		Hotelling's Trace	.264	.152
		Roy's Largest Root	.075	.233
	App * Trans	Pillai's Trace	<.001	.736
		Wilks' Lambda	<.001	.736
		Hotelling's Trace	<.001	.736
		Roy's Largest Root	<.001	.736
	App * Trans * ROLE	Pillai's Trace	.859	.059
		Wilks' Lambda	.868	.060
		Hotelling's Trace	.877	.060
		Roy's Largest Root	.612	.081

15.

a. Design: Intercept + ROLE
Within Subjects Design: App + Trans + App * Trans

b. Exact statistic
c. The statistic is an upper bound on F that yields a lower bound on the significance level.

16.

17.

Mauchly's Test of Sphericity[a]

Within Subjects Effect	Measure	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser
App	Acc	1.000	.000	0	.	1.000
	Time	1.000	.000	0	.	1.000
Trans	Acc	.000	.	5	.	.591
	Time	.000	.	5	.	.501
App * Trans	Acc	.000	.	5	.	.591
	Time	.000	.	5	.	.639

18.

Mauchly's Test of Sphericity[a]

		Epsilon	
Within Subjects Effect	Measure	Huynh-Feldt	Lower-bound
App	Acc	1.000	1.000
	Time	1.000	1.000
Trans	Acc	.660	.333
	Time	.551	.333

App * Trans	Acc	.660	.333
	Time	.719	.333

19.

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.[a]

a. Design: Intercept + ROLE
Within Subjects Design: App + Trans + App * Trans

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

20.

21.

Tests of Within-Subjects Effects

22.

Multivariate[a,b]

Within Subjects Effect		Value	F	Hypothesis df	Error df	Sig.
App	Pillai's Trace	.718	40.669[c]	2.000	32.000	<.001
	Wilks' Lambda	.282	40.669[c]	2.000	32.000	<.001
	Hotelling's Trace	2.542	40.669[c]	2.000	32.000	<.001
	Roy's Largest Root	2.542	40.669[c]	2.000	32.000	<.001
App * ROLE	Pillai's Trace	.241	2.259	4.000	66.000	.072
	Wilks' Lambda	.759	2.360[c]	4.000	64.000	.063
	Hotelling's Trace	.316	2.451	4.000	62.000	.055
	Roy's Largest Root	.315	5.196[d]	2.000	33.000	.011
Trans	Pillai's Trace	.625	14.987	6.000	198.000	<.001
	Wilks' Lambda	.376	20.608[c]	6.000	196.000	<.001
	Hotelling's Trace	1.658	26.804	6.000	194.000	<.001
	Roy's Largest Root	1.657	54.681[d]	3.000	99.000	<.001
Trans * ROLE	Pillai's Trace	.152	1.358	12.000	198.000	.189
	Wilks' Lambda	.850	1.378[c]	12.000	196.000	.179
	Hotelling's Trace	.173	1.398	12.000	194.000	.170
	Roy's Largest Root	.154	2.533[d]	6.000	99.000	.025
App * Trans	Pillai's Trace	.515	11.453	6.000	198.000	<.001
	Wilks' Lambda	.485	14.252[c]	6.000	196.000	<.001
	Hotelling's Trace	1.063	17.182	6.000	194.000	<.001
	Roy's Largest Root	1.063	35.069[d]	3.000	99.000	<.001
App * Trans * ROLE	Pillai's Trace	.060	.509	12.000	198.000	.908
	Wilks' Lambda	.941	.504[c]	12.000	196.000	.911
	Hotelling's Trace	.062	.499	12.000	194.000	.914
	Roy's Largest Root	.036	.590[d]	6.000	99.000	.738

23.

Multivariate[a,b]

Within Subjects Effect		Partial Eta Squared
App	Pillai's Trace	.718
	Wilks' Lambda	.718
	Hotelling's Trace	.718
	Roy's Largest Root	.718
App * ROLE	Pillai's Trace	.120

	Wilks' Lambda	.129
	Hotelling's Trace	.137
	Roy's Largest Root	.239
Trans	Pillai's Trace	.312
	Wilks' Lambda	.387
	Hotelling's Trace	.453
	Roy's Largest Root	.624
Trans * ROLE	Pillai's Trace	.076
	Wilks' Lambda	.078
	Hotelling's Trace	.080
	Roy's Largest Root	.133
App * Trans	Pillai's Trace	.258
	Wilks' Lambda	.304
	Hotelling's Trace	.347
	Roy's Largest Root	.515
App * Trans * ROLE	Pillai's Trace	.030
	Wilks' Lambda	.030
	Hotelling's Trace	.030
	Roy's Largest Root	.035

24.

a. Design: Intercept + ROLE
Within Subjects Design: App + Trans + App * Trans

b. Tests are based on averaged variables.
c. Exact statistic
d. The statistic is an upper bound on F that yields a lower bound on the significance level.

25.

Univariate Tests

Source	Measure		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
App	Acc	Sphericity Assumed	.941	1	.941	12.497	.001	.275
		Greenhouse-Geisser	.941	1.000	.941	12.497	.001	.275
		Huynh-Feldt	.941	1.000	.941	12.497	.001	.275
		Lower-bound	.941	1.000	.941	12.497	.001	.275
	Time	Sphericity Assumed	17132.788	1	17132.788	75.302	<.001	.695
		Greenhouse-Geisser	17132.788	1.000	17132.788	75.302	<.001	.695
		Huynh-Feldt	17132.788	1.000	17132.788	75.302	<.001	.695
		Lower-bound	17132.788	1.000	17132.788	75.302	<.001	.695
App * ROLE	Acc	Sphericity Assumed	.721	2	.361	4.791	.015	.225
		Greenhouse-Geisser	.721	2.000	.361	4.791	.015	.225
		Huynh-Feldt	.721	2.000	.361	4.791	.015	.225
		Lower-bound	.721	2.000	.361	4.791	.015	.225
	Time	Sphericity Assumed	293.552	2	146.776	.645	.531	.038
		Greenhouse-Geisser	293.552	2.000	146.776	.645	.531	.038
		Huynh-Feldt	293.552	2.000	146.776	.645	.531	.038
		Lower-bound	293.552	2.000	146.776	.645	.531	.038

Error(App)	Acc	Sphericity Assumed	2.484	33	.075			
		Greenhouse-Geisser	2.484	33.000	.075			
		Huynh-Feldt	2.484	33.000	.075			
		Lower-bound	2.484	33.000	.075			
	Time	Sphericity Assumed	7508.185	33	227.521			
		Greenhouse-Geisser	7508.185	33.000	227.521			
		Huynh-Feldt	7508.185	33.000	227.521			
		Lower-bound	7508.185	33.000	227.521			
Trans	Acc	Sphericity Assumed	.159	3	.053	1.780	.156	.051
		Greenhouse-Geisser	.159	1.773	.090	1.780	.181	.051
		Huynh-Feldt	.159	1.980	.080	1.780	.177	.051
		Lower-bound	.159	1.000	.159	1.780	.191	.051
	Time	Sphericity Assumed	12840.243	3	4280.081	52.477	<.001	.614
		Greenhouse-Geisser	12840.243	1.502	8549.997	52.477	<.001	.614
		Huynh-Feldt	12840.243	1.653	7768.193	52.477	<.001	.614
		Lower-bound	12840.243	1.000	12840.243	52.477	<.001	.614
Trans * ROLE	Acc	Sphericity Assumed	.087	6	.015	.488	.816	.029
		Greenhouse-Geisser	.087	3.547	.025	.488	.723	.029
		Huynh-Feldt	.087	3.961	.022	.488	.743	.029
		Lower-bound	.087	2.000	.044	.488	.618	.029
	Time	Sphericity Assumed	1169.544	6	194.924	2.390	.034	.127
		Greenhouse-Geisser	1169.544	3.004	389.385	2.390	.080	.127
		Huynh-Feldt	1169.544	3.306	353.780	2.390	.073	.127
		Lower-bound	1169.544	2.000	584.772	2.390	.107	.127
Error(Trans)	Acc	Sphericity Assumed	2.951	99	.030			
		Greenhouse-Geisser	2.951	58.523	.050			
		Huynh-Feldt	2.951	65.355	.045			
		Lower-bound	2.951	33.000	.089			
	Time	Sphericity Assumed	8074.554	99	81.561			
		Greenhouse-Geisser	8074.554	49.559	162.929			
		Huynh-Feldt	8074.554	54.547	148.031			
		Lower-bound	8074.554	33.000	244.683			
App * Trans	Acc	Sphericity Assumed	.159	3	.053	1.780	.156	.051
		Greenhouse-Geisser	.159	1.773	.090	1.780	.181	.051
		Huynh-Feldt	.159	1.980	.080	1.780	.177	.051
		Lower-bound	.159	1.000	.159	1.780	.191	.051
	Time	Sphericity Assumed	8261.977	3	2753.992	34.898	<.001	.514

		Greenhouse-Geisser	8261.977	1.917	4309.963	34.898	<.001	.514
		Huynh-Feldt	8261.977	2.156	3832.367	34.898	<.001	.514
		Lower-bound	8261.977	1.000	8261.977	34.898	<.001	.514
App * Trans * ROLE	Acc	Sphericity Assumed	.087	6	.015	.488	.816	.029
		Greenhouse-Geisser	.087	3.547	.025	.488	.723	.029
		Huynh-Feldt	.087	3.961	.022	.488	.743	.029
		Lower-bound	.087	2.000	.044	.488	.618	.029
	Time	Sphericity Assumed	262.004	6	43.667	.553	.766	.032
		Greenhouse-Geisser	262.004	3.834	68.339	.553	.690	.032
		Huynh-Feldt	262.004	4.312	60.766	.553	.710	.032
		Lower-bound	262.004	2.000	131.002	.553	.580	.032
Error(App*Trans)	Acc	Sphericity Assumed	2.951	99	.030			
		Greenhouse-Geisser	2.951	58.523	.050			
		Huynh-Feldt	2.951	65.355	.045			
		Lower-bound	2.951	33.000	.089			
	Time	Sphericity Assumed	7812.649	99	78.916			
		Greenhouse-Geisser	7812.649	63.259	123.502			
		Huynh-Feldt	7812.649	71.143	109.816			
		Lower-bound	7812.649	33.000	236.747			

26.

Tests of Within-Subjects Contrasts

Source	Measure	App	Trans	Type III Sum of Squares	df	Mean Square	F	Sig.
App	Acc	Linear		.941	1	.941	12.497	.001
	Time	Linear		17132.788	1	17132.788	75.302	<.001
App * ROLE	Acc	Linear		.721	2	.361	4.791	.015
	Time	Linear		293.552	2	146.776	.645	.531
Error(App)	Acc	Linear		2.484	33	.075		
	Time	Linear		7508.185	33	227.521		
Trans	Acc		Linear	.034	1	.034	.641	.429
			Quadratic	.077	1	.077	5.272	.028
			Cubic	.048	1	.048	2.233	.145
	Time		Linear	4239.506	1	4239.506	49.847	<.001
			Quadratic	5819.692	1	5819.692	112.665	<.001
			Cubic	2781.045	1	2781.045	25.755	<.001
Trans * ROLE	Acc		Linear	.034	2	.017	.317	.731
			Quadratic	.016	2	.008	.558	.577
			Cubic	.037	2	.019	.870	.428
	Time		Linear	120.377	2	60.188	.708	.500
			Quadratic	422.123	2	211.062	4.086	.026
			Cubic	627.044	2	313.522	2.904	.069
Error(Trans)	Acc		Linear	1.763	33	.053		
			Quadratic	.484	33	.015		
			Cubic	.704	33	.021		
	Time		Linear	2806.637	33	85.050		
			Quadratic	1704.613	33	51.655		

			Cubic	3563.304	33	107.979		
App * Trans	Acc	Linear	Linear	.034	1	.034	.641	.429
			Quadratic	.077	1	.077	5.272	.028
			Cubic	.048	1	.048	2.233	.145
	Time	Linear	Linear	2050.001	1	2050.001	17.084	<.001
			Quadratic	3946.463	1	3946.463	83.580	<.001
			Cubic	2265.514	1	2265.514	32.580	<.001
App * Trans * ROLE	Acc	Linear	Linear	.034	2	.017	.317	.731
			Quadratic	.016	2	.008	.558	.577
			Cubic	.037	2	.019	.870	.428
	Time	Linear	Linear	13.948	2	6.974	.058	.944
			Quadratic	107.440	2	53.720	1.138	.333
			Cubic	140.615	2	70.308	1.011	.375
Error(App*Trans)	Acc	Linear	Linear	1.763	33	.053		
			Quadratic	.484	33	.015		
			Cubic	.704	33	.021		
	Time	Linear	Linear	3959.732	33	119.992		
			Quadratic	1558.185	33	47.218		
			Cubic	2294.732	33	69.537		

27.

Tests of Within-Subjects Contrasts

Source	Measure	App	Trans	Partial Eta Squared
App	Acc	Linear		.275
	Time	Linear		.695
App * ROLE	Acc	Linear		.225
	Time	Linear		.038
Error(App)	Acc	Linear		
	Time	Linear		
Trans	Acc		Linear	.019
			Quadratic	.138
			Cubic	.063
	Time		Linear	.602
			Quadratic	.773
			Cubic	.438
Trans * ROLE	Acc		Linear	.019
			Quadratic	.033
			Cubic	.050
	Time		Linear	.041
			Quadratic	.198
			Cubic	.150
Error(Trans)	Acc		Linear	
			Quadratic	
			Cubic	
	Time		Linear	
			Quadratic	
			Cubic	
App * Trans	Acc	Linear	Linear	.019
			Quadratic	.138
			Cubic	.063
	Time	Linear	Linear	.341
			Quadratic	.717
			Cubic	.497
App * Trans * ROLE	Acc	Linear	Linear	.019

			Quadratic	.033
			Cubic	.050
	Time	Linear	Linear	.004
			Quadratic	.065
			Cubic	.058
Error(App*Trans)	Acc	Linear	Linear	
			Quadratic	
			Cubic	
	Time	Linear	Linear	
			Quadratic	
			Cubic	

28.

Tests of Between-Subjects Effects

Transformed Variable: Average

Source	Measure	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Intercept	Acc	131.864	1	131.864	1752.071	<.001	.982
	Time	459594.876	1	459594.876	1720.484	<.001	.981
ROLE	Acc	.721	2	.361	4.791	.015	.225
	Time	1228.075	2	614.038	2.299	.116	.122
Error	Acc	2.484	33	.075			
	Time	8815.327	33	267.131			

29.

30.

Estimated Marginal Means

31.

1. Grand Mean

Measure	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Acc	.922	.022	.877	.967
Time	54.439	1.312	51.769	57.110

32.

33.

2. App

Estimates

Measure	App	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
Acc	1	.844	.044	.755	.934
	2	1.000	.000	1.000	1.000
Time	1	64.950	2.238	60.396	69.504
	2	43.929	1.170	41.548	46.309

34.

Pairwise Comparisons

Measure	(I) App	(J) App	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b]	
						Lower Bound	Upper Bound
Acc	1	2	-.156*	.044	.001	-.245	-.066
	2	1	.156*	.044	.001	.066	.245

Time	1	2	21.022*	2.423	<.001	16.093	25.950
	2	1	-21.022*	2.423	<.001	-25.950	-16.093

Based on estimated marginal means
*. The mean difference is significant at the .05 level.
b. Adjustment for multiple comparisons: Bonferroni.

35.

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Pillai's trace	.718	40.669[a]	2.000	32.000	<.001	.718
Wilks' lambda	.282	40.669[a]	2.000	32.000	<.001	.718
Hotelling's trace	2.542	40.669[a]	2.000	32.000	<.001	.718
Roy's largest root	2.542	40.669[a]	2.000	32.000	<.001	.718

Each F tests the multivariate effect of App. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.
a. Exact statistic

36.

37.

3. Trans

Estimates

Measure	Trans	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Acc	1	.917	.036	.843	.990
	2	.917	.036	.843	.990
	3	.883	.031	.819	.947
	4	.972	.026	.920	1.024
Time	1	57.222	1.964	53.227	61.217
	2	57.222	1.964	53.227	61.217
	3	63.909	1.769	60.310	67.507
	4	39.405	1.536	36.279	42.531

38.

Pairwise Comparisons

Measure	(I) Trans	(J) Trans	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	95% Confidence Interval for Difference[b] Upper Bound
Acc	1	2	.000	.000	.	.000	.000
		3	.034	.042	1.000	-.084	.152
		4	-.056	.046	1.000	-.184	.073
	2	1	.000	.000	.	.000	.000
		3	.034	.042	1.000	-.084	.152
		4	-.056	.046	1.000	-.184	.073
	3	1	-.034	.042	1.000	-.152	.084
		2	-.034	.042	1.000	-.152	.084
		4	-.089	.039	.169	-.198	.020
	4	1	.056	.046	1.000	-.073	.184
		2	.056	.046	1.000	-.073	.184
		3	.089	.039	.169	-.020	.198
Time	1	2	.000	.000	.	.000	.000
		3	-6.687	2.730	.119	-14.349	.976
		4	17.817*	1.582	<.001	13.376	22.259
	2	1	.000	.000	.	.000	.000
		3	-6.687	2.730	.119	-14.349	.976
		4	17.817*	1.582	<.001	13.376	22.259

	3	1	6.687	2.730	.119	-.976	14.349
		2	6.687	2.730	.119	-.976	14.349
		4	24.504*	2.309	<.001	18.024	30.984
	4	1	-17.817*	1.582	<.001	-22.259	-13.376
		2	-17.817*	1.582	<.001	-22.259	-13.376
		3	-24.504*	2.309	<.001	-30.984	-18.024

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

b. Adjustment for multiple comparisons: Bonferroni.

39.

Multivariate Tests

	Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
Pillai's trace	.877	53.443[a]	4.000	30.000	<.001	.877
Wilks' lambda	.123	53.443[a]	4.000	30.000	<.001	.877
Hotelling's trace	7.126	53.443[a]	4.000	30.000	<.001	.877
Roy's largest root	7.126	53.443[a]	4.000	30.000	<.001	.877

Each F tests the multivariate effect of Trans. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

40.

41.

4. App * Trans

42.

Estimates

Measure	App	Trans	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Acc	1	1	.833	.072	.686	.980
		2	.833	.072	.686	.980
		3	.766	.063	.638	.894
		4	.944	.051	.841	1.048
	2	1	1.000	.000	1.000	1.000
		2	1.000	.000	1.000	1.000
		3	1.000	.000	1.000	1.000
		4	1.000	.000	1.000	1.000
Time	1	1	66.190	3.274	59.529	72.852
		2	66.190	3.274	59.529	72.852
		3	65.873	2.918	59.936	71.810
		4	61.548	2.497	56.468	66.627
	2	1	48.254	1.986	44.213	52.295
		2	48.254	1.986	44.213	52.295
		3	61.944	2.314	57.237	66.652
		4	17.262	.960	15.309	19.215

43.

Pairwise Comparisons

Measure	Trans	(I) App	(J) App	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound
Acc	1	1	2	-.167*	.072	.027	-.314
		2	1	.167*	.072	.027	.020
	2	1	2	-.167*	.072	.027	-.314

		2	1	.167*	.072	.027	.020
	3	1	2	-.234*	.063	<.001	-.362
		2	1	.234*	.063	<.001	.106
	4	1	2	-.056	.051	.284	-.159
		2	1	.056	.051	.284	-.048
Time	1	1	2	17.937*	3.729	<.001	10.350
		2	1	-17.937*	3.729	<.001	-25.523
	2	1	2	17.937*	3.729	<.001	10.350
		2	1	-17.937*	3.729	<.001	-25.523
	3	1	2	3.929	3.902	.321	-4.010
		2	1	-3.929	3.902	.321	-11.867
	4	1	2	44.286*	2.206	<.001	39.798
		2	1	-44.286*	2.206	<.001	-48.773

44.

Pairwise Comparisons

Measure	Trans	(I) App	(J) App	95% Confidence Interval for Difference Upper Bound
Acc	1	1	2	-.020
		2	1	.314
	2	1	2	-.020
		2	1	.314
	3	1	2	-.106
		2	1	.362
	4	1	2	.048
		2	1	.159
Time	1	1	2	25.523
		2	1	-10.350
	2	1	2	25.523
		2	1	-10.350
	3	1	2	11.867
		2	1	4.010
	4	1	2	48.773
		2	1	-39.798

45.

Based on estimated marginal means
*. The mean difference is significant at the .05 level.
b. Adjustment for multiple comparisons: Bonferroni.

46.

Multivariate Tests

Trans		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
1	Pillai's trace	.469	14.127[a]	2.000	32.000	<.001	.469
	Wilks' lambda	.531	14.127[a]	2.000	32.000	<.001	.469
	Hotelling's trace	.883	14.127[a]	2.000	32.000	<.001	.469
	Roy's largest root	.883	14.127[a]	2.000	32.000	<.001	.469
2	Pillai's trace	.469	14.127[a]	2.000	32.000	<.001	.469
	Wilks' lambda	.531	14.127[a]	2.000	32.000	<.001	.469
	Hotelling's trace	.883	14.127[a]	2.000	32.000	<.001	.469
	Roy's largest root	.883	14.127[a]	2.000	32.000	<.001	.469
3	Pillai's trace	.310	7.193[a]	2.000	32.000	.003	.310
	Wilks' lambda	.690	7.193[a]	2.000	32.000	.003	.310

	Hotelling's trace	.450	7.193[a]	2.000	32.000	.003	.310
	Roy's largest root	.450	7.193[a]	2.000	32.000	.003	.310
4	Pillai's trace	.931	215.397[a]	2.000	32.000	<.001	.931
	Wilks' lambda	.069	215.397[a]	2.000	32.000	<.001	.931
	Hotelling's trace	13.462	215.397[a]	2.000	32.000	<.001	.931
	Roy's largest root	13.462	215.397[a]	2.000	32.000	<.001	.931

Each F tests the multivariate simple effects of App within each level combination of the other effects shown. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

47.

48.

5. App * Trans

49.

Estimates

Measure	App	Trans	Mean	Std. Error	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Acc	1	1	.833	.072	.686	.980
		2	.833	.072	.686	.980
		3	.766	.063	.638	.894
		4	.944	.051	.841	1.048
	2	1	1.000	.000	1.000	1.000
		2	1.000	.000	1.000	1.000
		3	1.000	.000	1.000	1.000
		4	1.000	.000	1.000	1.000
Time	1	1	66.190	3.274	59.529	72.852
		2	66.190	3.274	59.529	72.852
		3	65.873	2.918	59.936	71.810
		4	61.548	2.497	56.468	66.627
	2	1	48.254	1.986	44.213	52.295
		2	48.254	1.986	44.213	52.295
		3	61.944	2.314	57.237	66.652
		4	17.262	.960	15.309	19.215

50.

Pairwise Comparisons

Measure	App	(I) Trans	(J) Trans	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b] Lower Bound	95% Confidence Interval for Difference[b] Upper Bound
Acc	1	1	2	.000	.000	.	.000	.000
			3	.067	.084	1.000	-.169	.304
			4	-.111	.091	1.000	-.367	.145
		2	1	.000	.000	.	.000	.000
			3	.067	.084	1.000	-.169	.304
			4	-.111	.091	1.000	-.367	.145
		3	1	-.067	.084	1.000	-.304	.169
			2	-.067	.084	1.000	-.304	.169
			4	-.179	.078	.169	-.397	.040
		4	1	.111	.091	1.000	-.145	.367
			2	.111	.091	1.000	-.145	.367
			3	.179	.078	.169	-.040	.397
	2	1	2	.000	.000	.	.000	.000
			3	.000	.000	.	.000	.000
			4	.000	.000	.	.000	.000

		2	1	.000	.000	.	.000	.000
			3	.000	.000	.	.000	.000
			4	.000	.000	.	.000	.000
		3	1	.000	.000	.	.000	.000
			2	.000	.000	.	.000	.000
			4	.000	.000	.	.000	.000
		4	1	.000	.000	.	.000	.000
			2	.000	.000	.	.000	.000
			3	.000	.000	.	.000	.000
Time	1	1	2	.000	.000	.	.000	.000
			3	.317	3.843	1.000	-10.469	11.104
			4	4.643	3.216	.949	-4.383	13.669
		2	1	.000	.000	.	.000	.000
			3	.317	3.843	1.000	-10.469	11.104
			4	4.643	3.216	.949	-4.383	13.669
		3	1	-.317	3.843	1.000	-11.104	10.469
			2	-.317	3.843	1.000	-11.104	10.469
			4	4.325	3.790	1.000	-6.311	14.962
		4	1	-4.643	3.216	.949	-13.669	4.383
			2	-4.643	3.216	.949	-13.669	4.383
			3	-4.325	3.790	1.000	-14.962	6.311
	2	1	2	.000	.000	.	.000	.000
			3	-13.690*	3.216	<.001	-22.717	-4.664
			4	30.992*	2.004	<.001	25.366	36.618
		2	1	.000	.000	.	.000	.000
			3	-13.690*	3.216	<.001	-22.717	-4.664
			4	30.992*	2.004	<.001	25.366	36.618
		3	1	13.690*	3.216	<.001	4.664	22.717
			2	13.690*	3.216	<.001	4.664	22.717
			4	44.683*	2.458	<.001	37.784	51.581
		4	1	-30.992*	2.004	<.001	-36.618	-25.366
			2	-30.992*	2.004	<.001	-36.618	-25.366
			3	-44.683*	2.458	<.001	-51.581	-37.784

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

b. Adjustment for multiple comparisons: Bonferroni.

51.

Multivariate Tests

App		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared
1	Pillai's trace	.195	1.821[a]	4.000	30.000	.151	.195
	Wilks' lambda	.805	1.821[a]	4.000	30.000	.151	.195
	Hotelling's trace	.243	1.821[a]	4.000	30.000	.151	.195
	Roy's largest root	.243	1.821[a]	4.000	30.000	.151	.195
2	Pillai's trace	.947	284.210[a]	2.000	32.000	<.001	.947
	Wilks' lambda	.053	284.210[a]	2.000	32.000	<.001	.947
	Hotelling's trace	17.763	284.210[a]	2.000	32.000	<.001	.947
	Roy's largest root	17.763	284.210[a]	2.000	32.000	<.001	.947

Each F tests the multivariate simple effects of Trans within each level combination of the other effects shown. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

a. Exact statistic

52. **Post Hoc Tests**

53. **ROLE**

54.

Multiple Comparisons

Bonferroni

Measure	(I) ROLE	(J) ROLE	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound
Acc	Maintenance Rep	Manager	.0357	.05987	1.000	-.1153
		Mechanic	.1354	.06261	.114	-.0225
	Manager	Maintenance Rep	-.0357	.05987	1.000	-.1867
		Mechanic	.0997*	.03510	.023	.0112
	Mechanic	Maintenance Rep	-.1354	.06261	.114	-.2933
		Manager	-.0997*	.03510	.023	-.1882
Time	Maintenance Rep	Manager	-7.3810	3.56659	.139	-16.3766
		Mechanic	-5.3125	3.73002	.491	-14.7204
	Manager	Maintenance Rep	7.3810	3.56659	.139	-1.6147
		Mechanic	2.0685	2.09110	.989	-3.2057
	Mechanic	Maintenance Rep	5.3125	3.73002	.491	-4.0954
		Manager	-2.0685	2.09110	.989	-7.3426

55.

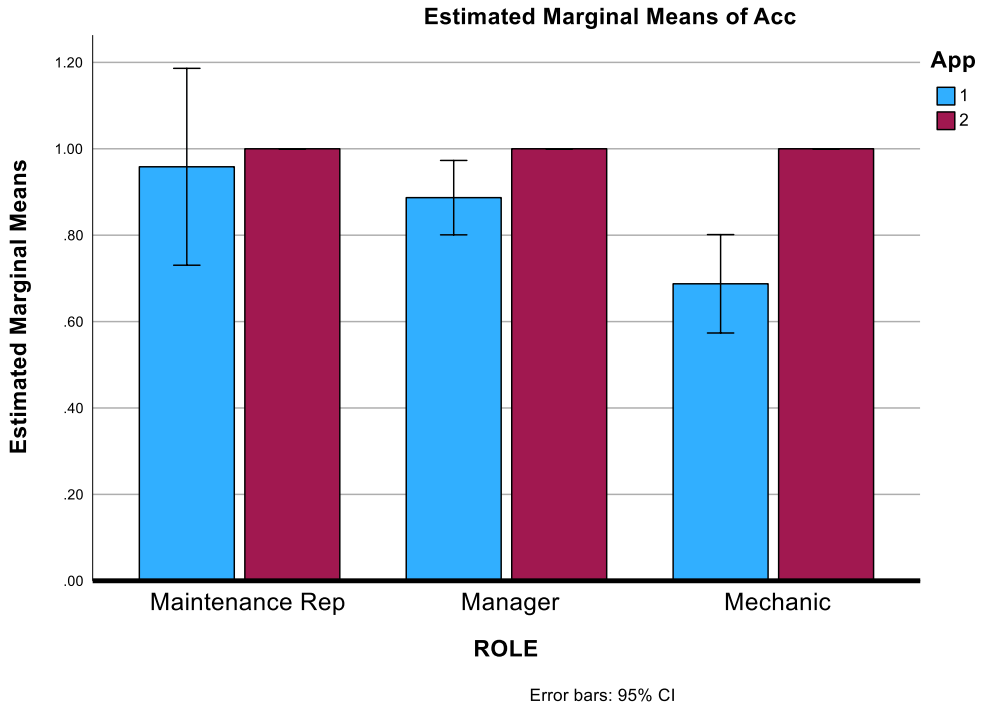
Multiple Comparisons

Bonferroni

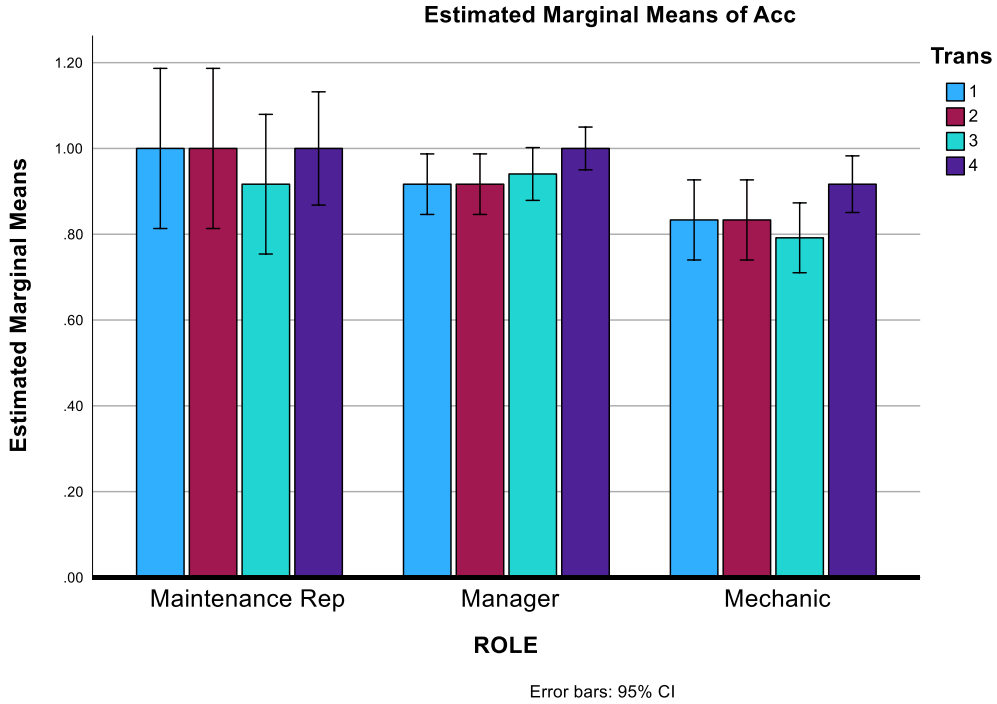
Measure	(I) ROLE	(J) ROLE	95% Confidence Interval Upper Bound
Acc	Maintenance Rep	Manager	.1867
		Mechanic	.2933
	Manager	Maintenance Rep	.1153
		Mechanic	.1882
	Mechanic	Maintenance Rep	.0225
		Manager	-.0112
Time	Maintenance Rep	Manager	1.6147
		Mechanic	4.0954
	Manager	Maintenance Rep	16.3766
		Mechanic	7.3426
	Mechanic	Maintenance Rep	14.7204
		Manager	3.2057

*. The mean difference is significant at the .05 level.

56.



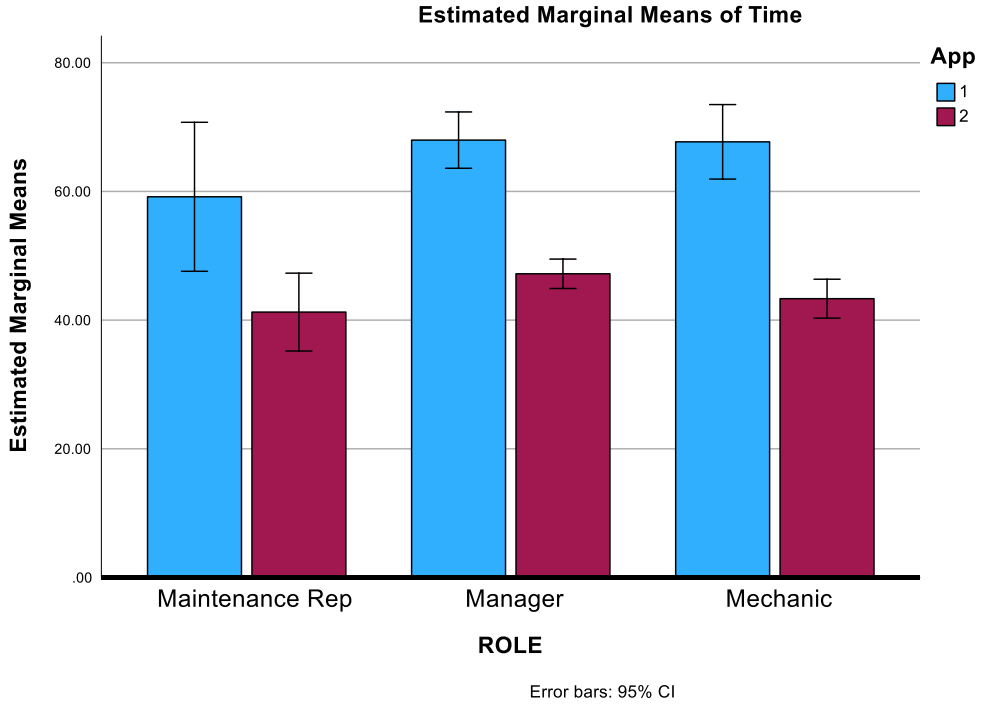
57.
58.



59.
60.
61.

Time

62.



63.

7.1.2 c. Difficulties Using the System

Page	Description of Error	Percentage of Subjects Demonstrating Difficulties
Quick Find	Difficulty understanding the menu options (search by part number or description or airline)	30%
	Did not realize they could combine search results to build a custom table	50%
All	Make fields typable	80%
	Internet Connection Issues	100%

10. Decentralized Application Simulations

a. Demonstration of Log Transactions

Read Sample Log

```
# Now read the log and transact the log
Log_file=pd.read_csv("/content/Sample_Log.csv") #replace by the path to log file
Log_file['Quantity']=[0 if np.isnan(x) else int(x) for x in Log_file['Quantity']]
index_names=Log_file[Log_file['Quantity']<=0].index
Log_file.drop(index_names, inplace = True)
Log_file.reset_index(drop=True, inplace=True)

Log_file['JFK_Location']=["0" if x=="nan" else str(x) for x in Log_file['JFK_Location']]
index_names1=Log_file[Log_file['JFK_Location']=="0"].index
Log_file.drop(index_names1, inplace = True)
  Log_file.reset_index(drop=True, inplace=True)

  Transaction_type1=Log_file['Transaction_type'].to_list()
  Airline1=Log_file['Airline'].to_list()
  Part_Name1=Log_file['Part_Number'].to_list()
  Description1=Log_file['Description'].to_list()
  JFK_Location1 = Log_file['JFK_Location'].to_list()
  Quantity1=[int(x) for x in Log_file['Quantity']]
  Serial_Number1=Log_file['Serial_Number'].to_list()
  Batch_Number1=Log_file['Batch_Number'].to_list()
  Time_Expiration1=Log_file['Time_Expiration'].to_list()

  Part_Number=[Airline1[x]+str(Part_Name1[x])+str(Description1[x])+str(JFK_Location1[x])+str(Serial_Number1[x])+str(Batch_Number1[x])+str(Time_Expiration1[x]) for x in range(len(Transaction_type1))]

  # Create 3 boolean lists for
  doesHaveSerialNumber=[False if Serial_Number1[x] =="NaN" else True for x in range(len(Transaction_type1)) ]
  doesHaveExpiry=[False if Time_Expiration1[x] =="1/1/2099" else True for x in range(len(Transaction_type1)) ]
  doesHaveBatchNumber=[False if Batch_Number1[x] =="NaN" else True for x in range(len(Transaction_type1)) ]

  #Get the list of lists ready
  Log1=[]
  Log1=[Transaction_type1,Airline1,Part_Name1,Part_Number,Description1,doesHaveSerialNumber,doesHaveExpiry,doesHaveBatchNumber,Quantity1,JFK_Location1,Serial_Number1,Time_Expiration1,Batch_Number1]
```

Visualize Sample Log

```
# Sample Log as imported form the CSV file
```

	Transaction_type	Airline	Part_Number	Description	JFK_Location	Quantity	Serial_Number	Batch_Number	Time_Expiration
0	IN	QF	ACREM1A	MEGAPHONE	H10-C-5-3-0	1	43661	NaN	1-Jun-26
1	IN	BA	2197	EASTMAN TURBO OIL	H10-B-1-1-0	336	NaN	E0561	30-Jan-26
2	OUT	ANA	83276C12730-1ANA	PLASTIC SEAL - Blue Seal	H10-D-4-5-0	100	NaN	A9047	1/1/2099
3	OUT	ANA	83276C12730-1ANA	PLASTIC SEAL - Blue Seal	H10-D-4-5-0	100	NaN	A9047	1/1/2099
4	IN	ANA	AF888	SPEED PATCH	H10-D-4-7-0	1	NaN	A29387A	22-Nov-24
5	TRANS-OUT	EI	TN640	A321 IDG OIL TN640	T8-G-2-1-0	-24	NaN	5012301536	1/1/2099
6	TRANS-IN	EI	TN640	A321 IDG OIL TN640	H10-D-6-1-0	24	NaN	5012301536	1/1/2099
7	IN	EI	NSA8205-171	O RING	H10-D-6-3-3	2	NaN	510001936	31-May-28
8	OUT	EI	NSA8205-171	O RING	H10-D-6-3-3	2	NaN	510001936	2/2/2024
9	IN	EI	NSA8205-171	O RING	H10-D-6-3-3	2	NaN	510001936	31-May-24
10	OUT	BA	MA1172577	PAPER ROLL FFF WITH A1172577:F9111	T8-D-2-4-0	1	NaN	7381091	1/1/2099
11	IN	BA	4-48558	WHEEL ASSY MAIN B777	T8-G-4-1-0	1	TBE0698	7529359	23-Nov-28
12	IN	BA	3505788-8	V STARTER TRENT	T8-G-4-1-0	1	3505788-1801	7473062	1/1/2099
13	TRANS-OUT	BA	CR3222-6-07	RIVET	T8-D-2-7-19	-1	NaN	3124103	1/1/2099
14	TRANS-IN	BA	CR3222-6-07	RIVET	T8-D-3-7-0	1	NaN	3124103	1/1/2099
15	TRANS-OUT	BA	CR3222-6-07	Rivet	T8-D-3-7-0	-1	NaN	3124103	1/1/2099
16	TRANS-IN	BA	CR3222-6-07	Rivet	T8-D-2-7-19	1	NaN	3124103	1/1/2099
17	OUT	BA	ESMP1U92AUGB	EXT. Lead GRD Crew H/SET -High Vis	T8-F-4-4-0	1	NaN	7494749	1/1/2099
18	IN	BA	ESMP1U92AUGB	EXT. LEAD GRD CREW H/SET -HIGH VIS	T8-F-4-4-0	1	NaN	7494749	1/1/2099
19	TRANS-OUT	BA	488550-09-01	ACE-ELEC. ACTUATOR CONTROL	T8-G-4-2-0	-1	NaN	2736119	1/1/2099
20	TRANS-IN	BA	488550-09-01	ACE-ELEC. ACTUATOR CONTROL	T8-G-3-2-0	1	NaN	2736119	1/1/2099

Log1 file: the transformed sampled log, ready to be transacted in the blockchain
Note the “Part_Unique_ID” field (red box) which tokenizes assets, and the format changes to match the Smart Contract’s constraints.

	Transaction_type	Airline	Part_Number	Part_uniqueID	Description
0	IN	QF	ACREM1A	QFACREM1AMEGAPHONEH10-C-5-3-043661nan1-Jun-26	MEGAPHONE
1	IN	BA	2197	BA2197EASTMAN TURBO OILH10-B-1-1-0nanE056130-J...	EASTMAN TURBO OIL
2	OUT	ANA	83276C12730-1ANA	ANA83276C12730-1ANAPLASTIC SEAL - Blue SealH10...	PLASTIC SEAL - Blue Seal
3	IN	ANA	AF888	ANAAF888SPEED PATCHH10-D-4-7-0nanA29387A22-Nov-24	SPEED PATCH
4	TRANS-OUT	EI	TN640	EITN640A321 IDG OIL TN640T8-G-2-1-0nan50123015...	A321 IDG OIL TN640
5	TRANS-IN	EI	TN640	EITN640A321 IDG OIL TN640H10-D-6-1-0nan5012301...	A321 IDG OIL TN640
6	IN	EI	NSA8205-171	EINSA8205-171O RINGH10-D-6-3-3nan51000193631-M...	O RING
7	OUT	EI	NSA8205-171	EINSA8205-171O RINGH10-D-6-3-3nan5100019362/2/...	O RING
8	IN	EI	NSA8205-171	EINSA8205-171O RINGH10-D-6-3-3nan51000193631-M...	O RING
9	OUT	BA	MA1172577	BAMA1172577PAPER ROLL FFF WITH A1172577:F9111T...	PAPER ROLL FFF WITH A1172577:F9111
10	IN	BA	4-48558	BA4-48558WHEEL ASSY MAIN B777T8-G-4-1-0TBE0698...	WHEEL ASSY MAIN B777
11	IN	BA	3505788-8	BA3505788-8V STARTER TRENT T8-G-4-1-03505788-1...	V STARTER TRENT
12	TRANS-OUT	BA	CR3222-6-07	BACR3222-6-07RIVETT8-D-2-7-19nan31241031/1/2099	RIVET
13	TRANS-IN	BA	CR3222-6-07	BACR3222-6-07RIVETT8-D-3-7-0nan31241031/1/2099	RIVET
14	TRANS-OUT	BA	CR3222-6-07	BACR3222-6-07RivetT8-D-3-7-0nan31241031/1/2099	Rivet
15	TRANS-IN	BA	CR3222-6-07	BACR3222-6-07RivetT8-D-2-7-19nan31241031/1/2099	Rivet
16	OUT	BA	ESMP1U92AUGB	BAESMP1U92AUGBEXT. Lead GRD Crew H/SET -High V...	EXT. Lead GRD Crew H/SET -High Vis
17	IN	BA	ESMP1U92AUGB	BAESMP1U92AUGBEXT. LEAD GRD CREW H/SET -HIGH V...	EXT. LEAD GRD CREW H/SET -HIGH VIS
18	TRANS-OUT	BA	488550-09-01	BA488550-09-01ACE-ELEC. ACTUATOR CONTROLT8-G-4...	ACE-ELEC. ACTUATOR CONTROL
19	TRANS-IN	BA	488550-09-01	BA488550-09-01ACE-ELEC. ACTUATOR CONTROLT8-G-3...	ACE-ELEC. ACTUATOR CONTROL

Run the Transactions and Verify Result

Transaction of the log, its output and some specs (average cost in eth, average duration)

```
#start a timer to monitor time taken to complete transactions
start=time.time()
Hxs_log, PartInfo_log=transact_loglists( Log1,Acc_Dict)
stop=time.time()
```

```
IN completed
proceeding to i = 1
IN completed
proceeding to i = 2
IN completed
proceeding to i = 3
TRANS-OUT completed
proceeding to i = 4
TRANS-IN completed
proceeding to i = 5
IN completed
proceeding to i = 6
OUT completed
proceeding to i = 7
IN completed
proceeding to i = 8
OUT completed
proceeding to i = 9
IN completed
proceeding to i = 10
IN completed
proceeding to i = 11
TRANS-OUT completed
proceeding to i = 12
TRANS-IN completed
proceeding to i = 13
TRANS-OUT completed
proceeding to i = 14
TRANS-IN completed
proceeding to i = 15
OUT completed
proceeding to i = 16
IN completed
proceeding to i = 17
TRANS-OUT completed
proceeding to i = 18
TRANS-IN completed
```

```
def get_previous_transaction_cost(Hx):
  receipt=w3.eth.get_transaction(Hx)
  transaction_cost=receipt['v']*w3.eth.gas_price
  return transaction_cost
```

```
Avg_time=(stop-start)/len(Hxs_log)
Avg_cost=np.sum([get_previous_transaction_cost(x) for x in Hxs_log])/len(Hxs_log)
```

```
print('N entries in log file: %s'%len(Log_file))
print('N transactions on the blockchain: %s'%len(PartInfo_Log))
print('Average time per transaction: %s'%Avg_time)
print('Average Transaction Cost: %s'%Avg_cost)
```

```
N entries in log file: 19
N transactions on the blockchain: 19
Average time per transaction: 19.770716428756714
Average Transaction Cost: 344958080.1818182
```

Run the Transactions and Verify Result

Query the smart contract to verify all transactions in the log went through

```
PartInfo_Log_df=pd.DataFrame(PartInfo_Log).drop_duplicates()
PartInfo_Log_df.columns=['Part_Number','Part_uniqueID','Description','doesHaveSerialNumber','doesHaveExpiry'
,'doesHaveBatchNumber','Quantity','JFK_Location','Serial_Number','Time_Expiration','Batch_Number','a']
PartInfo_Log_df
```

	Part_Number	Part_uniqueID	Description
0	ACREM1A	QFACREM1AMEGAPHONEH10-C-5-3-043661nan1-Jun-26	MEGAPHONE
1	2197	BA2197EASTMAN TURBO OILH10-B-1-1-0nanE056130-J...	EASTMAN TURBO OIL
2	AF888	ANAAF888SPEED PATCHH10-D-4-7-0nanA29387A22-Nov-24	SPEED PATCH
3	TN640	EITN640A321 IDG OIL TN640T8-G-2-1-0nan50123015...	A321 IDG OIL TN640
4	TN640	EITN640A321 IDG OIL TN640H10-D-6-1-0nan5012301...	A321 IDG OIL TN640
5	NSA8205-171	EINSA8205-171O RINGH10-D-6-3-3nan51000193631-M...	O RING
6	NSA8205-171	EINSA8205-171O RINGH10-D-6-3-3nan5100019362/2/...	O RING
7	NSA8205-171	EINSA8205-171O RINGH10-D-6-3-3nan51000193631-M...	O RING
8	MA1172577	BAMA1172577PAPER ROLL FFF WITH A1172577:F9111T...	PAPER ROLL FFF WITH A1172577:F9111
9	4-48558	BA4-48558WHEEL ASSY MAIN B777T8-G-4-1-0TBE0698...	WHEEL ASSY MAIN B777
10	3505788-8	BA3505788-8V STARTER TRENT T8-G-4-1-03505788-1...	V STARTER TRENT
11	CR3222-6-07	BACR3222-6-07RIVETT8-D-2-7-19nan31241031/1/2099	RIVET
12	CR3222-6-07	BACR3222-6-07RIVETT8-D-3-7-0nan31241031/1/2099	RIVET
13	CR3222-6-07	BACR3222-6-07RivetT8-D-3-7-0nan31241031/1/2099	Rivet
14	CR3222-6-07	BACR3222-6-07RivetT8-D-2-7-19nan31241031/1/2099	Rivet
15	ESMP1U92AUGB	BAESMP1U92AUGBEXT. Lead GRD Crew H/SET -High V...	EXT. Lead GRD Crew H/SET -High Vis
16	ESMP1U92AUGB	BAESMP1U92AUGBEXT. LEAD GRD CREW H/SET -HIGH V...	EXT. LEAD GRD CREW H/SET -HIGH VIS
17	488550-09-01	BA488550-09-01ACE-ELEC. ACTUATOR CONTROLT8-G-4...	ACE-ELEC. ACTUATOR CONTROL
18	488550-09-01	BA488550-09-01ACE-ELEC. ACTUATOR CONTROLT8-G-3...	ACE-ELEC. ACTUATOR CONTROL

b. Error Simulations

Based on the previous Smart Contract query, create a log to trigger error responses and verify that incorrect transactions are being detected and stopped by the Smart Contract.

	Transaction_type	Airline	Part_Nb	Description	JFK_Location	Quantity	Serial_Number	Batch_Number	Time_Expiration
0	IN	EI	SIM1	O RING	H10-D-6-3-3	2	N/A	5012301536	1/1/2099
1	IN	EI	SIM1	O RING	H10-D-6-3-3	2	N/A	5012301536	1/1/2099
2	OUT	EI	SIM1-w	O RING	H10-D-6-3-3	2	N/A	5012301536	1/1/2099
3	OUT	EI	SIM1	O RING	H10-D-6-3-3	2	N/A	5012301536	20/1/2099
4	OUT	EI	SIM1	O RING	H10-D-6-3-3	2	N/A	1111111111	1/1/2099
5	OUT	EI	SIM1	O RING	H10-D-50-3-3	2	N/A	5012301536	1/1/2099

A hypothetical log was imagined, where the first two entries were inputting 4 simulated parts into Aer Lingus' catalog. To verify that such simulated parts did not exist in the catalog at the onset of the simulation, the smart contract was queried and returned the table below, proving the parts were new to the catalog.

	Part_Number	Part_uniqueID	Description	Quantity	JFK_Location	Serial_Number	Time_Expiration	Batch_Number	airline
0				0					EI
1				0					EI
2				0					EI
3				0					EI
4				0					EI
5				0					EI

The first and second transactions should add quantities to the same record, bringing the total amount to 4. After running these first and second transaction the quantity for the simulated part indeed was added (Image below – red box).

```
[372]
    my_contract.functions.getPartInfo(Acc_Dict['EI'].address,'EISIM4O RINGH10-D-6-3-3nan50123015361/2/2099').call()

    ('SIM4',
     'EISIM4O RINGH10-D-6-3-3nan50123015361/2/2099',
     'O RING',
     True,
     True,
     True,
     4,
     'H10-D-6-3-3',
     'nan',
     '1/2/2099',
     '5012301536',
     4)
```

The fourth, fifth and sixth entries in the simulated log (top table) are attempting to transact the "SIM1" simulated part out of the store using the wrong Part Nb (SIM1-w), wrong expiry date (20/1/2099) or wrong Batch Number (11111).
The smart contract identified the errors and outputs a corresponding error message (see figures below).

```
[335] dirc='/content/'
      w3,my_contract=initialize()
      Log1, Log1_df, Hxs, PartInfo, PartInfo_log_before_df, PartInfo_Log_df=transact_log()

      proceeding to i = 0
      nonce 624
      ('', '', '', False, False, False, 0, '', '', '', '', 0)
      IN completed
      proceeding to i = 1
      nonce 625
      ('SIM1', 'EISIM1O RINGH10-D-6-3-3nan50123015361/1/2099', 'O RING', True, False, True, 2, 'H10-D-6-3-3', 'nan', '1/1/2099', '5012301536', 4)
      IN completed
      proceeding to i = 2
      nonce 626
      ---------------------------------------------------------------------------
      ContractLogicError                        Traceback (most recent call last)
      <ipython-input-335-8634c8fa3545> in <cell line: 3>()
            1 dirc='/content/'
            2 w3,my_contract=initialize()
      ----> 3 Log1, Log1_df, Hxs, PartInfo, PartInfo_log_before_df, PartInfo_Log_df=transact_log()

      ------------------------------------ 14 frames -------------------------------------
      /usr/local/lib/python3.10/dist-packages/web3/_utils/error_formatters_utils.py in raise_contract_logic_error_on_revert(response)
          160         # Geth Revert with error message and code 3 case:
          161         if error.get("code") == 3:
      --> 162             raise ContractLogicError(message, data=data)
          163         # Geth Revert without error message case:
          164         elif "execution reverted" in message:

      ContractLogicError: execution reverted: Part does not exist in sender's catalog
```

Figure 24- Simulation transaction 0 to 2: 1 and 2 go through but on 3 the ContractLogicError (red box) arises.

```
[337] dirc='/content/'
      w3,my_contract=initialize()
      Log1, Log1_df, Hxs, PartInfo, PartInfo_log_before_df, PartInfo_Log_df=transact_log()

      proceeding to i = 0
      nonce 626
      ---------------------------------------------------------------------------
      ContractLogicError                        Traceback (most recent call last)
      <ipython-input-337-8634c8fa3545> in <cell line: 3>()
            1 dirc='/content/'
            2 w3,my_contract=initialize()
      ----> 3 Log1, Log1_df, Hxs, PartInfo, PartInfo_log_before_df, PartInfo_Log_df=transact_log()

      ------------------------------------ 14 frames -------------------------------------
      /usr/local/lib/python3.10/dist-packages/web3/_utils/error_formatters_utils.py in raise_contract_logic_error_on_revert(response)
          160         # Geth Revert with error message and code 3 case:
          161         if error.get("code") == 3:
      --> 162             raise ContractLogicError(message, data=data)
          163         # Geth Revert without error message case:
          164         elif "execution reverted" in message:

      ContractLogicError: execution reverted: Part does not exist in sender's catalog
```

Figure 25- Simulation Transaction 3 & 4 & 5, for each, attempted standalone, the same error message (red box) appeared

11. Survey at Time 1

a. Survey Questions

Stores Keeper App User Experience and Feedback Survey

The aim of this survey is to gather essential user feedback for continuous improvement, highlighting usability, identifying training needs, and enhancing user engagement. It supports data-driven decision-making, enabling the alignment of the app's development with user needs and operational requirements.

App user context.

This section is to provide a general idea of the user.

Please select your role. *

- ◯ AMM
- ◯ SMM
- ◯ DMM
- ◯ Maint. Rep
- ◯ Mech

How long have you been working at BA JFK? *

- ◯ 1-5 years
- ◯ 5-10 years
- ◯ 10-15 years
- ◯ 20+ years

User Experience & The System

Questions regarding the user experience of the app and the inventory management system.

3

Have you used the Stores Keeper App on your own, at least once? *

- ◯ Yes
- ◯ No

4

Did Stores Keeper App save you time transacting parts? *

- ◯ Yes
- ◯ No

5

Do you believe the Stores Keeper App can help reduce cost? *

- ◯ Yes
- ◯ No

6

Do you think the Stores Keeper App will help prevent misplacing parts? *

- ◯ Yes
- ◯ No

7

Has the app improved the accuracy of inventory management? *

0	1	2	3	4	5	6	7	8	9	10

Not at all — Extremely

8

How easily were you able to find the items you were looking for using the Stores Keeper App? *

0	1	2	3	4	5	6	7	8	9	10

Not at all likely — Extremely likely

9

What's the most useful feature of the Stores Keeper App? *

- ◯ Quick Find
- ◯ Receipting IN
- ◯ Receipting OUT
- ◯ Parts Transferring
- ◯ Transaction Log

10

How would you rate the user interface of the 'Stores Keeper' app? *

0	1	2	3	4	5	6	7	8	9	10

Poor — Amazing

11

Do you have any data privacy and security concerns regarding the Stores Keeper App? *

- ◯ Yes
- ◯ No

12

How would you describe your overall experience with the "Stores Keeper" app? *

13

How reliable has the Stores Keeper App been ? *

0	1	2	3	4	5	6	7	8	9	10

Not at all likely — Extremely likely

In your opinion, how does the "Stores Keeper" app compare to the previous inventory management system? *

15

How likely are you to continue using the 'Stores Keeper' app in the long term? *

0	1	2	3	4	5	6	7	8	9	10

Not at all likely — Extremely likely

16

Finally, what overall feedback or comments would you like to share about your experience with the "Stores Keeper" app? *

Thank you for taking the time to complete this survey.

This is a steppingstone to future improvements and integration on our road to make JFK the best line station in the world.

b. Survey Results

Id	Start time	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
1	2/25/2024	DMM	1-5 years	Yes	Yes	Yes	Yes	9	8	Quick Find	10	Q11
2	2/25/2024	DMM	1-5 years	Yes	Yes	Yes	Yes	9	10	Quick Find	9	No
3	2/25/2024	DMM	1-5 years	Yes	Yes	Yes	Yes	10	10	Quick Find	8	No
4	2/25/2024	DMM	1-5 years	Yes	Yes	Yes	Yes	10	10	Quick Find	10	No
5	2/26/2024	DMM	10-15 years	Yes	Yes	No	Yes	8	8	Receipting IN	10	No
6	2/27/2024	Maint. Rep	5-10 years	No	No	Yes	Yes	10	5	Quick Find	6	No
7	2/27/2024	Mech	1-5 years	Yes	Yes	Yes	Yes	9	9	Quick Find	5	Yes
8	2/27/2024	Mech	5-10 years	Yes	Yes	Yes	Yes	9	9	Quick Find	9	No
9	2/27/2024	AMM	10-15 years	Yes	Yes	Yes	Yes	8	8	Quick Find	9	No
10	2/27/2024	SMM	5-10 years	Yes	Yes	Yes	Yes	7	7	Quick Find	8	No
11	2/27/2024	Maint. Rep	10-15 years	No	No	Yes	Yes	6	6	Quick Find	7	Yes
12	2/27/2024	Maint. Rep	10-15 years	No	No	Yes	Yes	6	6	Quick Find	6	No
13	2/27/2024	Maint. Rep	10-15 years	No	No	Yes	Yes	7	7	Quick Find	7	No
14	2/27/2024	Maint. Rep	10-15 years	Yes	No	Yes	Yes	7	7	Quick Find	7	No
15	2/27/2024	Maint. Rep	10-15 years	Yes	Yes	Yes	Yes	7	7	Quick Find	7	No
16	2/27/2024	Maint. Rep	10-15 years	No	No	Yes	Yes	7	7	Quick Find	7	No
17	2/27/2024	Maint. Rep	10-15 years	Yes	Yes	Yes	Yes	7	7	Quick Find	7	No
18	2/27/2024	DMM	5-10 years	Yes	Yes	Yes	Yes	7	7	Quick Find	8	No
19	2/27/2024	Mech	10-15 years	Yes	Yes	Yes	Yes	7	7	Quick Find	10	No
20	2/27/2024	Mech	5-10 years	Yes	Yes	Yes	Yes	7	7	Quick Find	9	No
21	2/27/2024	Mech	5-10 years	Yes	Yes	Yes	Yes	6	6	Quick Find	8	No
22	2/27/2024	Mech	5-10 years	Yes	Yes	Yes	Yes	6	6	Quick Find	10	No

7.2

Id	Q11	Q12	Q13
1	No	Amazing App	9
2	No	User friendly, requiring little training and very quick	10
3	No	Positive, good work. Very useful.	10
4	No	Great!	10
5	No	A necessary step in inventory management	6
6	Yes	Good tool	5
7	No	Good	9
8	No	Ok	7
9	No	Good	8
10	Yes	Good	8
11	No	Not much	6
12	No	None	6
13	No	Ok	7
14	No	Nice	7
15	No	Ok	7
16	No	Ok	7
17	No	Good	7
18	No	Ok	7
19	No	Good	7
20	Yes	Good	7
21	No	Fair	6
22	Yes	Positive	7

7.3

Id	Q14	Q15	Q16

1	Stores keeper app is more reliable and more accurate saving time	9	Great idea
2	Quicker, less burdensome, simple and not requiring special office device.	9	Same as these previously mentioned.
3	Much better.	10	I'd like to see more emphasis on it use.
4	Much better. More over sight.	10	Excellent solution!
5	A more encompassing view and tracking system for parts and receipting	8	A great start to a troublesome issue
6	Improved	7	I need to use it
7	Better	10	Fine
8	Quicker	8	Thank you
9	Improved	8	None
10	Better	8	Good work
11	Prettier	5	None
12	.	6	None
13	Same	7	None
14	Similar	6	Ok
15	Better	6	Nothing to say
16	Newer	6	Thanks
17	Better	7	.
18	Better	7	Integration with SAP
19	Improved	7	Doesn’t integrate with SAP. Also more training needed
20	Doesn’t transact in SAP	7	Should correlate with SAP
21	.	6	More information needed
22	Different. More intuitive	7	Everybody should use it.

7.4

7.5

12. Focus Group at Time 1