Smart Inventory Management Optimization (SIMO)

Project S34 Review 2 Report

1004234	Justin Peng Zheng Wei
1004327	Hour Youlinsery Devid
1004343	Chester Lim Zi Hao
1004497	Chen Ken, Olivia
1004523	Cui Wenqi
1004526	Sun Kairan
1004609	Chia Yu Ying

SIA ENGINEERING COMPANY

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Executive Summary

Singapore Airlines Engineering Company (SIAEC) is Asia's foremost maintenance, repair and overhaul (MRO) provider and delivers integrated solutions to a large client base consisting of both international airline and aerospace companies ("SIA Engineering Company",n.d.). In this project, we are tasked to investigate and implement various innovative methods to improve on their current provisioning forecasting standards, whilst keeping the software user-friendly and sustainable. In this report, we included a breakdown structure on this project, namely Operations Research and Analysis and Web Application Design. Throughout the report we explored each of these components in depth using online papers and examples which are relevant to our project.

Under Operations Research and Analysis, we further decomposed this component into Inventory Optimization and Price Monitoring System. For Inventory Optimization, we looked at various ideas and methods that we can use when considering factors which affect important key performance indicators (KPI) such as optimizing, forecasting and active tracking of their various aircraft components whilst ensuring that the specific constraints for the customer fleets are met.

The Price Monitoring System will be the second part of our Operations Research and Analysis component whereby we explore the various ways to acquire the latest prices quoted by the vendors and formulation of links and relationships between key variables that affects performance levels of each aircraft component, following which blending them together to provide SIAEC a tool to help with an overall evaluation to decide the best price-for-value aircraft component to purchase.

For our web application we will be going through the various design considerations that we made when planning the software components of our web application. The web application would consist of a frontend interface which the users would interact with, a backend server that handles the application's logic and a database which stores the data required for the web application to work. We have also evaluated the various frameworks and database services using certain criteria such as performance, applicability of features and ease of implementation.

Table of Content

Overview

1. Company Background

SIAEC provides frontline maintenance services to more than 60 airlines[1] that fly through Singapore whilst ensuring a high level of punctuality for their customers' flight takeoffs. We are working with the Asset Management Department whereby their focus is to collate data from the various spare part vendors (Lim, E, 2022), after which the asset management department would access the orders from the various customers to provision them with the various aircraft components that they need to be replaced or renewed. SIAEC provides a wide range of aircraft parts, and they are divided into mainly repairables, rotables and expendables (International Air Transport Association, 2015). For our project, we will be focusing on optimizing rotable inventory.

2. Problem Analysis

SIAEC's current[2] methods of managing inventory mainly focuses on maintaining high customer service level but does not consider minimizing inventory cost. (Detailed explanations of how SIAEC currently handles inventory management can be found in Annex A – 1) Our proposed methods aim to minimize inventory cost while maintaining customer service level. Despite efforts by SIAEC in deploying the best inventory management practices, the company still over provisions rotables and spends unnecessary capitals. The underlying cause of this overprovisioning includes unoptimized inventory management practices, inefficient forecasting technique, and lack of a customized[3] interface in managing inventory.

3. User Requirements

We have identified three groups of stakeholders (Annex B - 1): primary, secondary, and tertiary stakeholders. Primary stakeholders are those that are involved directly in using our web-based application which is the asset management department. Our secondary stakeholders are the repair shop department and finance department. For the tertiary stakeholders, we have passengers and crews. We only focus on our primary stakeholders as they are the direct end users (Annex B - 2). We generate empathy maps (Annex B - 3) via interviews[4], and together with SIAEC we have established a 4-step milestone which is as shown in Table 1, arranged in descending order of importance (Table 1 and 2 below). For example, asset management is the most important milestone where the only constraint is the potential lack of contextual data[5].

---

[1] Some companies include TigerAir, SilkAir, Scoot, Singapore Airlines

[2] Detailed analysis on SIAEC's current methods on inventory management can be found in annex A

[3] Currently, SIAEC only refers to their data using various Microsoft Excel files, with no actual interface in place

[4] Approximately 10 interviews

[5] Some examples of contextual data include operating conditions of aircrafts, equipment age, etc

Milestones	Needs	Constraints
1. Asset Optimization	Recommendations on inventory stocking[6]	Potential lack of contextual data for implementing certain algorithms
	Identification of surplus inventory[7]	
	Recommendations on surplus inventory[8]	
	Applicability of inventory pool to current customer fleet size[9]	
2. Asset Management	**Data visualization of current inventory pool**	
	Recommendations on user possibilities for asset recovery[10]	
3. Just in Time Provisioning	Recommend level of "active"[11] inventory to hold in stock[12]	Potential lack of contextual data for implementing certain algorithms
4. Price monitoring with Performance Measure	Provide forecast on trending market pricing based on historical data	
	Evaluation of performance of Used Serviceable Materials (USM)[13]	

Table 1. General Needs and Constraints[14]

4. Refined problem statement

Thus, this project aims to create a web-based application with dynamic features such as having variable inputs, scenario-based simulations, and data visualization to help manage and optimize rotables in hopes of a more user-friendly and cost-saving alternative for the company that fits into the company's current workflow.

---

[6] Advise SIAEC on the optimal quantity to stock up for each rotable part

[7] Determine the number of excess rotables

[8] To visualize the number of excess inventories for each type of rotables.

[9] Ensure that the recommendations are only given for relevant rotables. Relevance means rotables that are in the current contract between SIAEC and its customers

[10] Advise SIAEC which rotables should be recovered after the contract ends between SIAEC and its customers

[11] Active rotables are those that are freshly repaired and ready to be fitted on aircraft

[12] Refer to glossary terms section on what "active" means

[13] To compare the performances between different vendors that sell rotables

[14] Milestones are listed in descending order of importance based on feedback during the interview with users from SIAEC. Those underlined needs are part of data visualization and thus will be directly under the implementation stage in Review 3.

Design Direction

With user requirements and a refined problem statement, we explore all relevant design concepts from various literature reviews. We begin with concept generation as an overview of all subsystems involved and then explore each sub-system in detail.

1. Concept Generation

In concept generation (Figure 1 below), we divided our system (Smart Inventory Management Optimization: SIMO) into two main subsystems: Operations Research & Analysis (ORA) and Web-app Design.

In ORA, we will look in detail into two components: how rotables inventory at SIAEC can be further optimized and how the prices of rotables can be monitored. We will apply a Pugh chart to all methods mentioned in each component in the ORA subsystem as part of the concept selection technique.

Similarly, in the Web-app Design, we will be going into details of the 4 components: software architecture, frontend, backend, and database. We also used Pugh charts to evaluate our choices for each component in the Web-app Design subsystem.

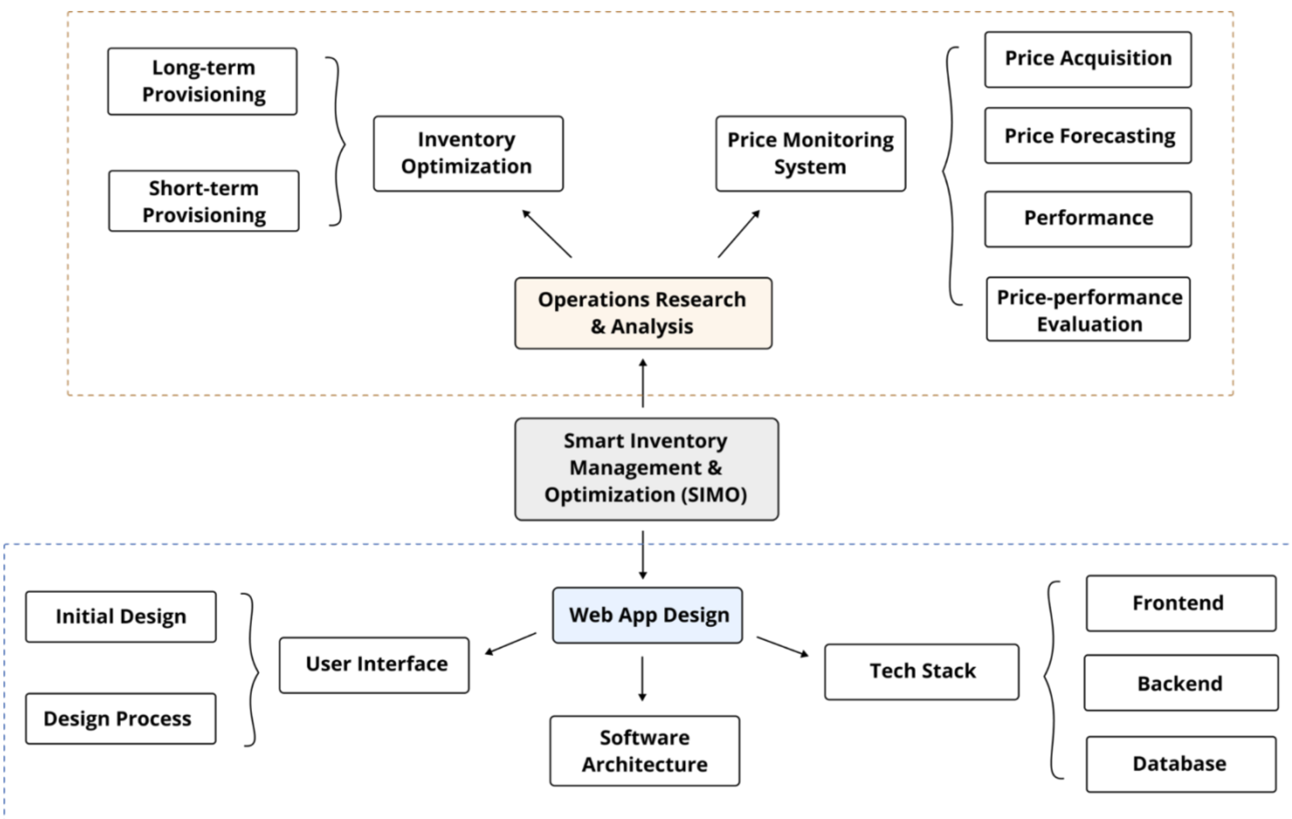


Figure 1: Generation of Concept

2. Operations Research & Analysis

2.1. Inventory Optimization

From our problem analysis in section II, we have looked at the overview of SIAEC's current inventory management. After close examination on how SIAEC manages their inventory through conducting interviews[15], we gathered some insights to further improve the way SIAEC manages their inventory. Each insight comes with a proposed approach that we will be taking to further optimize SIAEC's inventory management. For a more detailed insight description, please refer Annex A under Insights & Approaches.

	Insights	Approaches
A	SIAEC only requires rotables in the same class to be on average the required service level and their current method results in the overprovisioning[16] of rotables	Formulation of an optimization model, which takes into consideration the cost and availability of rotables
B	MTBUR[17] may not a good estimation of the time a spare part can last on an aircraft	If we can accurately predict MTBUR, we will be able to know when a demand will likely occur and hence provide supply for it
C	SIAEC assumes each spare part fails according to Poisson distribution	Explore more about the other demand distributions
D	SIAEC currently uses moving averages for short-term forecasting	Further explore other forecasting techniques that are more suited for spare parts to help improve short-term forecasting method

Table 2. Insights and Approaches (More in-depth details on each insight and approach can be found in Annex A - 2)

We will be elaborating on each of these approaches (A-D) in sequential order in section 2.1.1 and 2.1.2. In the overview of our proposed Inventory Optimization Process (Figure 2 below), we have split our proposed inventory optimization process into two portions, mainly the long-term provisioning and the short-term provisioning of rotables. Approaches A, B and C as mentioned in section 1 aims to improve the long-term provisioning while Approach D aims to improve short-term provisioning. Each of the processes as shown in Figure 2 will be explained in detail in section 2.1.1 and 2.1.2.

---

[15] Approximately 10 interviews

[16] There exists overprovision due to the current method that SIAEC is provisioning. In the event that SIAEC requires a group of rotables to be at 95% service level, they would ensure that each part number achieves at least 95% and not the average. This causes the average ESS level to reach significantly higher than 95%, potentially incurring additional costs for provision

[17] Mean Time Between Unscheduled Removal: Average duration a rotable can last on an aircraft without failing

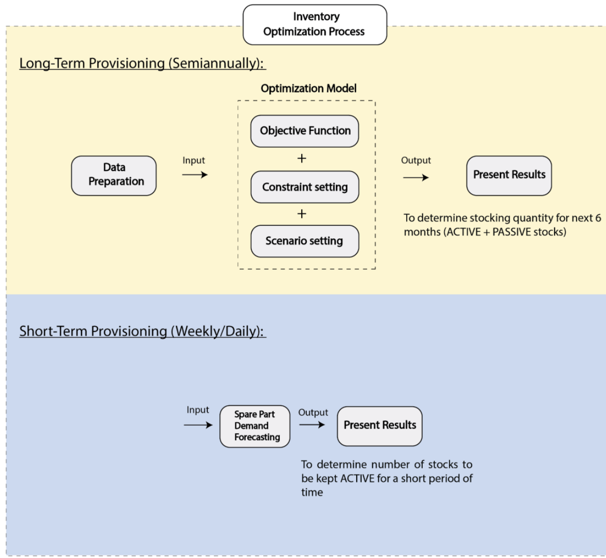


Figure 2: Overview of our proposed Inventory Optimization Process (bigger picture shown in Annex A - 3)

2.1.1. Long-Term Provisioning (Semiannually)

Our approach to optimizing long-term provisioning is our first milestone under asset optimization.

A. Data Preparation

We would need to first sift out those parts that are no longer applicable[18] to the inventory pool using data cleaning tools in Python (Pandas) to avoid stocking up on rotables that are no longer applicable to SIAEC's current customer pool.

B. Optimization Model (Formulation)

To tackle the issue as mentioned in insight A in section 1, the formulation of an optimization problem, specifically an Integer Binary Program could be used (Michael MacDonnell, & Benjamin Clegg, n.d.). We take into consideration the relative cost of each spare part and the availability of parts to ensure that the overall average service level for each class of rotables is met at the lowest total inventory cost possible.

Our optimization model has an objective function, a set of inputs, constraints, and outputs. On top of that, we have also included a scenario setting in our optimization model, to allow users to modify the input of the data and conduct relevant scenario analysis.

Scenario analysis allows our users to examine and evaluate possible outcomes that could take place in the future, which allows SIAEC to manage risks more proactively by assessing the impact of the potential future unforeseen circumstances. For example, with the lift of COVID restrictions, it is likely that demand will begin to pick up in the aviation industry (SeeKit, 2021). Hence, performing scenario analysis on post-COVID can help SIAEC to potentially predict the outcome of the future~~,~~ allowing them to better manage risks and demands.

[18] Applicable arotables are parts that SIAEC would need to potentially supply for its customer (as stated under the contractual agreement between them)

In the aviation context, two main parameters, “Flight Hours (FH)” and “Number Of Aircrafts (NOA)” are used in setting different scenarios. Hence, we will have the option for users to alter these 2 variables on the web-app when needed, before running the optimization model.

The formulation of our optimization model[19] is as follows:

Optimization Model (Formulation):

- **Objective**
  - To minimize the total cost to purchase the required rotables
- **Potential Inputs**
  - Number of Aircraft
  - Quantity Per Aircraft
  - On-Hand Inventory
  - Flight hours
  - Unit Price
  - Turnaround time[20] (TAT) [SIAEC requested this to be a user-input value]
  - MTBUR OR MTBSR[21]
- **Constraints**
  - Average customer service level for each class of rotables are to be met
  - Total cost incurred should not exceed SIAEC’s budget
  - Some airline customers require at least a certain number of specific types of rotable to always be available in the warehouse
- **Output**
  - Top up quantity for each rotable type
  - Required service level for each class
  - Total cost incurred
- **Scenario Setting**
  - Users can modify the FH and NOA to examine how the outputs will change

By formulating our optimization as an integer binary program (Annex B - 4), we have tried-solving our optimization model using *lpsolve ()*[22] package in R, using a small test size of 5 rotables, all of which belonging to ESS2 (i.e., Targeted average service level of 95%). To simplify our model, we have assumed on hand-inventory to be zero for this experiment. The results generated (Annex B - 5) is a good indication that we are in the right direction in solving the over provisioning issue in SIAEC as seen by the lowered inventory cost while maintaining service level well above 95%.

Even though we used R programming for this test experiment, our actual optimization will be implemented in Python. By formulating this optimization model above will be our approach A in optimizing inventory in SIAEC as mentioned in Table 2.

C. Further Improve our Optimization Model

With the formulation of our optimization model as the backbone of long-term provisioning, we have looked into ways in which we can potentially improve the outputs of our model, by increasing

---

[19] We have also considered the time complexity of the model which is $O(n^2)$. However, considering our data size of around 4000 rows, we will try to reduce the time complexity further by using techniques such as multi-threading.

[20] Turnaround time includes the repair, transportation and logistics time

[21] Mean Time Between Scheduled Removal: Average duration a rotable is scheduled for check

[22] Ipsolve() is a package that is used for solving linear, integer and mixed integer programs

the quality of our input data (section 1 below), as well as how we can better model the failure times of rotables (section 2 below).

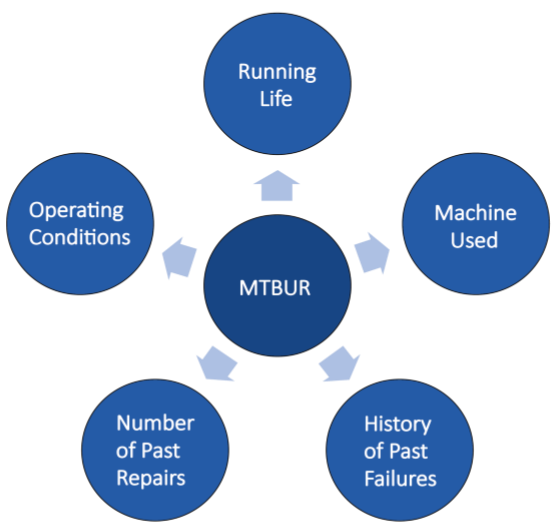
1) Predicting MTBUR using Machine Learning & Simulation

From the set of inputs of our optimization, we noticed that most of the parameters that SIAEC currently considers, such as QPA, FH and NOA are fixed variables (i.e., cannot be better predicted). The parameter TAT was also requested by SIAEC to be set as a user-input variable, which means there is no need for us to predict the TAT. Hence, the only input variable that we can better predict would be the MTBUR.

Majority of the MTBUR that SIAEC is currently using are very generalized as they are aggregated across all the aircrafts in the world. Hence, this generalization of MTBUR is often skewed, as they do not consider the flying patterns[23] of SIAEC's pool of customers, which might potentially lead to inaccurate calculations. Only a minority[24] rotables have historical MTBUR available. Hence, we hope to predict MTBUR more accurately for those parts without historical MTBUR. The two state of the arts methods that we have found that are used to predict MTBUR of rotables are through the use of machine learning and simulation.

Machine Learning:

From our research, we have found that the use of machine learning techniques, specifically the use of supervised learning algorithms, are quite commonly used in predicting MTBUR (Wilkinson, G., 2020). The use of a hybrid data preparation, specifically using feature selection tool, reliefF, for attribute evaluation followed by using K-means algorithm for data noise elimination, before training the data with machine learning algorithm was proved to be effective in predicting MTBUR. (Celikmih, Inan, & Uguz, 2020). Some commonly used factors to predict MTBUR that we have found are shown in Figure 3 below. On top of this, some of the supervised methods (Jigsaw Academy, 2022) that we found could potentially be used to predict MTBUR are shown in Figure 4 below.



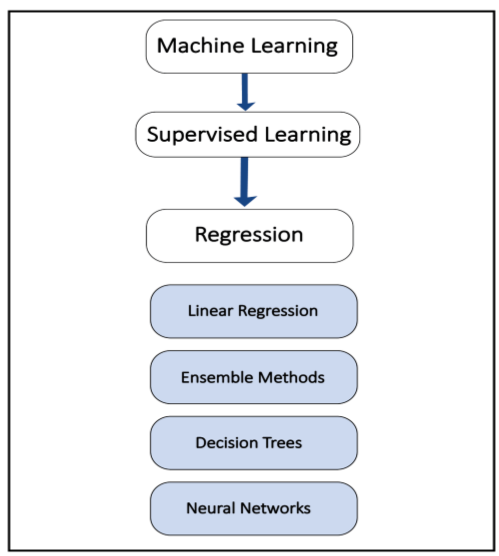


Figure 3: Factors affecting MTBUR (left) Source: Adapted from (Wilkinson, G., 2020)
Figure 4: Overview of supervised learning algorithms (right)

[23] For example, airplanes that frequently fly to more warm areas might corrode faster (hence lower MTBUR) as compared to cold climates. (Lim, E, 2022). Depending on different flying patterns, MTBUR might vary very differently for the same rotable part.

[24] Approximately only 14% of rotable parts (out of ~4000 rotable parts) have historical MTBUR available

Despite the usefulness of machine learning, the use of machine learning is not feasible due to the lack[25] of contextual data as shown in Figure 3, the training of a machine learning model to predict MTBUR is not feasible for our project.

Simulation:
From our research, we have found that even though the use of simulation to predict MTBUR produces accurate results, one needs to have a very good understanding of the dynamics[26] of the system which can then be modeled using contextual data (Kong & Lai, 2007). As mentioned, important contextual data required to predict MTBUR which includes aircraft operating conditions, history of past failures etc, as shown in Figure 3 above. With the lack of such data, it impedes us from using simulation to predict MTBUR.

Hence, we will not be pursuing the idea of better predicting MTBUR.

2) Statistical Distribution of Demand

As the accuracy of the outputs of our optimization model are only as accurate as how we model[27] the failure times of rotables, we have also investigated ways in which we can improve the way we can model the failure times of rotables. As mentioned in Table 2, SIAEC currently assumes that all rotables fail according to Poisson distribution.

From our literature review, we have found that the use demand classifications are widely used in many fields of production and operations management, including aircraft spare part inventory management (Chawla & Miceli, 2019). The categorization of demand patterns facilitates the selection of forecasting techniques that is best suited for each group of products. (Syntetos, Boylan, & Croston. 2017) Demand is commonly classified into 4 different classes (Figure 5): Smooth, Erratic, Lumpy and Intermittent. (Syntetos, Boylan, & Croston, 2005) The method used to classify demand patterns is based on two parameters: average demand interval (ADI) and the square of coefficient of variation of demand (CV^2). According to Chwala & Miceli (Chawla & Miceli, 2019), Demand is considered smooth if (ADI<1.32, CV2<0.49), intermittent if (ADI>=1.32, CV^2<0.49), erratic if (ADI <1.32, CV^2>=0.49), and lumpy if (ADI >=1.32, CV^2>=0.49).

Extending on the idea of the classification scheme above, Syntetos, Babai, Lengu and Altay (Syntetos et al., 2011) established an inductive rule that proposes suitable distributions for different demand classes. Refer to figure 6 below, the rule is based on the 2 same dimensional factors: ADI and CV^2. For example, normal distribution is used with demand for rotables that have low ADI and CV^2 while Poisson is used with demand for rotables that have low CV^2. Gamma is for demand of rotables that have high ADI and CV^2 while Negative Binomial Distribution and Stuttering Poisson are for the rest of the regions in Figure 6.

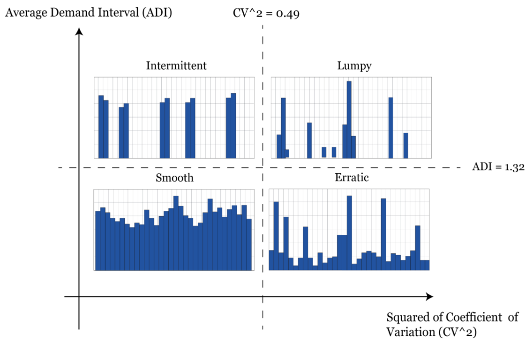
By classifying the rotables by their ADI and CV^2 according to Figure 6, suitable distribution can then be applied to each group.

---

[25] As highlighted in Section III. User Requirements under the constraint in milestone 1

[26] For example, we will need a very clear understanding of the flight operations

[27] The average customer service level from the constraint of our optimization model is calculated Poisson Distribution as SIAEC currently assumes all rotable fails according to Poisson Distribution



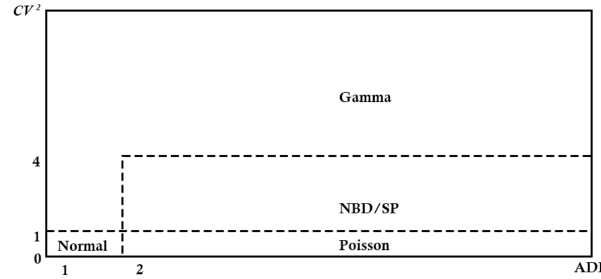


Fig 5: Rotable classification into 4 main groups (left) (Bigger picture shown in Annex A - 4)
Source: Adapted from (frepple APS. frePPLe., 2021)
Fig 6: Demand Distributional Assumptions (right) (bigger picture shown in Annex A - 5)
Source: Adapted from (Syntetos, Babai, Leng, Altay, 2011)

2.1.2. Short-Term Provisioning (Monthly)

Short-term provisioning, which is under “Just in Time Provisioning”, will be our approach D as mentioned in Table 2. Total quantity of rotables calculated from the Long-Term Provisioning includes both active[28] and passive[29] rotables. Active rotables allow for optimal condition when fitted on aircraft while passive rotables are those that are kept frozen in the cold room to preserve the parts by slowing down the deterioration rate (Lim, E, 2022). Hence, it is imperative that there is short-term provisioning to determine active rotables needed in a short timeframe. A detailed methodology used to solve for short-term provisioning will be described in the following.

In short-term provisioning, we use the historical issuance data to forecast. Historical issuance data is the past demand data for rotables. It is challenging to forecast demand for rotables as the demand draws a non-smooth pattern with mostly zero values and random non-zero demands (Sahin, 2021). Methods such as moving average and exponential smoothing are biased in the presence of such a demand pattern (Turrini, & Meissner, 2017). Figure 7 below shows various methodologies proposed for forecasting of rotables. There are three major streams branching out of spare part demand forecasting: Time Series Forecasting, Contextual Information and Forecast Improvement Strategies. Time Series forecasting methods are relatively easy to implement (Pinçe, Turrini, & Meissner, 2021). They rely mainly on historical data without contextual information (e.g., expert judgments, product characteristics, maintenance information) to identify the drivers of rotables demand (Pinçe, Turrini, & Meissner, 2021). Contextual Information method, however, takes into account information such as maintenance schedules, equipment age, and weather conditions (Pinçe, Turrini, & Meissner, 2021). Due to the unavailability of contextual data (i.e. equipment age, operating conditions, etc), we will only consider the time series approach for our spare part demand forecasting while using demand classification (see section 1.1.4) as the forecast improvement strategy to first understand the underlying demand characteristics before assigning the suggested time series approach by literature review. Time series can be divided into Parametric and Non-parametric and both methods will be discussed in detail below.

[28] Active rotables are those that are freshly repaired and ready to be fitted on aircraft

[29] Refers to current on-hand inventory that is kept as back up and in unserviceable state to be stored in the cold room and will only be brought out to convert into serviceable state when required.

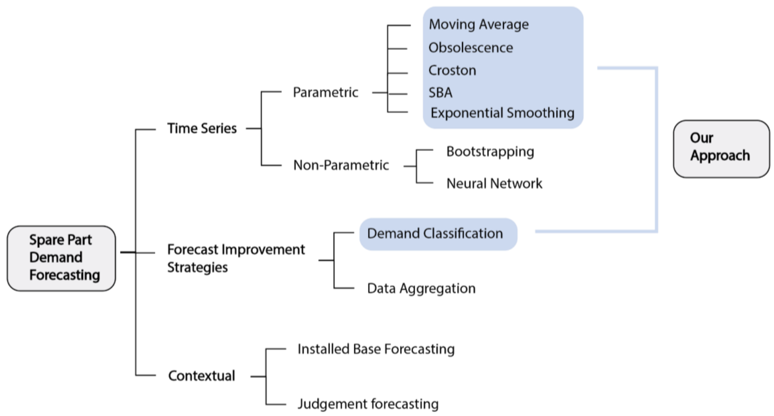


Figure 7: Spare Part Demand Forecasting Methods (Bigger picture shown in Annex A - 6). Source: Adapted from (Pinçe, Turrini, & Meissner, 2021)

Croston Method: In recent years, attention has been given to intermittent demand forecasting and significant advancements have been made (Pedrosa, Conceição, 2011). The use of Single exponential smoothing (SES) technique was inappropriate for use on items with intermittent demand (Pedrosa, Conceição, 2011). Croston was the first paper to investigate a new forecasting method for intermittent demand (Pinçe, Turrini, & Meissner, 2021).

Syntetos–Boylan approximation (SBA): There is an error in the mathematical deviation of expected demand size in Croston's method as reported in Syntetos and Boylan paper (Syntetos and Boylan, 2005). There is a bias associated with Croston's method and a modification is needed on the original estimator of mean demand in Croston and this modification later come to be known as the Syntetos–Boylan approximation (SBA), yields an approximately unbiased estimator" (Gutierrez,2007).

Obsolescence method: In the context of rotables forecasting, demand obsolescence refers to a spare part that is not needed or demanded anymore (Pinçe, Turrini, & Meissner, 2021). Sudden obsolescence occurs, for example, when a facility that uses a part is unexpectedly moved or out of service, or when a new version of a part is replaced by an older version due to an upgrade (Pinçe, Turrini, & Meissner, 2021). This operational environment coincides with that of SIAEC where they would search for applicable rotables that are in the warehouse. Hence, this method will be considered part of our data cleaning process in the above section.

Bootstrap Method: Another way to address the sporadic need for rotables is the bootstrap method. "The methods proposed by (Willemain, 2004) are widely accepted and provide a more accurate forecast of demand distribution over a given lead time than exponential smoothing or Croston methods" (Pedrosa, Conceição, 2011). Willemain modified the traditional bootstrap method to better model intermittent inventory data with autocorrelation, frequently repeated values, and relatively short time series (Pinçe, Turrini, & Meissner, 2021). A Markov model is used to assess the probability of an empirical transition between zero and non-zero demand for various items in order to estimate information about demand during lead times (Pedrosa, Conceição, 2011). However, this method is computationally intensive and complex with marginal improvement in accuracy (Turrini, & Meissner, 2017). Therefore, it is unlikely that we will consider the bootstrapping method in our project.

Neural Network: Other development categories of non-parametric methods are machine learning methods, and the general idea behind these methods is to learn directly from data using algorithms such as neural networks (Pinçe, Turini, Meissner, 2021). Neural networks are versatile tools that can detect nonlinear patterns in data (Pinçe, Turini, Meissner, 2021). Data availability is the key to developing machine learning models (Rožanec, & Mladenic, 1970). This is a company's limitation, so it is unlikely that you will use machine learning technology.

We come up with a Pugh chart below to select our preferred design concepts based on 4 design criteria. From the total score, we will consider Parametric and Forecast Improvement Strategies (i.e. Demand Classification) in our implementation stage. In Parametric, we will consider the Croston method, Syntetos–Boylan approximation and Obsolescence method to forecast our demand together with moving average and exponential smoothing methods. To select one of the methods in Parametric, we will be trying out each of the methods on each demand types (i.e. Intermittent, Lumpy, Smooth and Erratic) and measure its performance before selecting the one best method. Using demand classification with appropriate forecasting techniques above will be our approach D in optimizing inventory in SIAEC as mentioned in Table 2.

	Alternative Design Concepts	
Design Criteria*	**Parametric**	**Non-parametric**
Data Requirement	+	0
Accuracy	0	+
Time	+	0
Ease of implementation	+	0
Totals	+++	+

** Further explanation of the Criteria used and assessment rubrics can be find in Annex A - 7*

2.2. Price monitoring system

Our approach to the price monitoring system is our fourth milestone under "Price monitoring with Performance Measure". Price monitoring or price intelligence refers to the awareness of pricing in the market and the response to these changes in pricing. However, the process that actually drives this system would be the metasearch engine. This works by sending queries through multiple search engines and aggregating these results. There are several processes that are involved such as ranking the query results and fusion which is a process that filters out the information collected and presents only information that is relevant (Ofiwe, 2021). Some examples of companies utilizing this technique include Trivago and Skyscanner.

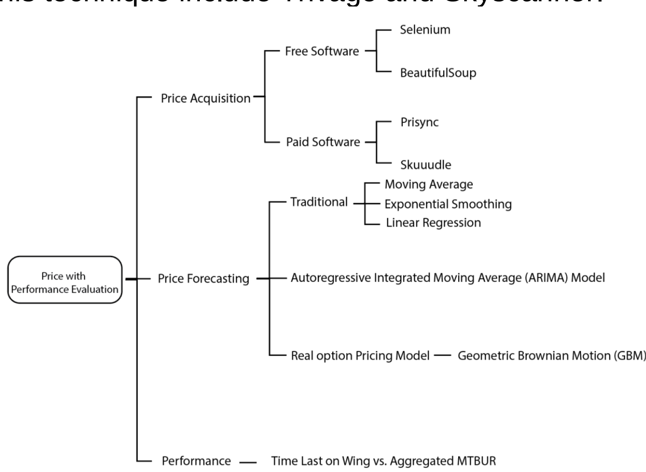


Figure 8: Price to Performance Evaluation Breakdown (Bigger picture shown in Annex A - 8)
Source: Adapted from (Wattanarat, Phimphavong & Matsumaru, 2011) & (Ofiwe, 2021)

We break the price monitoring system down into price acquisition from vendors, price forecasting and performance tracking. We will then use a combination of pricing and performance to help SIAEC conduct price-to-performance analysis to identify best parts to purchase from the respective vendors.

2.2.1 Price Acquisition

Web scraping is a necessary and relevant tool for us to be able to obtain the latest price that vendors quoted for SIAEC on each of their rotables. Under this, we have explored 2 options for web scraping in Python which is the library BeautifulSoup and Selenium. We mapped this in a pugh matrix to look at the best softwares or library for us to use in the implementation stage.

Laying out the Pugh matrix on three factors (Cost, Difficulty, Adaptability), we have decided to use Selenium as our choice of web scraper because Selenium is able to be used in more situations and is able to interact with dynamic pages and content (Pornaras, 2021). Since scraping speed is not a significant factor we decided to go with Selenium

	Alternative Design Concepts			
Design Criteria	**Skuuudle**	**Prisync**	**BeautifulSoup**	**Selenium**
Affordability	-	0	+	+
Ease of Integration	-	0	+	+
Adaptability	+	+	-	0
Total	-	+	+	++

** Further explanation of the Criteria used and assessment rubrics can be find in Annex A - 9*

2.2.2 Price Forecasting

After acquiring our latest prices through the use of web scraping, we now look upon the forecasting of these prices to help SIAEC better understand the future price movement patterns and help them to judge on the relative prices for each of their aircraft rotables that are provided by the vendors. Like in short-term provisioning in section 1.2, prices are difficult to forecast due to their nature of uncertainty. If we factor other components such as realistic economic factors, nature of competition for prices and latest technological advancement considerations, the model will be more robust and long lasting across a long period of time.

Looking at the research methodologies that had been done in the past, we will look specifically at the price forecasting models for strategic planning in the supply chain. Here, supported by Figure 9, it breaks down the price forecasting methods into mainly Linear Regression, Moving Average, Real Option Pricing which mainly contain the Geometric Brownian Motion (GBM), and the Autoregressive Integrated Moving Average (ARIMA) Model (Wattanarat, Phimphavong & Matsumaru, 2011). As mentioned under short-term provisioning, moving average is not recommended for use as it is considered as a primitive model whereby the methodology tends to be biased especially so in the event where the prices of the aircraft components fluctuate over time (Schmitt, n.d.). The use of exponential smoothing such as simple exponential smoothing has its drawbacks that it relies on non-seasonal patterns and, it does not take into account corrective information being added which potentially makes the error grow exponentially (Ostertagova & Ostertag, 2012). We have omitted linear regression as this methodology is highly sensitive to outliers and hence any changes in prices will heavily shake the regression model for our price forecasting (Flom, 2018). As such, we will focus our exploration on the ARIMA Model and GBM.

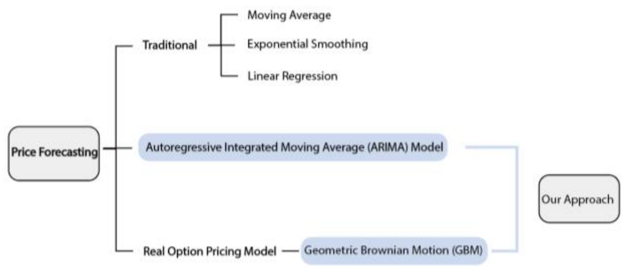


Figure 9: Price Forecasting Breakdown (Bigger picture shown in Annex A - 10)
Source: Adapted from (Wattanarat, Phimphavong & Matsumaru, 2011)

Autoregressive Integrated Moving Average (ARIMA) Model: This model is well-known in both stochastic modeling as well as econometrics and is a mathematical model mainly utilized in the investigate on the forecasting of time series onto factors such as pricing and demand by fitting data into the model (Wattanarat, Phimphavong & Matsumaru, 2011). This method can also be seen as a combination of having a trend model coupled with random walk (Wattanarat, Phimphavong & Matsumaru, 2011). ARIMA incorporates both auto regression whereby it fits the current data points into a linear function and moving average by adding several consecutive data points to obtain their mean, combining it to obtain estimates for its forecasting (Wattanarat, Phimphavong & Matsumaru, 2011).

Geometric Brownian Motion (GBM): This model originates from the field of finance where it is mainly used to calculate the forecasting the value of the returns of a stock. The core of this model anchors on the fact that the logarithm of an underlying asset's price follows a random walk (Wu & Buyya, n.d.), an economic terminology whereby changes in prices in the asset follows the same distribution and are independent of one another and does not take into account prices taking a random and unpredictable path. This means that the changes in price are uncorrelated with historical prices and there will not contain patterns to these changes of prices (Ibe, n.d.). As such, GBM can be considered as a form of random walk theory with mean reverting properties (Wattanarat, Phimphavong & Matsumaru, 2011).

Therefore, with the exploration above through various literature reviews, we come up with a Pugh chart below to select our preferred design concepts based on 4 design criteria. From the total score, we will exclude Traditional methods such as Moving Average, Exponential Smoothing and Linear Regression, whilst only considering Geometric Brownian Motion and ARIMA our implementation stage.

As researched in our literature analysis, it is shown that GBM has an absolute standard deviation of 0.06%, compared to ARIMA which has 0.6% standard deviation[30] which would be a strong backing for us that GBM would more likely be the stronger model to help in our price forecasting (Wattanarat, Phimphavong & Matsumaru, 2011).

[30] Detailed graphs and results from to be shown in annex

Design Criteria*	Alternative Design Concepts	
	Geometric Brownian Motion	ARIMA
Data Requirement	+	+
Accuracy	+	0
Time	+	0
Ease of implementation	0	+
Totals	+++	++

** Further explanation of the Criteria used and assessment rubrics can be find in Annex A - 11*

2.2.3 Vendor Performance Evaluation

We aim to analyze and evaluate the performance of the part numbers across each vendor. Currently we have a wide variety of information that we can utilize as a measure for the performance of the various part numbers. Out of the few, we identified that the Time Lasted on Wing (TSI) is a suitable variable for us to consider as this tracks the duration at which the part is fitted on the operational aircraft until it is in defect.

One of the most similar comparisons to this would be the aggregated MTBUR that is given by Airbus as MTBUR here is standardized internationally by aircraft manufacturer companies such as Airbus and this serves as a benchmark for the performance of the parts that SIAEC uses for their customer fleets. MTBUR has a very similar logic as the TSI recorded by SIAEC for each of the vendors. To compare, we aim to do a ratio between the TSI for each vendor on each part number to the aggregated MTBUR that is issued by the airline operators, of which this will have direct proportion to the performance of the part across each vendor.

2.2.4 Price-Performance Evaluation

Combining price and performance, we now have a view on the price of each of the part numbers and the TSI across each vendor. The method that we drafted would be to take the ratio of the pricing given by the OEMs[31] and the MTBUR used by the airline companies as a ratio benchmark. Similarly, we will obtain ratios of the prices quoted and the TSI that was obtained by each vendor through both the web scraping process and historical records from SIAEC respectively and comparing this with the benchmark that we have set previously. From there, we are able to get a clear and simplistic comparison between the vendors.

[31] Original Equipment Manufacturer

3. Web Application Design

3.1. Decision on Software Architecture

Architecture	Advantages	Disadvantages
Client Server (*Why Use Serverless Computing? | Pros and Cons of Serverless*, n.d.)	Easy set up	Lock in with vendor
	Cheaper than physical hardware	Expensive to scale
	Security implementations by cloud provider	
	Easy network configurations	
Client Serverless (Lange, 2021)	Server management not required	Difficult to debug
	Pay per usage	Security issues
	Easily scalable	Not cost-effective for long data intensive computations

Table 3: Exploration of the various software architectures

Although a serverless architecture requires less management, it is not suitable for our project as sharing the same machines with the other users from the same cloud provider might lead to potential security issues. Besides, the pricing model of serverless architectures are by the amount of data used. Since our project is data intensive, it might incur significant cost. Even though using a client-server architecture might require more codes, the advantage of serverless architecture is not significant due to the limited error logs provided. This would possibly lead to us spending more time on debugging the codes, instead of using the time in developing the application.

3.2. User Interface

3.2.1. Initial design

Designing of Workflow of the Web-application

Taking into consideration the needs and requirements of our users mentioned in section Overview III, we designed an updated workflow for our web app (as shown in Figure 10). This diagram helps to contextualize the needs of the users into the design and provides a visualization of the user's workflow as well as the required features and functionalities.

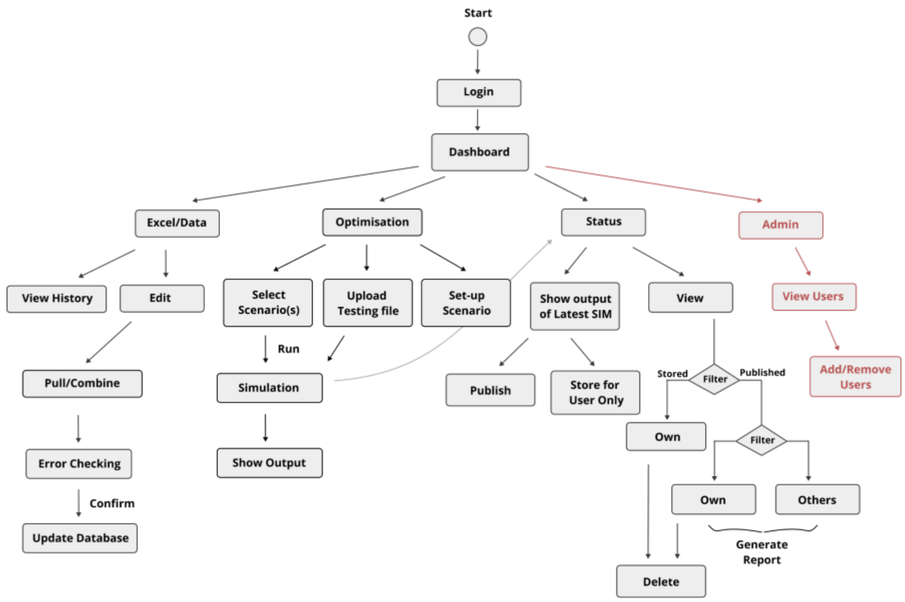


Figure 10: User Action-Flow Diagram (*Bigger picture shown in annex A - 12*)[32]

Designing of Pages of the Web-application

Following the User Action-Flow Diagram, we planned the pages of the web application based on the user actions. The diagram below helps us to visualize the transition of pages of the application and assist in planning of the features of the interface.

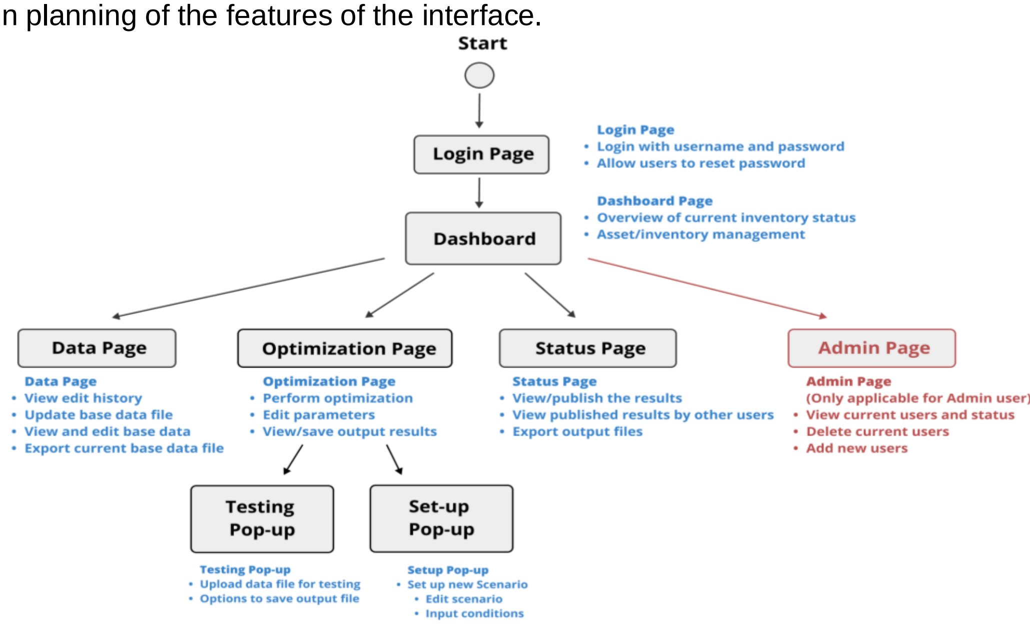


Figure 11: Page-Flow Diagram (Bigger picture shown in Annex A - 13)

3.2.2. UI design process

Based on the planning of the pages, we went through the process of selecting a design that is the most suitable for our project. Following are examples of some of the UI design aspects that we considered.

---

[32] The actions highlighted in red are only applicable to the admin user.

Dashboard View vs Scrollable View

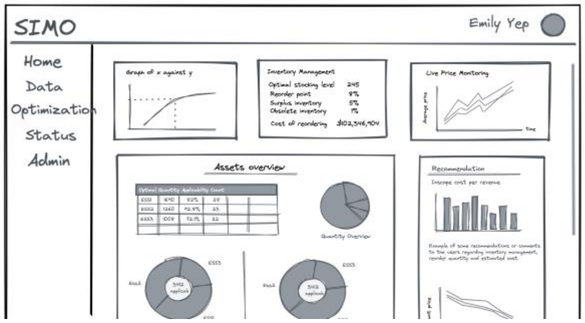


Figure 12.1: Homepage as Scrollable View

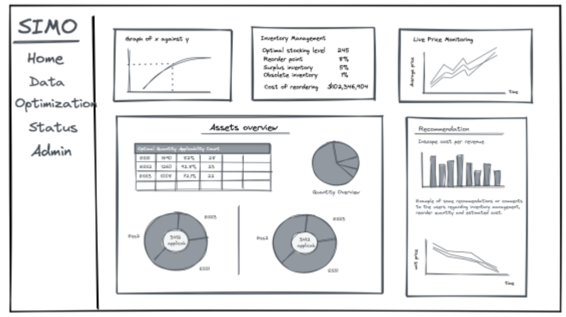


Figure 12.2: Homepage as Dashboard

Since the main purpose of the application is to perform inventory optimization and handle management, we evaluated that it is better to use a dashboard design than a scrollable view. This is because dashboard view provides a more effective way of displaying data as it serves to provide important data or results for the users at the first glance (Bakusevych, 2018).

Navigation bar

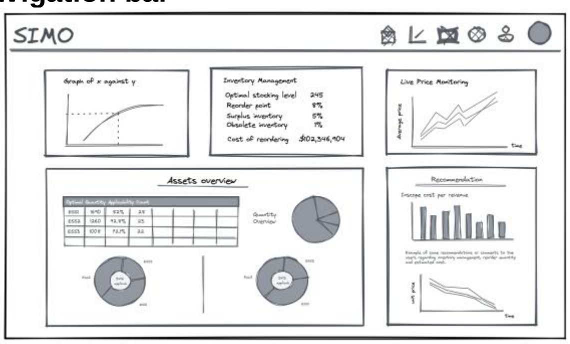


Figure 13.1: Homepage with Top Navigation Bar

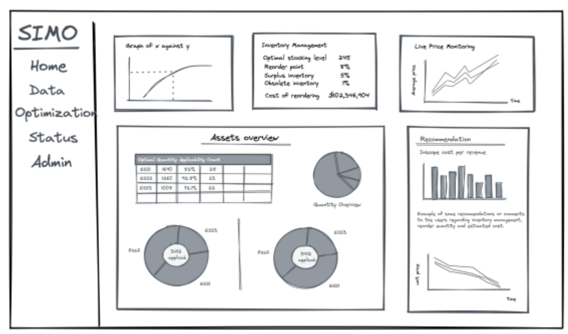
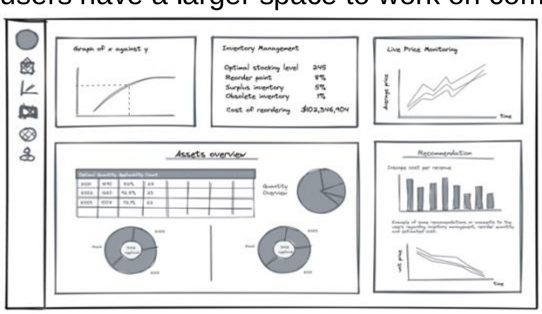


Figure 13.2: Homepage with Left Navigation Bar

Top navigation bar saves a lot of space and is more predominantly used for websites that have mega menu[33] as it organizes and provides a more structural view for the users (Bakusevych, 2021). Side bar, on the other hand, is vertically aligned and considered more intuitive for users' vision scan (Bakusevych, 2021). For our case of the application, our navigation bar is relatively simple and direct, and we would prefer the navigation bar to not take up too much space so the users have a larger space to work on complex data.



In this case, we decided to combine the advantages of both designs, and use a left bar that is collapsible while the users are performing tasks on the main screen. In this case, we are able to fully utilize the screen, at the same time, making the design across the application more consistent and intuitive for the users.

Figure 13.3: Homepage with Left Navigation Bar[34]

[33] Mega menus are a type of expandable menu where choices are displayed in a two-dimensional dropdown layout. They are designed for accommodating a large number of options or for revealing lower-level site pages at a glance (Bakusevych, 2021).

[34] Paper scketch of the full improved dashboard designs can be found in Annex A - 20.

3.3. Decision on Frontend Framework

Framework (Borrelli, 2021)	**Advantages**	**Disadvantages**
React[35]	Virtual DOM increases performance and optimizes app workloads	Extra libraries required at times
	Re-render only when needed to improve efficiency	Insufficient documentation due to constant update
	Bundling and tree-shaking to minimize the load of end user's resources	
	Support server-side rendering	
	One-way data binding to better control data flow	
	Output monitoring to help with debugging	
Angular[36]	Real-time synchronization between model and view	Heavier applications due to the amount of features, leading to slower responses
	MVC structure to allow separation of tasks to reduce load time of a webpage	Addition of new and significant changes would introduce a steep learning curve
Vue[37]	Consistently updated documentation	Relative new framework
	Fast performance	Small community

Table 5: Exploration of the various frontend frameworks (Dhaduk et al., 2020)

	Alternative Design Concepts		
Design Criteria	**React**	**Angular**	**Vue**
Ease of Learning	+	-	-
Performance	+	-	+
Applicability of Features	+	+	0
Total	+++	-	0

** Further explanation of the Criteria used and assessment Rubrics can be found in Annex A - 14*

Though all three potential frameworks are popular and have plenty of available resources and libraries, React can optimize our development time further since some of our members have prior experience. Moreover, React has sufficient features and good performance to fulfill the requirements of SIAEC and support our design ideas based on our research.

[35] An open-source framework designed to allow easy maintenance of application.
[36] A framework based on TypeScript that supports two-way data binding.
[37] A versatile framework that supports two-way data binding.

3.4. Decision on Backend Framework

Framework	Advantages	Disadvantages
Golang gin[38] (Rathod, 2021) (Gnedoy, n.d.)	Fast execution with the use of radix tree based routing and small memory footprint	No OOP which reduces the reusability of code
	Easy to create middlewares which can be plugins	Huge binaries due to the power support of virtual machine
	Able to handle the crash of HTTP request	Small and relatively inactive community
Express.js[39] (Volodymyr, 2017) (Gnedoy, n.d.)	Allows to use same language (JavaScript) for both frontend and backend, thus the app development process can be faster and easier	A single threader framework with event loop, so callbacks are needed which could be difficult to understand and maintain
	Middleware packages to support for solving development problems	Reduced performance when handling CPU intensive tasks
	Event-driven to handle multiple concurrent I/O requests with the support of Node.js	Heavy code changes due to unstable API
	Large and active community	
Django[40] (*Django Advantages and Disadvantages*, n.d.) (*Benchmark IT Solutions, n.d.*) (Bhatt, 2020)	Readymade packages to help with the implementation of features easier	Lack of code conventions
	Model View Template design pattern to provide fast development and processing	Monolithic frameworks make it harder to use external packages
	Has Content Management System with built-in Admin Interface	
	Scalable since Django is based on loosely coupled architecture	
	Large and active community	

Table 6: Exploration of the various backend frameworks

Design Criteria	Alternative Design Concepts		
	Golang Gin	**Express.js**	**Django**
Ease of Learning	-	+	0
Ease of Implementation	0	-	+
Performance	0	-	+
Total	-	-	++

** Further explanation of the Criteria used, and assessment rubrics can be found in Annex A - 15*

---

38 A popular framework from REST API that has supported libraries and is designed to handle CPU intensive tasks.
39 A Node.js framework that is adaptable and lightweight.
40 A high-level Python-based web framework that encourages rapid development and pragmatic design.

Though we are more familiar with Express.js, it does not fulfill all the needs of this project. For example, it cannot handle CPU-intensive tasks which is required by our optimization model. Golang’s ability to efficiently handle concurrency would not be significantly value-adding to our project either since the integration with the optimization model written in Python would reduce its performance. Since Django can be integrated with the model easily as it uses the same language and is able to fulfill the other requirements at the same time, we decided to go with Django.

3.5. Decision on Database Service

Framework	Advantages	Disadvantages
Firebase[41] (Guriev, 2021)	Real-time database design and file storage	Not suitable for client-server architecture due to admin sdk being limited
	Faster development time using Firebase services	No constraints on table schema
	In-app messaging and push notifications	Limited querying due to data stream model
	Integration with sign-in providers	Pricing model expensive for our use case
	Provides dynamic links and ML-kits	Has no security by default
Amazon Web Services Relational Database Service (AWS RDS)[42] (Maras, 2020)	Instant provisioning and autoscaling with Elastic Compute Cloud (EC2)	Slow with a large number of concurrent writes on same data
	High availability because of replica	Only vertical scaling which is costly
	Robust security	Queries are complex to write
	Automatic backup and recovery	
	Many Services from Amazon	
MongoDB Atlas[43] (Bruce, 2021)	Sharding for cloud-base storage	Limited document size of 16MB
	Flexible and document-based	Hard to manage relations of data
	Real-time data reporting and analytics	Relative higher chance to have stale or duplicate data
	High availability because of replica	High memory usage

Table 7: Exploration of the various database service providers (Wodehouse, 2018) (Singh, n.d.)

[41] Non-relational database offered by Google, comes with many other services.
[42] Relational Database service offered by Amazon. Easy to integrate with the many services offered by Amazon.
[43] Non-relational database service provided by MongoDB.

Design Criteria	Alternative Design Concepts		
	Firebase	AWS RDS	MongoDB Atlas
Performance	-	+	+
Applicability of Features	-	+	0
Cost-Efficiency	-	0	0
Total	---	++	+

** Further explanation of the Criteria used and assessment rubrics can be found in Annex A - 16*

Due to the need of a logical error checking feature and the nature of the dataset which is structured, a relational database would be a better choice considering its fast concurrent accessing ability and the ease of scalability. Additionally, due to the need for our web application to communicate with SIAEC's data which is in AWS Redshift. For ease and efficiency of integration with the various components, we decided to go with AWS RDS.

3.6. System Overview

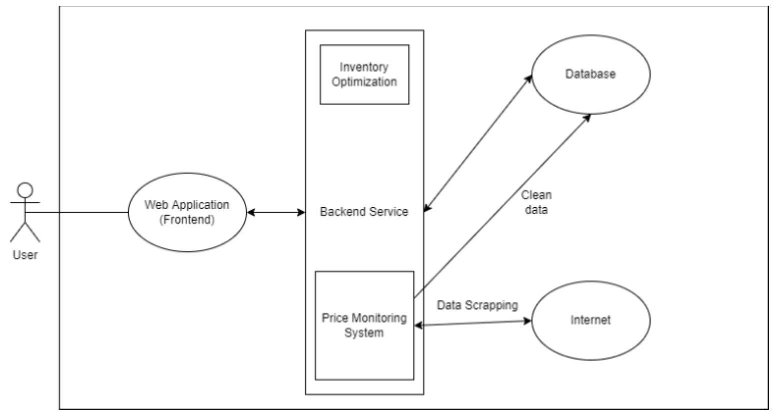


Figure 14: Overview of Interactions between Different systems (Bigger picture shown in Annex - 17)

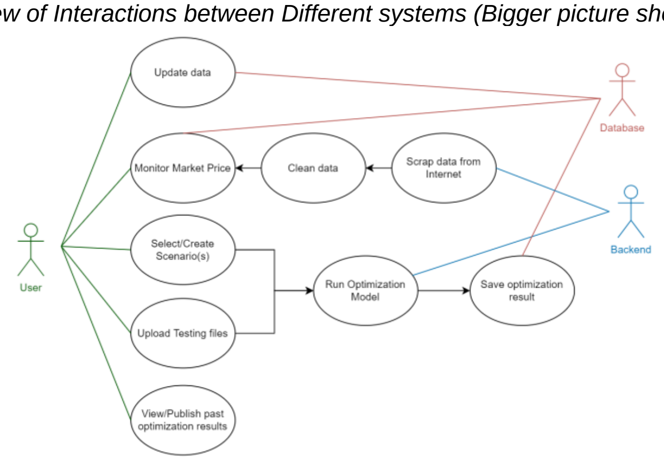


Figure 15: Overview of Interactions between Users and Systems (Bigger picture shown in Annex - 18)

With reference to figure 14 and 15, we can see that the user only interacts with the frontend portion of the system which then makes appropriate calls to the backend server that services the requests made. Only the backend server interacts with the other components of the system like the database, internet, and frontend client. The backend server is also responsible for handling the price monitoring system where data needs to be scrapped daily, cleaned and then stored in the database. The backend also encompasses the inventory optimization system which runs the optimization code using data stored in the database.

Risk Assessment

After conducting research, we decided to use the Consequences-Frequency Matrix (Van Westen, n.d.) to break down and visualize the potential effects and consequences. For the context of our project, we classify Impact as the how consequential the damage is to our software should the event happen, and ranges from Small (minimal effect) to Moderate (require significant levels of attention) to High (potential detrimental consequences to the project and SIAEC). For frequency we look at how often that particular event could potentially happen in a period of time. For low frequency, the event occurs once every 3 years and medium will be once every year. For high and very high frequency it will occur once every 6 months and 3 months respectively. The risk levels are color coded with green being low risk, yellow being moderate risk, orange being moderate-high risk and red being high risk.

		Impact		
		Small	Moderate	High
Frequency	Very High			
	High			
	Medium		Inaccurate data entry	Inaccurate Forecasting
	Low		Server failure	Security Breach Data Loss

Table 8: Consequences- Frequency Matrix

Following this, we classified some of the risk events that could potentially happen and assessed the respective potential consequences as shown in Annex - 21.

Environment, Health & Safety

Web-app needs	**Web-app constraints**
Have a clear instruction manual regarding how to use the web-app	Hard to be flexible to future changes due to the update of the used software tools and libraries
Connect with the data sharepoint of SIAEC to pull data using Redshift	Vulnerability of open-source libraries
Check error and conflicts in manual input data	Data might be out-of-date since it needs to be manually pulled from the data sharepoint of SIAEC
Be flexible and convenient to do scenario simulation and optimization	Availability of service
Import/Export optimization data	Confidentiality of user information
Can be easily maintained by users from non-technical background	

Table 9. Needs and Constraints for Web-application

Project Management

1. COVID-19 Limitations

Some limitations we foresee are team members falling ill due to Covid-19, this decreases our productivity. Other issues that we faced was that working hours are shortened due to the school's covid measurements which causes classrooms to close earlier. We resolved these issues by setting clear goals and objectives for each member to compensate for the loss in productivity. We also delegated most of the tasks so that even though our meeting times are short they are still productive. For our project schedule, we are not expecting any delays as our project is software based and user testing can be conducted online. Since the work for the project can be done online, Covid-19 is not a significant factor. Some possible delays could be from the data handover process from SIAEC to us.

2. Project Task Allocation

Responsibility Matrix (Exploration and Implementation Phase for Prototype) R – Responsible S - Supporting								
	Task	Justin	Olivia	Kairan	Chester	Yu Ying	Devid	Wenqi
1	UI/UX design	S	R					
2	Database Schema	R		R	S		S	
3	Data Cleaning	R		R	R	R	R	R
4	Data Integration	R		R				
5	Operations Research				S	R	S	R
6	Risk Management	R			R			
7	Project Management	R			R			
8	Updating of schedule	S			R			
9	Updates to budget		R				R	
10	Report	S	S	S	S	R	R	S
11	Slides	S	R	S	S	S	S	S
12	Frontend	S	R					
13	Database	R		S				
14	Backend	S		R				
15	System architecture	S		R				

Table 10: Team Responsibility Table

3. Project Schedule

We have listed the tasks in a Gantt chart where each column represents the duration of a week.

<table>
<tr><th colspan="4">March</th><th colspan="4">April</th><th colspan="4">May</th></tr>
<tr><td colspan="4">Concept selection and application selection</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="4">Wireframe for UI</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="5">Research theory and application</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="8">Database Planning</td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Frontend exploration</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Backend exploration</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Database exploration</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="6">Formulation and completion of Asset Optimization</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td colspan="2">Evaluation of choices</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td colspan="3">Review 2 report</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="3">Review 2 slides</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td colspan="7">Formulation and completion of Just In Time Provisioning</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="4">Formulation and completion of Asset Management</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Frontend implementation</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Backend implementation</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Testing</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Price Monitoring System</td></tr>
</table>

<table>
<tr><th colspan="4">June</th><th colspan="4">July</th><th>August</th></tr>
<tr><td>Price Monitoring System</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">Frontend Implementation</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">Backend Implementation</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">Testing</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">Model fine tuning</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Review 3 slides</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="2">Deploy backend</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="2">Deploy frontend</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="2">User testing</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="3">Final Report</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Poster</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Summary of analysis result</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="3">Software Documentation</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Exhibition</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Final Presentation</td></tr>
</table>

4. Budget Allocation

	May	Jun	Jul	Aug	*Comments*
Available funds					
Total Fund (SGD)	*4,000*	*3,410*	*2,820*	*2,230*	*Fund in the beginning of the period*
Expenditure					
Tools					
Jira	52.50	52.50	52.50	52.50	*$7.5/pax/month*2 months*7pax*
Software					
RDS	200	200	200	200	*$0.276/hr*1440 hrs*
EC2	287.50	287.50	287.50	287.50	*$0.328/hr*1440 hrs*
CloudFront	0	0	0	10	*$10/usage*
SSL Cert	0	0	0	25	*$25/certificate*
Transportation					
Taxi	100	100	100	100	*$25/trip/car*2cars*4trips*
Total Expenditure (SGD)	*590*	*590*	*590*	*625*	
Net Fund (SGD)	**3,410**	**2,820**	**2,230**	**1,605**	*Fund at the end of the period*

Table 11: Budget Allocation

Conclusion

This report has explored many concepts in each sub-system (Operations Research & Analysis and Web Application Design) starting from the analysis of the problems and users' needs & constraints together with various literature reviews as the foundation to derive many creative ideas to pursue in the next stage of the project (Review 3) where the detailed design, testing and implementing will take place. Various concepts, techniques, tools, libraries and programming languages are carefully considered and weighed upon for the pros and cons before selecting each design concept based on various criteria.

Reference

Bakusevych, T. (2018, 07 17). *10 rules for better dashboard design | by Taras Bakusevych*. UX Planet. Retrieved April 21, 2022, from https://uxplanet.org/10-rules-for-better-dashboard-design-ef68189d734c

Bakusevych, T. (2021, March 2). *Top vs side navigation: Which one is better for your product?* UX Collective. Retrieved April 22, 2022, from https://uxdesign.cc/top-navigation-vs-side-navigation-wich-one-is-better-24aa5d835643

Bhatt, S. (2020, August 31). *Pros and Cons of Django Framework for App Development*. DZone. Retrieved April 24, 2022, from https://dzone.com/articles/pros-and-cons-of-django-framework-for-app-developm

Borrelli, P. (2021, October 26). *Angular vs. React vs. Vue.js: Comparing performance*. LogRocket Blog. Retrieved April 21, 2022, from https://blog.logrocket.com/angular-vs-react-vs-vue-js-comparing-performance/

Bruce, D. (2021, April 21). *Understanding the Pros and Cons of MongoDB*. KnowledgeNile. Retrieved April 13, 2022, from https://www.knowledgenile.com/blogs/pros-and-cons-of-mongodb/

Celikmih, K., Inan, O., & Uguz, H. (2020, 08 28). *Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method*. Hindawi. Retrieved April 24, 2022, from https://www.hindawi.com/journals/sp/2020/8616039/

Dhaduk, H., Kaneriya, T., & Shah, H. (2020, October 19). *Best Frontend Frameworks for Web Development in 2022*. Simform. Retrieved April 13, 2022, from https://www.simform.com/blog/best-frontend-frameworks/

Django Advantages and Disadvantages - Why You Should Choose Django? (n.d.). DataFlair. Retrieved April 24, 2022, from https://data-flair.training/blogs/django-advantages-and-disadvantages

Lim, E (2022). Personal communication [Personal interview]

Flom, P. (2018, March 13). *The Advantages of Using an Independent Group T-Test*. Sciencing. Retrieved April 20, 2022, from https://sciencing.com/advantages-using-independent-group-ttest-8647277.html

Gnedoy, P. (n.d.). *Node.js vs Go: Which is Better for Backend Development?* Uptech. Retrieved April 13, 2022, from https://www.uptech.team/blog/nodejs-vs-go

Guriev, B. (2021, March 26). *Firebase Pros and Cons: When You Should and Shouldn't Use Firebase*. OSDB. Retrieved April 13, 2022, from https://osdb.io/firebase-pros-and-cons-when-you-should-and-shouldnt-use-firebase-osdb/

Gutierrez, R.S., Solis, A.O., & Mukhopadhyay, S. (2007, 2 21). *Lumpy demand forecasting using neural networks.* International Journal of Production Economics. Retrieved 02 2, 2022, from https://www.sciencedirect.com/science/article/abs/pii/S0925527307000540

Ibe, O. C. (n.d.). *Random Walk*. https://www.sciencedirect.com/topics/mathematics/random-walks

International Air Transport Association. (2015). *Guidance Material and Best Practices for Inventory Management, 2nd Edition, 2015*. IATA. Retrieved April 24, 2022, from https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/inventory-mgmt-2nd-edition.pdf

Kaneriya, T., & Shah, H. (2022, February 28). *Advantages & Disadvantages of Node.js : Why to Use Node.js?* Simform. Retrieved April 13, 2022, from https://www.simform.com/blog/nodejs-advantages-disadvantages/

Kong, W. L., & Lai, P. C. (2007). *A virtual warehouse simulation tool for Aerospace Rotables Management.* ResearchGate. https://www.researchgate.net/publication/224699181_A_Virtual_Warehouse_Simulation_Tool_for_Aerospace_Rotables_Management

Lange, K. (2021, February 25). *What's Serverless? Pros, Cons & How Serverless Computing Works*. BMC Software. Retrieved April 22, 2022, from https://www.bmc.com/blogs/serverless-computing/

Let's Understand the Pros and Cons of Using Django. (2020, October 27). Benchmark IT Solutions. Retrieved April 13, 2022, from https://www.benchmarkit.solutions/lets-understand-the-pros-and-cons-of-using-django/

MacDonnell, M., & Clegg, B. (n.d.). *Management of rotable aircraft spares inventory: review of practice and development of new solutions. [004-0250]*. POMS. Retrieved April 24, 2022, from https://www.pomsmeetings.org/confpapers/004/004-0250.pdf

Maras. (2020, April 6). *Amazon RDS Pros and Cons - A detailed overview*. Saras Analytics. Retrieved April 13, 2022, from https://sarasanalytics.com/blog/amazon-rds-pros-and-cons

Miceli, V., & Chawla, G. (2019). *Demand Forecasting and Inventory Management for Spare Parts*. DSpace@MIT. Retrieved April 24, 2022, from https://dspace.mit.edu/handle/1721.1/121291

Ofiwe, M. (2021, October 28). *Metasearch Engines*. SEMrush. Retrieved April 21, 2022, from https://www.semrush.com/blog/metasearch-engine/

Ohri, A. (2022, 02 12). *10 Popular Regression Algorithms In Machine Learning Of 2021*. Jigsaw Academy. Retrieved April 24, 2022, from https://www.jigsawacademy.com/popular-regression-algorithms-ml/

Ostertagova, E., & Ostertag, O. (2012, December). *Forecasting Using Simple Exponential Smoothing Method*. ResearchGate. https://www.researchgate.net/publication/256086712_Forecasting_Using_Simple_Exponential_Smoothing_Method

Pedrosa, G., & Conceição, S.V. (1970, 1 1). *comparison of a new bootstrapping method with parametric approach considering stochastic lead time: Semantic scholar.* COMPARISON OF A NEW BOOTSTRAPPING METHOD WITH PARAMETRIC APPROACH CONSIDERING STOCHASTIC LEAD TIME | Semantic Scholar. Retrieved 04 2, 2022, from https://www.semanticscholar.org/paper/COMPARISON-OF-A-NEW-BOOTSTRAPPING-METHOD-WITH-LEAD-Pedrosa-Concei%C3%A7%C3%A3o/15d1f7ce93a98165ba4a984f3a872c8838ce04e9

Pornaras, G. (2021, October 7). *Selenium vs. Beautiful Soup: A Full Comparison*. BlazeMeter. Retrieved April 24, 2022, from https://www.blazemeter.com/blog/selenium-vs-beautiful-soup-a-full-comparison

Pricing intelligence: what is it and why matters? (n.d.). Competitor Monitor. Retrieved April 10, 2022, from https://www.competitormonitor.com/blog/pricing-intelligence-what-is-it-and-why-does-it-matter/

Prince, C., Turrini, L., & Meissner, J. (2021, 07 3). *Intermittent demand forecasting for spare parts: A critical review*. Omega. Retrieved 03, 27, from https://www.sciencedirect.com/science/article/pii/S0305048321001225

Rak, V. (2021, February 5). *Pros & Cons of Ruby on Rails You Should Know Before Choosing the Technology for Your Startup*. Sloboda Studio. Retrieved April 13, 2022, from https://sloboda-studio.com/blog/pros-and-cons-of-ruby-on-rails/

Rathod, A. (2021, January 15). *10 Potent Golang Frontend Frameworks in 2021*. CMarix. Retrieved April 13, 2022, from https://www.cmarix.com/blog/top-golang-web-frameworks-for-developers-in-2021/

Rožanec, J.M., & Mladenic, .. (2021, March 23). *[PDF] Reframing demand forecasting: a two-fold approach for lumpy* ... Semantic Scholar. Retrieved April 6, 2022, from https://www.semanticscholar.org/paper/Reframing-demand-forecasting%3A-a-two-fold-approach-Ro%C5%BEanec-Mladenic/c7aff9f711c7a65ee5fa74116310f6ea46c750c7

Sahin, M., Eldemir, F., & Turkyilmaz, A. (2021, 12 8). Retrieved 03 27, 2022, from https://www.sciencedirect.com/science/article/pii/S2352146521008553

Schae, J. (2020). *A RealWorld Comparison of Front-End Frameworks 2020 | by Jacek Schae | DailyJS*. Medium. Retrieved April 13, 2022, from https://medium.com/dailyjs/a-realworld-comparison-of-front-end-frameworks-2020-4e50655fe4c1

Schmitt, K. R. (n.d.). *What are the main advantages and disadvantages of using a Simple Moving Average (SMA)?* Investopedia. Retrieved April 20, 2022, from https://www.investopedia.com/ask/answers/013015/what-are-main-advantages-and-disadvantages-using-simple-moving-average-sma.asp

SIAEC. (n.d.). *SIA Engineering Company*. SIA Engineering Company. Retrieved April 21, 2022, from https://www.siaec.com.sg/company_profile.html

SIA Engineering Company. (n.d.). SIA Engineering Company. Retrieved April 10, 2022, from https://www.siaec.com.sg/company_profile.html

Singh, M. (n.d.). *Firebase vs. AWS vs. MongoDB*. Morioh. Retrieved April 24, 2022, from https://morioh.com/p/fd327efa3b32

Syntetos, Boylan, & Croston. (2017, 12 21). *On the categorization of demand patterns: Journal of the Operational Research Society: Vol 56, No 5*. Taylor & Francis Online. Retrieved April 24, 2022, from https://www.tandfonline.com/doi/abs/10.1057/palgrave.jors.2601841

Syntetos, A.A., Babai, M.Z., & Altay, N. (2011). *Distributional Assumptions for Parametric Forecasting of Intermittent Demand.* Altay, N., Litteral, L. (eds) Service Parts Management. Springer, London. Retrieved 04, 2022, from https://www.google.com/url?q=https://link.springer.com/chapter/10.1007/978-0-85729-039-7_2%23citeas&sa=D&source=docs&ust=1650796582207425&usg=AOvVaw1wOdOT-_gN_7nbLLlggCNW

Syntetos, A.A., & Boylan, J.E. (2001). *On the bias of intermittent demand estimates*. Journal of Production Economics, 71(1-3), 457–466. Retrieved 04, 2022, from https://doi.org/10.1016/s0925-5273(00)00143-2

Tang @SeeKitCNA, S.K. (2021, October 15). *Worst may be over for Singapore's aerospace industry as businesses start hiring again*. CNA. Retrieved April 24, 2022, from https://www.channelnewsasia.com/business/aerospace-industry-singapore-recovery-businesses-hiring-jobs-2241216

Turrini, L., & Meissner, J. (2017, 09 29). *Spare parts inventory management: New evidence from distribution fitting*. European Journal of Operational Research. Retrieved 03 27, 2022, from https://www.sciencedirect.com/science/article/pii/S037722171730886X

Van Westen, C. (n.d.). *5.5 Risk assessment methods*. Caribbean Disaster Emergency Management Agency - CDEMA. Retrieved April 20, 2022, from https://www.cdema.org/virtuallibrary/index.php/charim-hbook/methodology/5-risk-assessment/5-5-risk-assessment-methods

Volodymyr, T. (2017, June 8). *Express.js Mobile App Development: pros and cons of Node.js framework*. Apiko. Retrieved April 24, 2022, from https://apiko.com/blog/express-mobile-app-development/

Wattanarat, V., Phimphavong, P., & Matsumaru, M. (2011, January 25). *Demand and Price Forecasting Models for Strategic and Planning Decisions in a Supply Chain*. https://www.u-tokai.ac.jp/uploads/sites/12/2021/03/No.2_PP37-42.pdf

Why use serverless computing? | Pros and cons of serverless. (n.d.). Cloudflare. Retrieved April 22, 2022, from https://www.cloudflare.com/en-gb/learning/serverless/why-use-serverless/

Wilkinson, G. (2020, November 11). *Asset management optimization for repairable spare parts – AnyLogic Simulation Software*. AnyLogic. Retrieved April 24, 2022, from https://www.anylogic.com/blog/asset-management-optimization-for-repairable-spare-parts/

Willemain, T.R., & Smart, C.N. (n.d.). *A new approach to forecasting intermittent demand for service parts inventories*. International Journal of Forecasting. Retrieved 3 31, 2022, from https://www.sciencedirect.com/science/article/abs/pii/S016920700300013X

Wodehouse, C. (2018, December 4). *Firebase vs. AWS vs. MongoDB: 3 Technologies Behind Modern Software Stacks*. Business 2 Community. Retrieved April 24, 2022, from https://www.business2community.com/brandviews/upwork/firebase-vs-aws-vs-mongodb-3-technologies-behind-modern-software-stacks-02147649

Wu, C., & Buyya, R. (n.d.). *Real Option Theory and Monte Carlo Simulation*. https://www.sciencedirect.com/topics/computer-science/geometric-brownian-motion

Annex A

1. Detailed Problem Analysis

Figure 0 below summarizes the methods that SIAEC is currently taking to manage inventory. SIAEC's goal is to maintain a high customer service level while minimizing inventory cost.

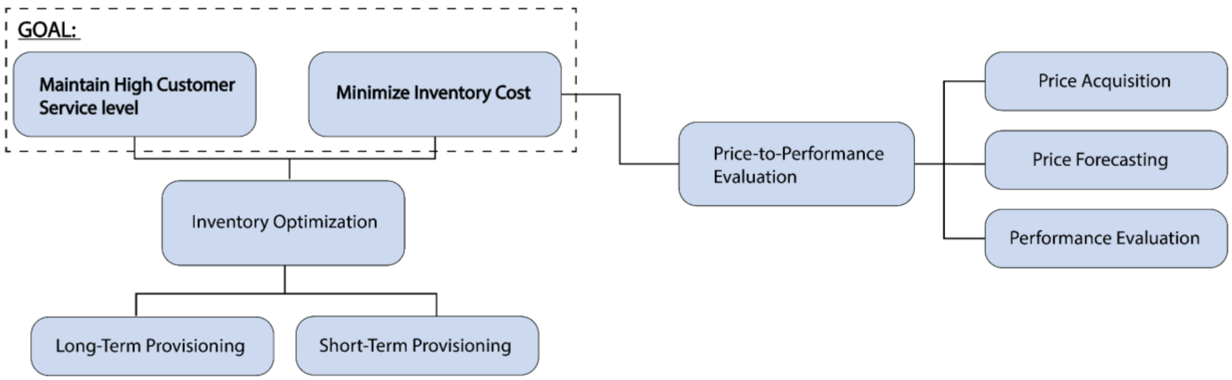


Figure 0: Summary of SIAEC's methods for managing inventory

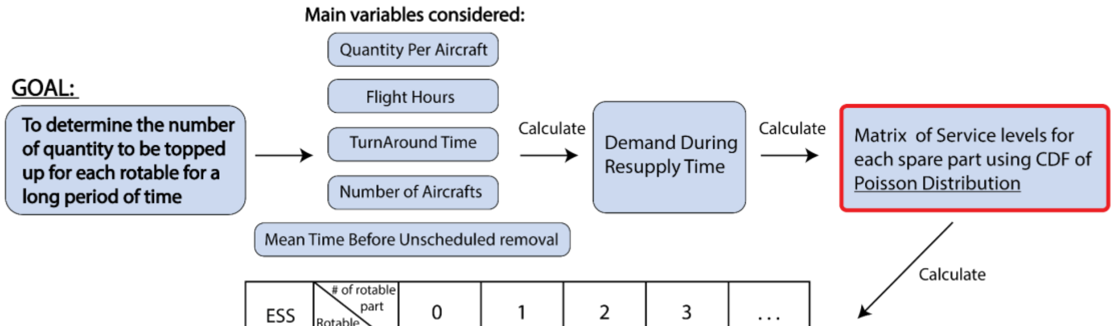
In the aviation industry, managing rotables is an essential operation that helps to support equipment availability for aircrafts, allowing for smooth continuity of airline operations. Due to the highly intermittent[44] demand of rotables, it is inherently more uncertain to manage rotables demand as compared to other traditional fast-moving products. As rotables are critical to the continuity of airline operations, SIAEC tends to overstock inventories to mitigate the risk of irregular spare part demand patterns to maintain high customer service level. As a result, unnecessary costs have been spent on maintaining and purchasing excess rotables to meet the required customer service level. To tackle this issue, we aim to improve the demand forecast accuracy and meet the required customer service level while at the same time, minimize inventory cost. Besides having a good demand forecasting algorithm to predict the optimal number of each rotable to hold, it is also important to have a reliable price monitoring and accurate evaluation of the performances of different vendors that sells rotable, to have a gauge on which vendors provides most cost-effective rotables, allowing SIAEC to make better purchasing decisions, further minimizing inventory costs.

SIAEC currently performs two types of provisioning of the rotables for inventory management, which includes short-term provisioning monthly and long-term provisioning every semiannually. The purpose of long-term provisioning is to determine the inventory quantity to be topped up to meet customer service level at a low inventory cost within the 6 month period. Short-term provisioning is to determine the number of rotables to be kept active out of the total current on-hand inventory that SIAEC has.

Figure 1 below shows the current process flow of the steps involved when SIAEC conducts their long-term provisioning of aircraft components in Microsoft Excel.

[44] Intermittent refers to demand drawing from a non-smooth pattern with mostly zero values and random non-zero demands

Long-Term Provisioning by SIAEC:



ESS	Rotable Part # \ # of rotable part	0	1	2	3	...
1	A	80.1%	90.1%	99.3%	100.0%	
1	B	88.1%	94.1%	99.1%	100.0%	
3	C	85.1%	91.1%	95.1%	96.1%	
3	D	90.1%	98.2%	100.0%	100.0%	
3	E	80.1%	90.1%	99.9%	100.0%	
2	F	80.5%	85.6%	95.1%	100.0%	
1	G	80.3%	88.8%	94.1%	99.8%	
	...	...	...	...	...	...

Figure 1: Overview of long-term provisioning in SIAEC

From the top left hand corner of figure 1, to maintain customer service level and minimize total inventory cost, SIAEC takes into account a few important variables, which are mainly quantity per aircraft (QPA), flight Hours (FH), turnaround time (TAT), number of aircrafts (NOA) and mean time before unscheduled removal (MTBUR). Using these variables, the demand during resupply time[45], is then calculated. As SIAEC currently assumes all rotables to fail according to Poisson distribution, the cumulative distribution function (CDF) of Poisson distribution is then used to calculate the service levels for each spare part given that a certain number of spare part (columns of the table) is stored as spares in the warehouse. By taking demand during resupply time as the rate parameter for the Poisson distribution, a matrix of service levels is then calculated as shown in the red outlined table in figure 1, where the rows and columns correspond to the rotable part number, and number of rotable quantity to be stored in the warehouse respectively.

The table in figure 1 shows a snapshot of how SIAEC determines the "optimal" stock for each spare part. Each rotable part is labeled with an importance level (i.e., ESS level[46]), either ESS1, ESS2 or ESS3, having to meet a targeted service level of 98%, 95% and 88% respectively. Taking into account the ESS level, SIAEC then ensures that each rotable part meets at least the targeted service level as stated by OEM. For example, with reference to row 1 of the table in figure 1, rotable part A is of ESS 1, and the CDF of the Poisson distribution indicates that there needs to be at least 2 rotable part A in the warehouse in order to ensure a service level of 99.3% (at least 98%). Hence, the "optimal" quantity required to be in the warehouse would be 2. The optimal **stocking** quantity would simply be a subtraction of the on-hand inventory from the required

[45] The average number of demands expected to arise during the turnaround time (repair+logistics time) for a rotable part

[46] ESS: Essentiality Code; ESS1 is the most important, followed by ESS2 and ESS3

optimal quantity. In the event where the on-hand inventory is greater than the required optimal quantity, the optimal stocking quantity would be zero.

Figure 2 below shows the current process flow of the steps involved when SIAEC conducts their long-term provisioning of aircraft components in Microsoft Excel.

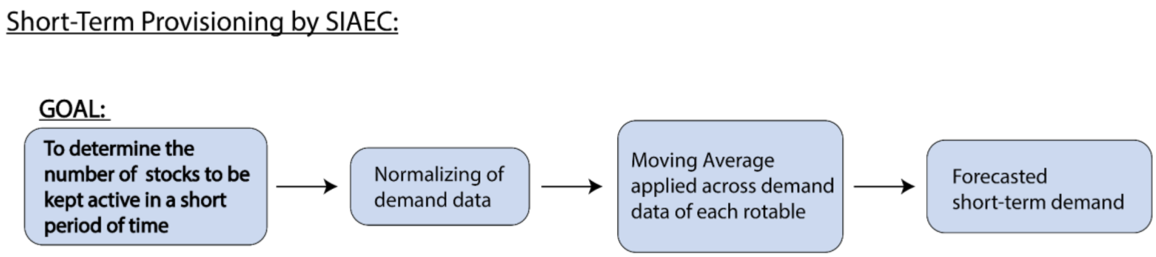


Figure 2: Overview of short-term provisioning in SIAEC (bigger picture shown in annex)

Apart from having long-term provisioning (6 months period), SIAEC also conducts a short-term provisioning (monthly): quantity of rotables required to be kept active. The importance of having a short-term provisioning will be explained in detail under "Short-Term Provisioning (Monthly)" in section 1.2. The current process aims to forecast future demand (short-term) in order to determine the active rotables. The forecasted method used in SIAEC is to normalize demand data[47] and applying moving averages across each rotable to be kept active in a short period of time.

The price-to-performance evaluation methods that SIAEC are currently using are kept rather elementary and thus may potentially incur unnecessary costs in the purchase of rotables.

2. Insights & Approaches

Insight A: Currently, SIAEC ensures that each rotable meets the required service level. In fact, SIAEC only requires rotables in the **same class** to be on **average** the required service level. We can see that the method that SIAEC currently uses results in the overprovisioning[48] of rotables, and is not cost-saving as it does not take into consideration the relative cost of the rotables.

Approach A: It makes sense from an operational perspective to manage low stock levels of very expensive rotables, while providing higher levels of safety stock for cheaper rotables.To tackle this issue, we have formulated an optimization model, which takes into consideration the cost and availability of rotables.

Insight B: MTBUR, is a very general estimation of the time a spare part can last on an aircraft and hence may not be accurate.

Approach B: If we are able to accurately predict MTBUR, we will be able to know when a demand will likely occur and hence provide supply for it. Some of the existing tools that we have researched include the use of machine learning and simulation.

---

[47] This is to ensure the data is kept aggregated for ease of calculations

[48] There exists overprovision due to the current method that SIAEC is provisioning. In the event that SIAEC requires a group of rotables to be at 95% service level, they would ensure that each part number achieves at least 95% and not the average. This causes the average ESS level to reach significantly higher than 95%, potentially incurring additional costs for provision

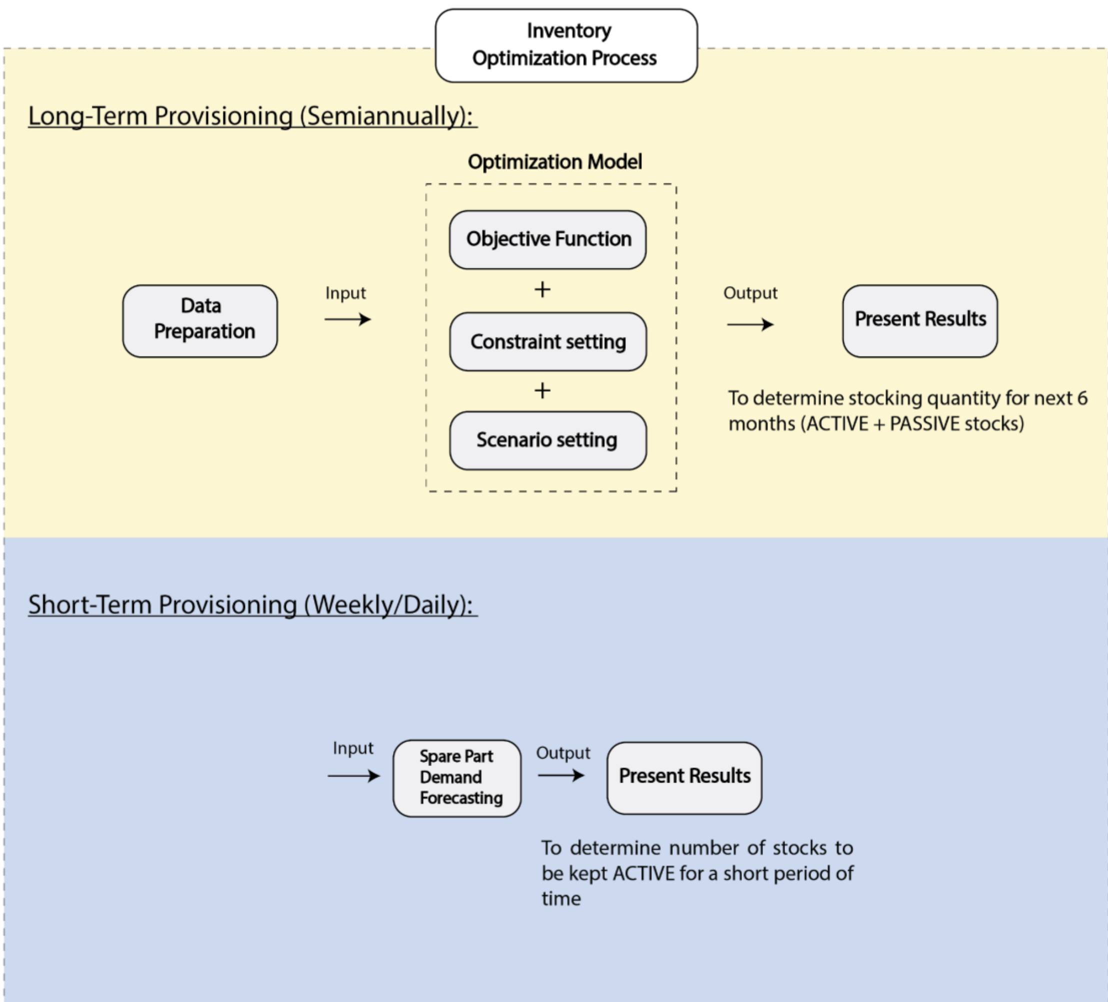
Insight C: SIAEC currently assumes that each spare part fails according to Poisson distribution.

Approach C: While it is appropriate to assume Poisson distribution for rotables failure times, we would explore more about the other demand distributions that could potentially perform better than Poisson.

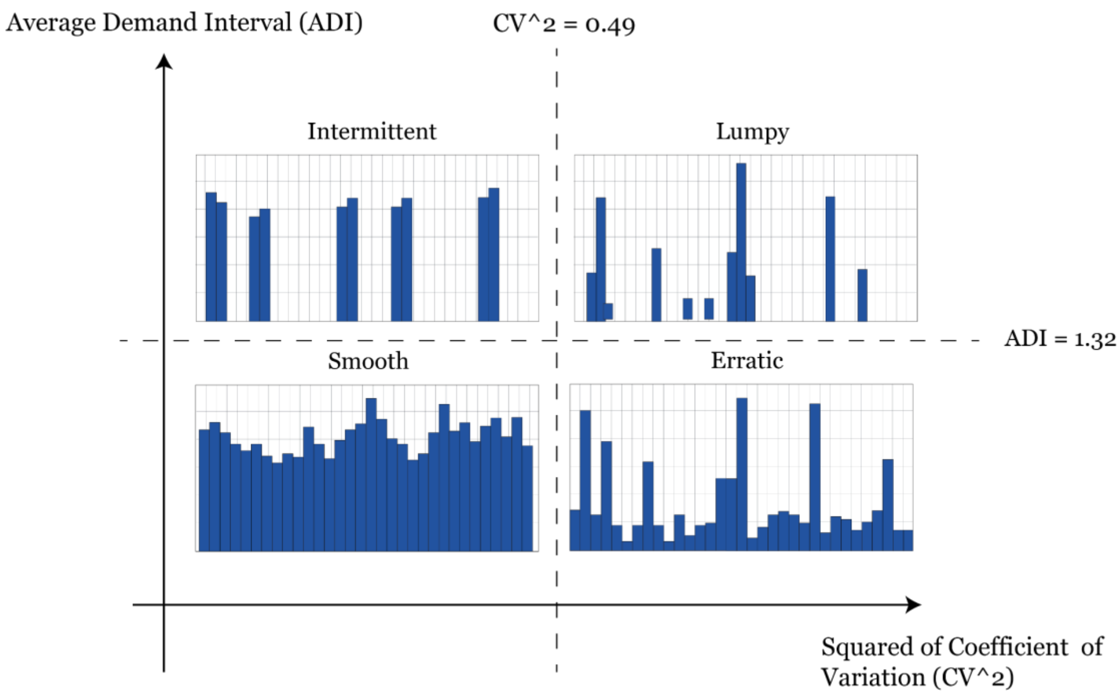
Insights D: SIAEC currently uses moving averages for short-term forecasting. However, findings from our literature reviews have proven that the use of classical forecasting methods such as moving average and exponential smoothing are not suitable for rotables because of their highly intermittent demand pattern (Turrini, & Meissner, 2017).

Approach D: We have further explored other forecasting techniques that are more suited for spare parts in general, which could help to better improve the short-term forecasting method.

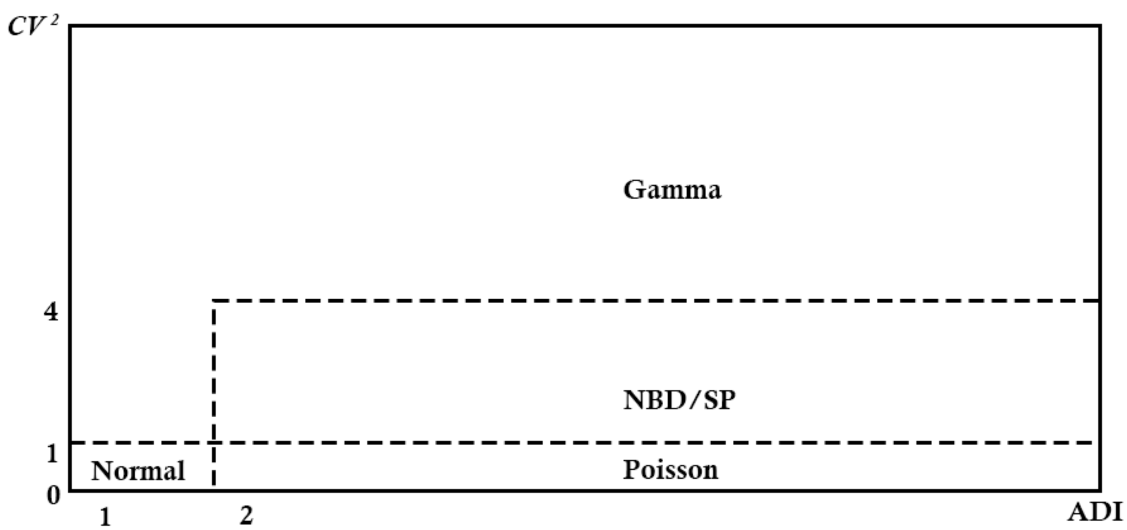
3. Overview of Our proposed Inventory Optimization Process



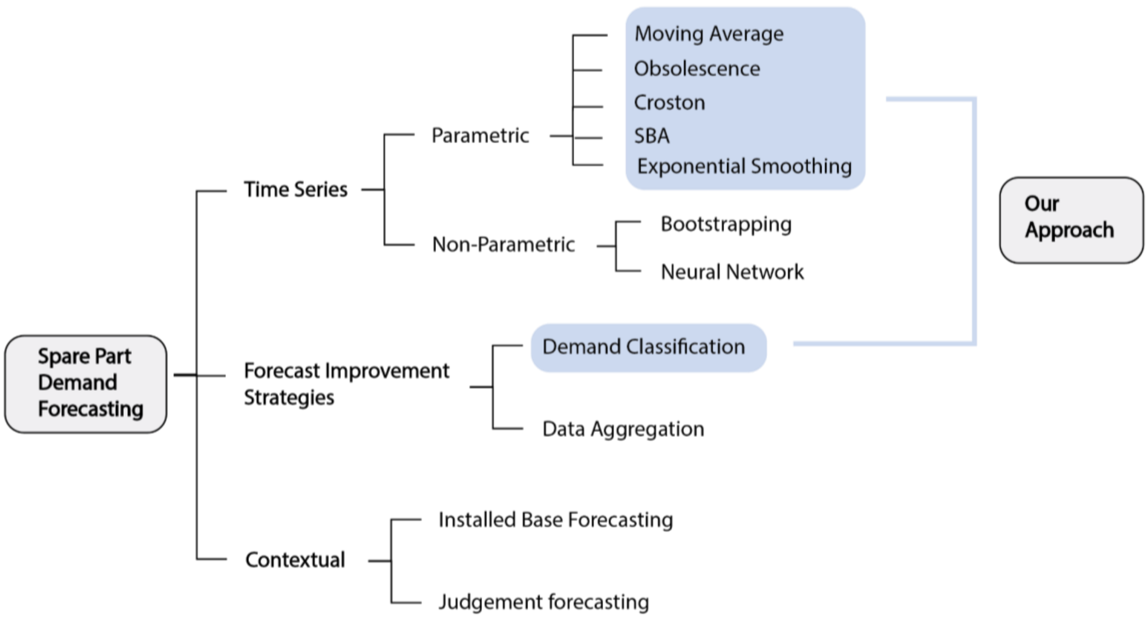
4. Rotable Classification into 4 Main Groups



5. Demand Distribution Assumptions



6. Spare Part Demand Forecasting Methods



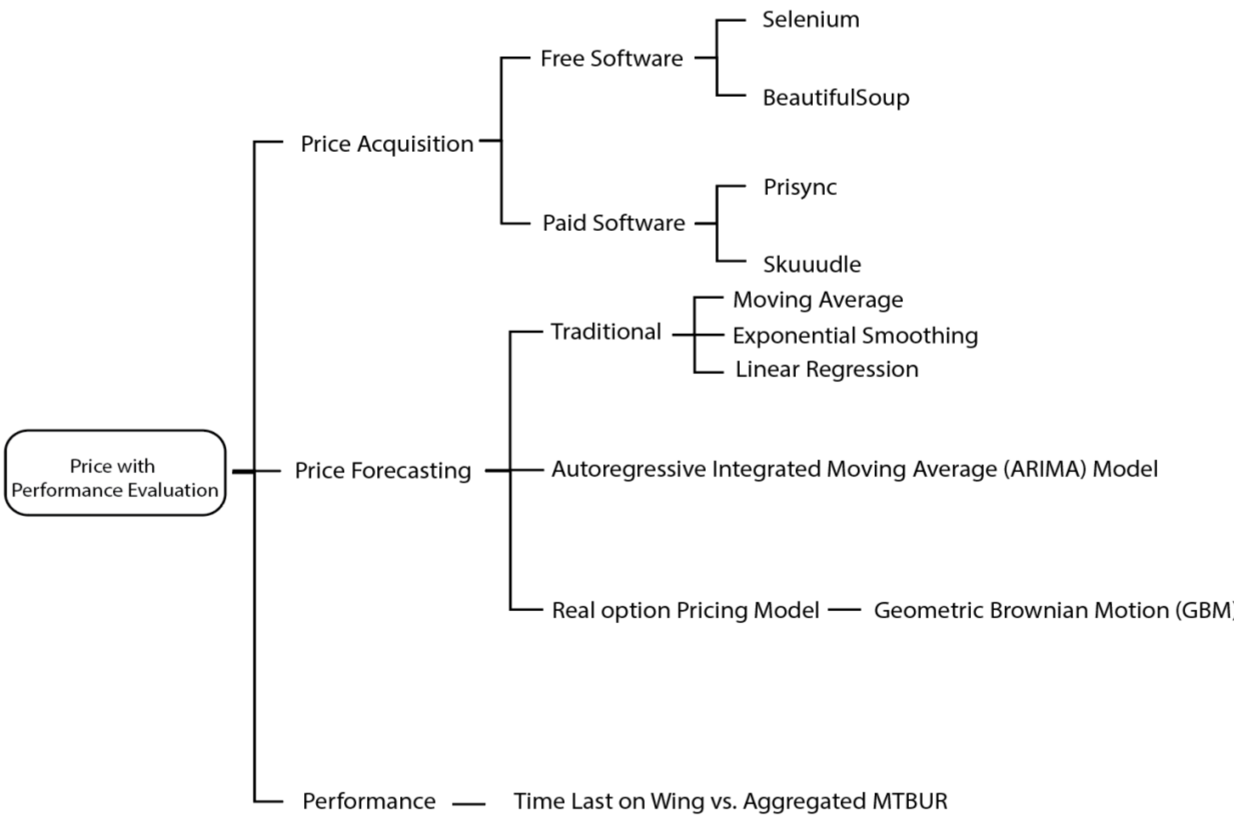
7. Short-Term Provisioning

Design Criteria*	Alternative Design Concepts	
	Parametric	Non-parametric
Data Requirement	+	0
Accuracy	0	+
Time	+	0
Ease of implementation	+	0
Totals	+++	+

Criteria for Short-Term Provisioning

Criteria\Score	-	0	+
Data Requirement	The method requires a few years of historical data	The method requires a few months a historical data	The method requires a month worth of historical data
Accuracy	The method produces poor accuracy result	The method produces acceptable accuracy result	The method produces exceptional accuracy result
Time	The method requires a lot of time to implement	The method uses moderate amount of time to implement	The method requires very little time to implement
Ease of Implementation	The method is computationally intensive with many mathematical equations and concepts	The method is moderately easy to implement with fair amount of mathematical equations and concepts	The method is easy to implement with little mathematical equations and concepts

8. Price to Performance Evaluation Breakdown



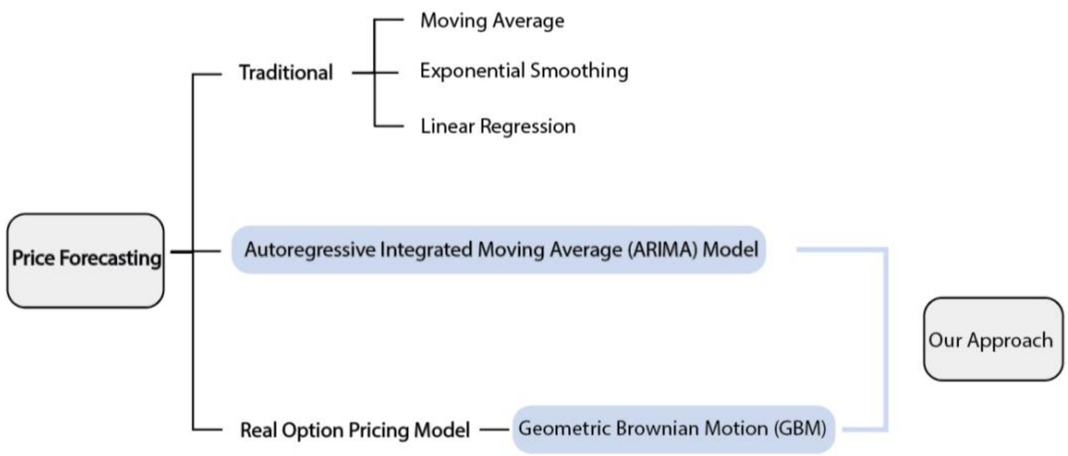
9. Price Monitoring System

	Alternative Design Concepts			
Design Criteria	**Skuuudle**	**Prisync**	**BeautifulSoup**	**Selenium**
Affordability	-	0	+	+
Ease of Integration	-	0	+	+
Adaptability	+	+	-	0
Total	-	+	+	++

Criteria for Price Monitoring System

Criteria\Score	-	0	+
Affordability	Software costs more than SGD 100	Software costs less than SGD 100	This software is free to use
Ease of Integration	Software is difficult to integrate relative to the other alternatives	Software is reasonably difficult to integrate relative to the other alternatives	Software is easy to integrate relative to the other alternatives
Adaptability	Requires major tuning whenever website is changed relative to the other alternatives	Requires moderate tuning whenever website is changed relative to the other alternatives	Requires little to no tuning whenever website is changed relative to the other alternatives

10. Price Forecasting Breakdown



11. Price Monitoring System

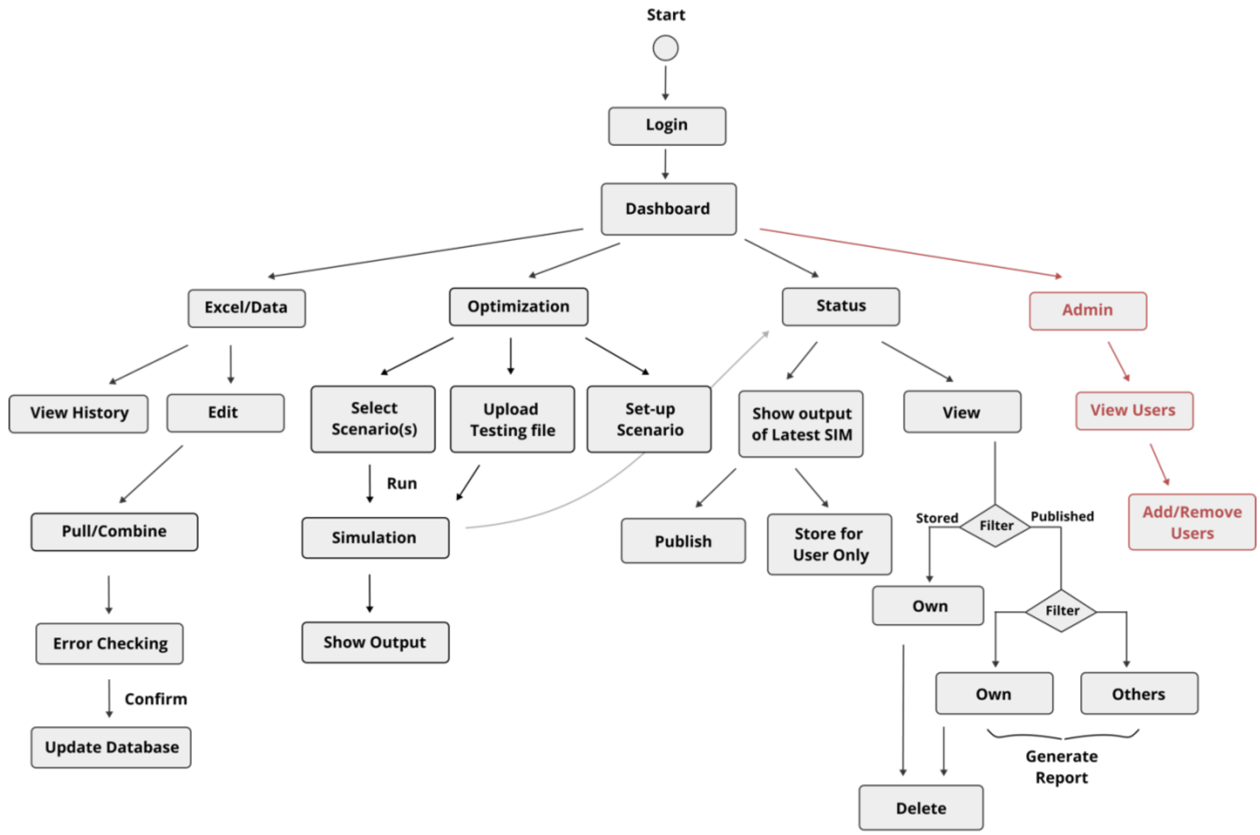
Design Criteria*	Alternative Design Concepts	
	Geometric Brownian Motion	**ARIMA**
Data Requirement	+	+
Accuracy	+	0
Time	+	+
Ease of Implementation	0	0
Totals	+++	++

** Further explanation of the Criteria used and assessment rubrics can be find in Annex x.x*

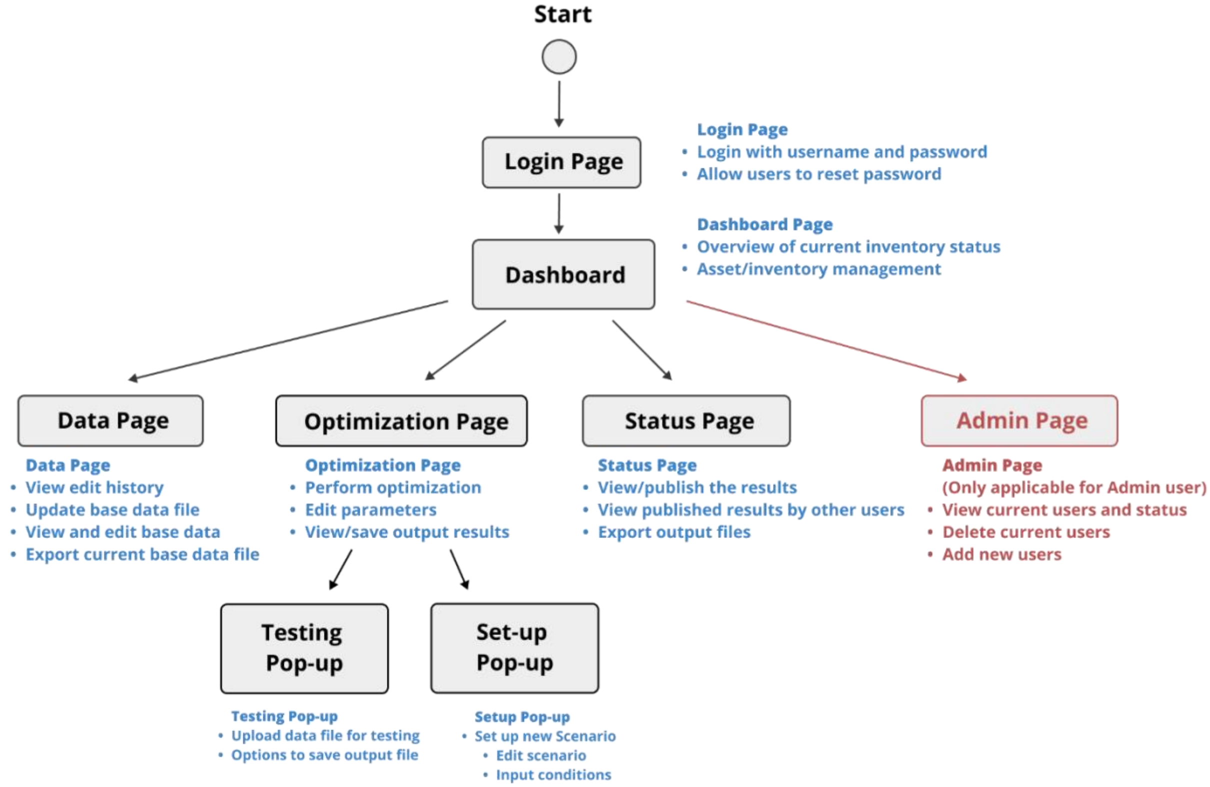
Criteria for Price Forecasting

Criteria\Score	-	0	+
Data Requirement	The method requires a few years of historical data	The method requires a few months a historical data	The method requires a month worth of historical data
Accuracy	The method produces poor accuracy result	The method produces acceptable accuracy result	The method produces exceptional accuracy result
Time	The method requires a lot of time to implement	The method uses moderate amount of time to implement	The method requires very little time to implement
Ease of Implementation	The method is computationally intensive to implement with many mathematical equations and concepts	The method is moderately easy to implement with fair amount of mathematical equations and concepts	The method is easy to implement with little mathematical equations and concepts

12. User Action-Flow Diagram



13. Page-Flow Diagram



14. Frontend

Design Criteria	Alternative Design Concepts		
	React	Angular	Vue
Ease of Learning	+	-	-
Performance	+	-	+
Applicability of Features	+	+	0
Total	+++	-	0

Criteria for Frontend

Criteria\Score	-	0	+
Ease of Learning	All team members have no prior knowledge with the framework	Some team members have some prior knowledge with the framework	Every members have prior experience with the framework
Performance (Schae, 2020)	Has slow rendering speed, slower response speed with increased number of features relative to the other alternatives	Has a moderate rendering speed, no significant reduction in response speed with increased number of features relative to the other alternatives	Has fast rendering speed, no significant reduction in response speed with increased number of features relative to the other alternatives
Applicability of Features	Offers few relevant features fulfilling the needs and constraint of the project relative to the other alternatives	Offers some relevant features fulfilling the needs and constraint of the project relative to the other alternatives	Offers many relevant and useful features fulfilling the needs and constraint of the project relative to the other alternatives

15. Backend

Design Criteria	Alternative Design Concepts		
	Golang Gin	Express.js	Django
Ease of Learning	-	+	0
Ease of Implementation	0	-	+
Performance	0	-	+
Total	-	-	++

Criteria for Backend

Criteria\Score	-	0	+
Ease of Learning	All team members have no experience with the framework, few learning resources available	All team members have no experience with the framework, some learning resources available	Some team members have experience with the framework, some learning resources available
Ease of Implementation	Framework requires a lot of code, rigid in implementation	Framework requires a moderate amount of code, relatively flexible in implementation	Framework requires a moderate amount of code, flexible in implementation
Performance	Slow response time relative to the other alternatives	Moderate response time relative to the other alternatives	Fast response time relative to the other alternatives

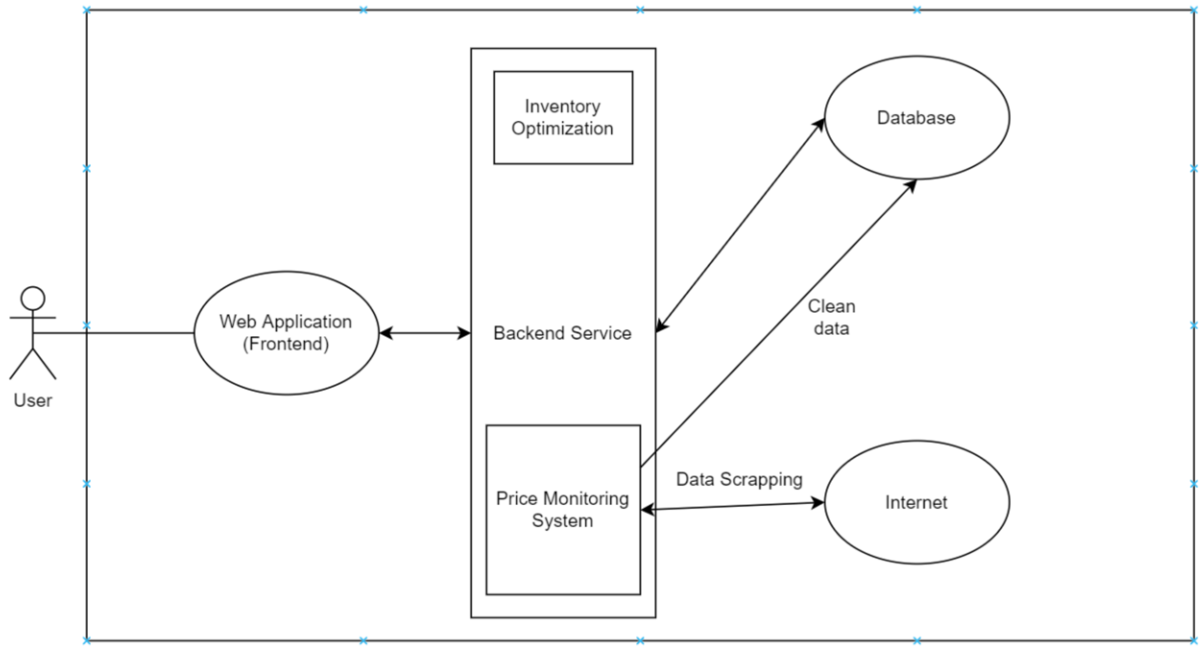
16. Database

Design Criteria	Alternative Design Concepts		
	Firebase	AWS RDS	MongoDB Atlas
Performance	-	+	+
Applicability of Features	-	+	0
Cost-Efficiency	-	0	0
Total	- - -	++	+

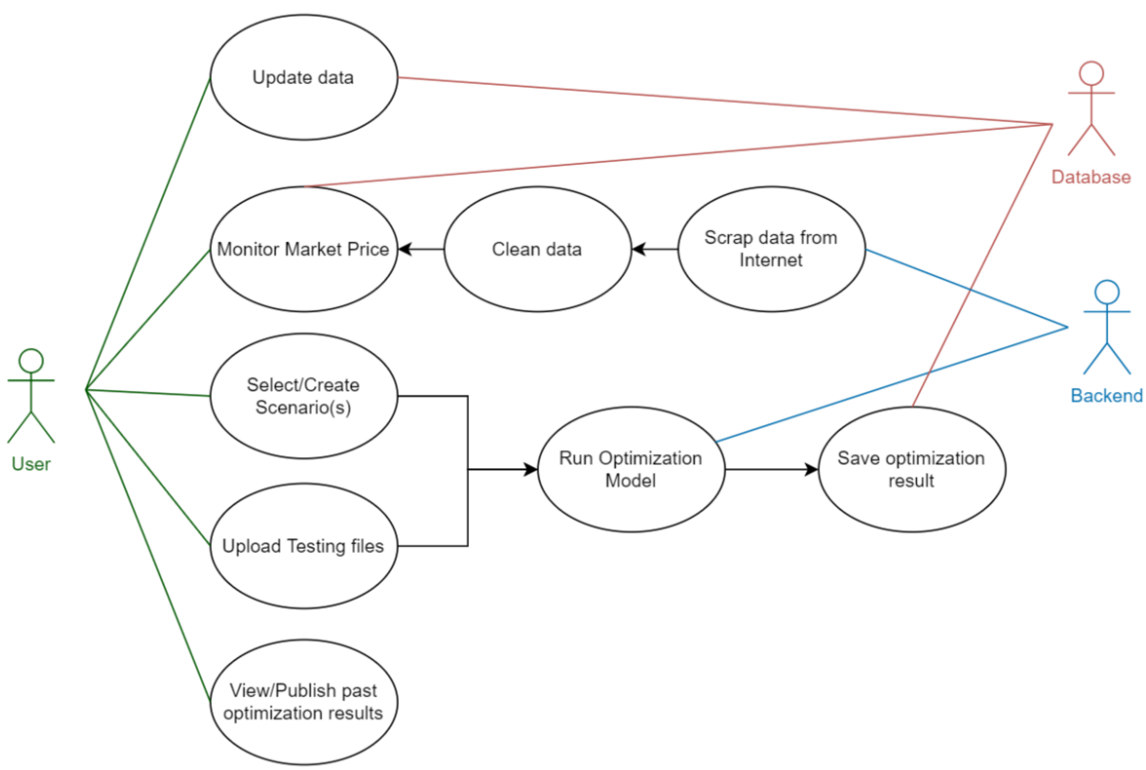
Criteria for Database

Criteria\Score	-	0	+
Performance	Slow reads and writes with poor concurrency capabilities relative to the other alternatives	Fast reads and writes with moderate concurrency capabilities relative to the other alternatives	Fast reads and writes with good concurrency capabilities relative to the other alternatives
Applicability of Features	Has few relevant features offered by service provider relative to the other alternatives	Has some relevant features offered by service provider relative to the other alternatives	Has many relevant and useful features offered by service provider relative to the other alternatives
Cost-Efficiency	Monthly expenditure is above SGD 300	Monthly expenditure is between SGD 300 and SGD 100	Monthly expenditure is below SGD 100

17. System Overview of Interaction between Different Systems



18. System Overview of Interaction between Users and Systems



19. Total Price Forecast & Absolute Deviation of Demand (ARIMA vs GBM)

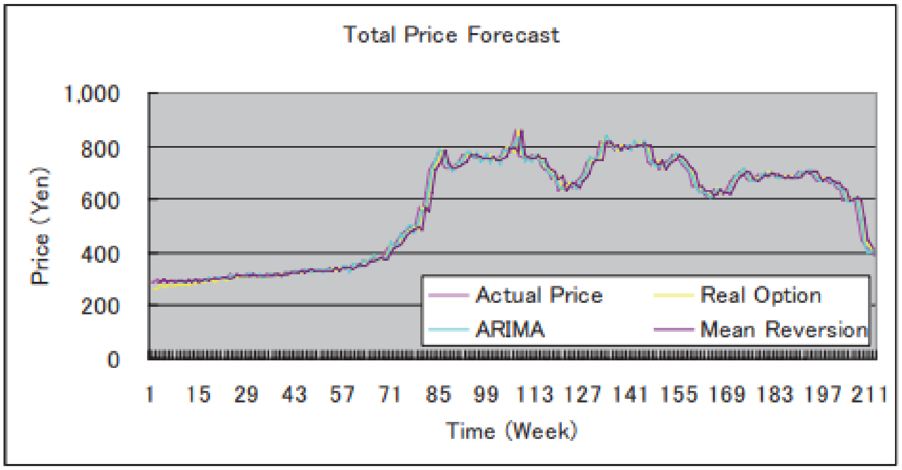


Table 4 Absolute deviation of demand

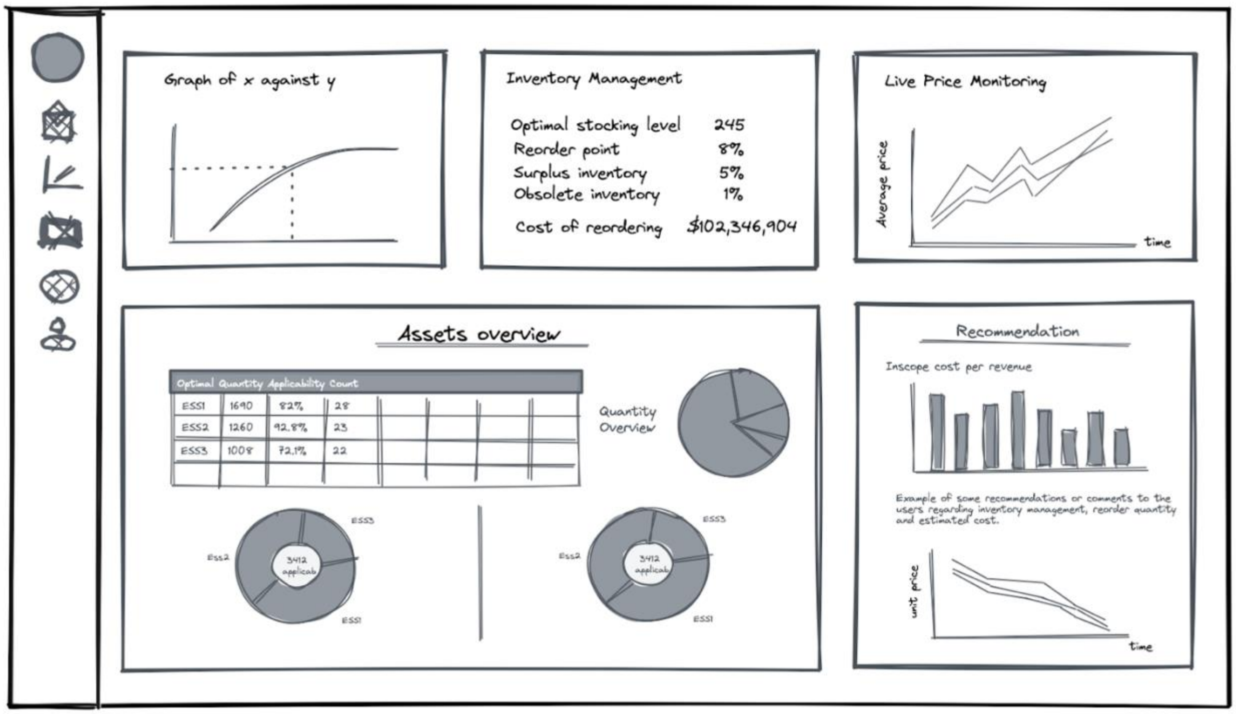
Real Option% AD	ARIMA Model%AD
0.182	0.168

Table 5 Absolute deviation of price

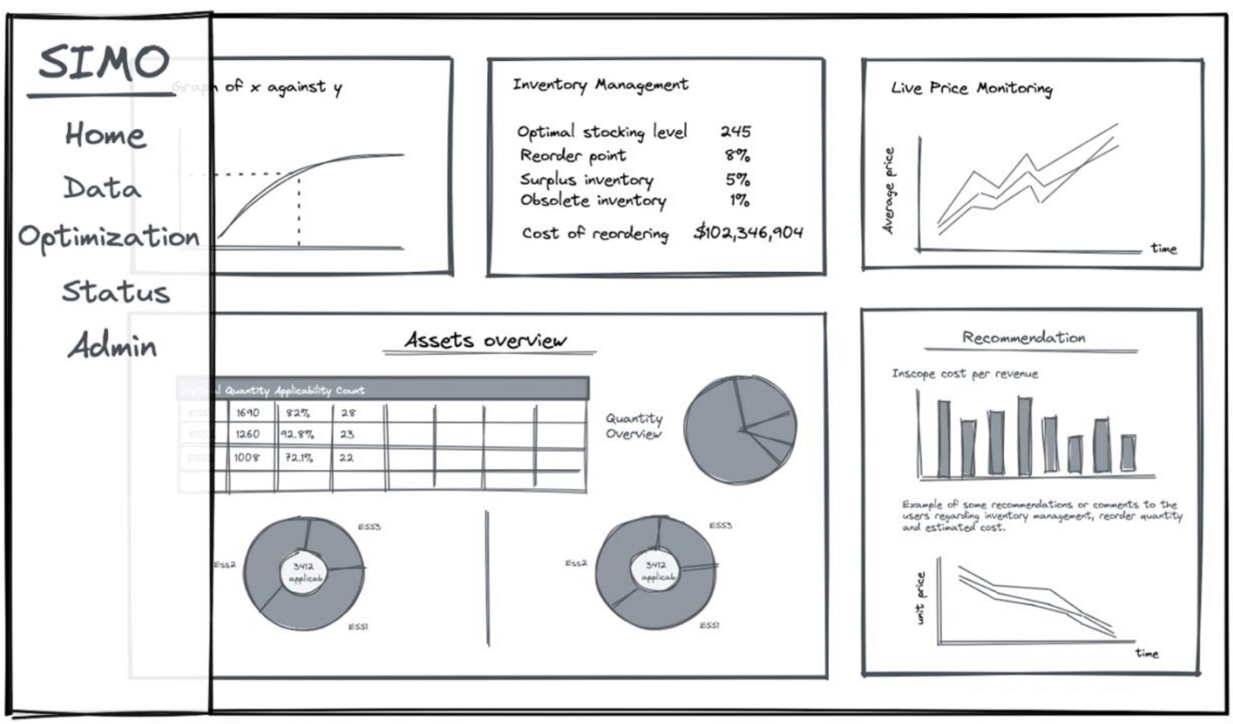
Real Option% AD	ARIMA Model%AD
0.06	0.6

20. Improved Dashboard Design

Collapsed



Full View

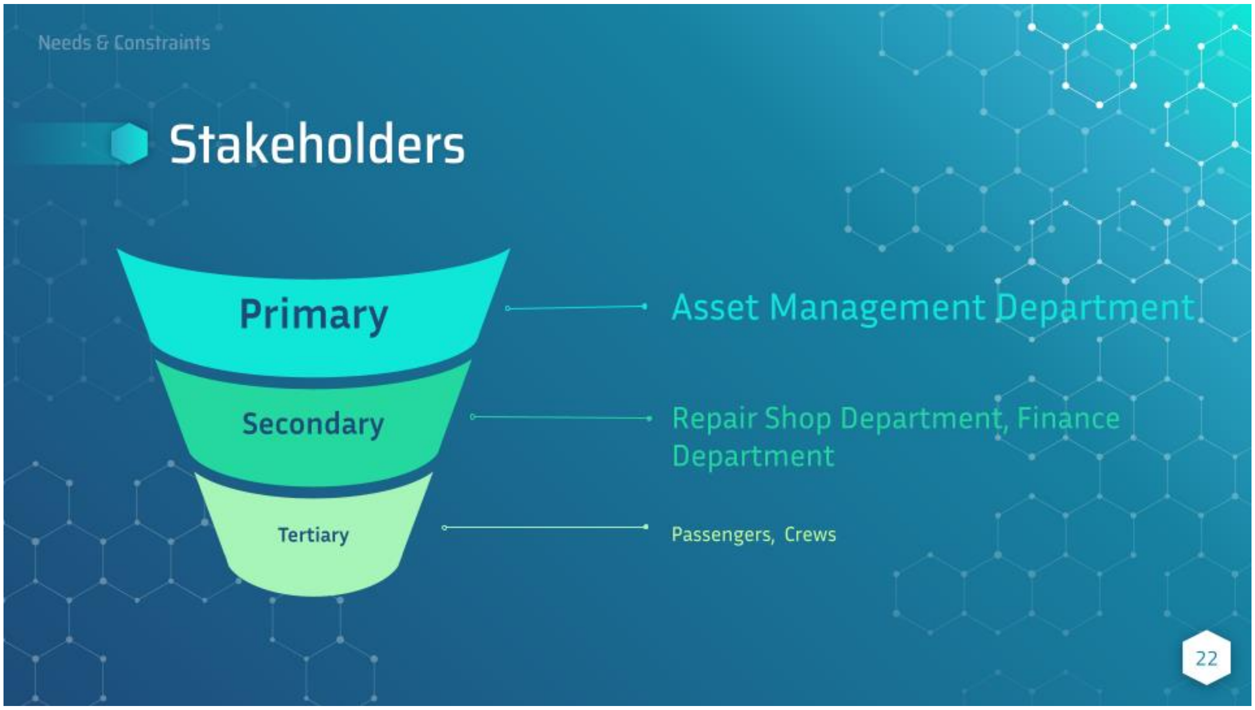


21. Risk Assessment

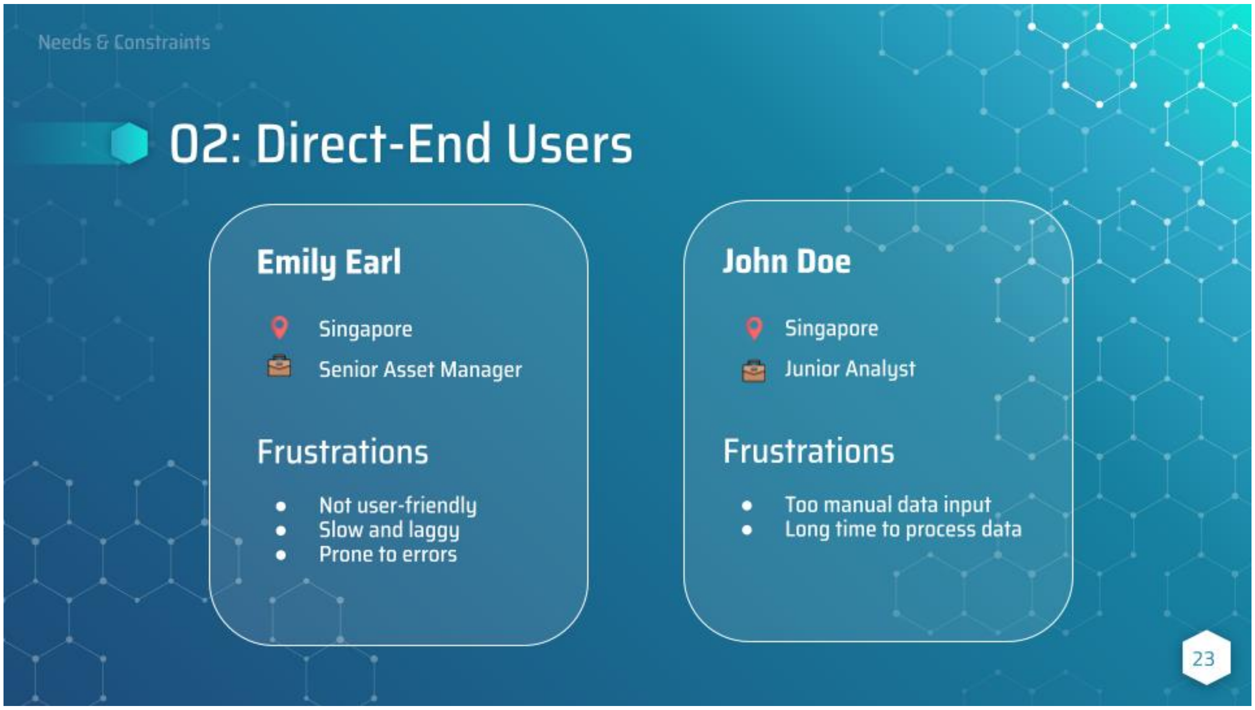
Event	Risk	Description	Consequences	Mitigations
Server Failure	Low	Caused by disk failure, virus attack, failed updates and physical disasters Low risk of occurring as we will be using cloud service. Only physical disasters or virus attacks can result in a server failure	Web application would be down.	Writing test cases for algorithms
Inaccurate Data Entry	Moderate	Data input into the database is wrong	Inaccurate data displayed Decisions made based on incorrect data	Input validation
Security Breach	Moderate	Caused by human error and loopholes in software. Low risk of occurrence as we will follow the latest established security practices	Stolen data Web application down	Follow the latest established security practices
Data Loss	Moderate	Caused by human error, viruses, natural disasters, power failure Low risk of occurring as data is hosted in the cloud and thus secured by the cloud provider. Data loss is hence mitigated by measures implemented by the cloud provider.	Loss of valuable company data and used in other operations of the company Web application unable to function	Data loss is hence mitigated by measures implemented by the cloud provider such as data replication
Inaccurate Forecasting	Moderate-High	Caused by the algorithm's inaccuracy to forecast out the optimal number of parts for SIAEC to provision or to keep as active stock Medium-High risk to occur as the forecasting may occur should the demand pattern changes across part numbers drastically due to unforeseen events such as the pandemic, especially in a shorter period of time.	Potential increase in investment costs for SIAEC, leading to over provisioning or unnecessary spending to maintain the active stock in their warehouse.	Will be using cloud service. Only physical disasters or virus attacks can result in a server failure

Annex B

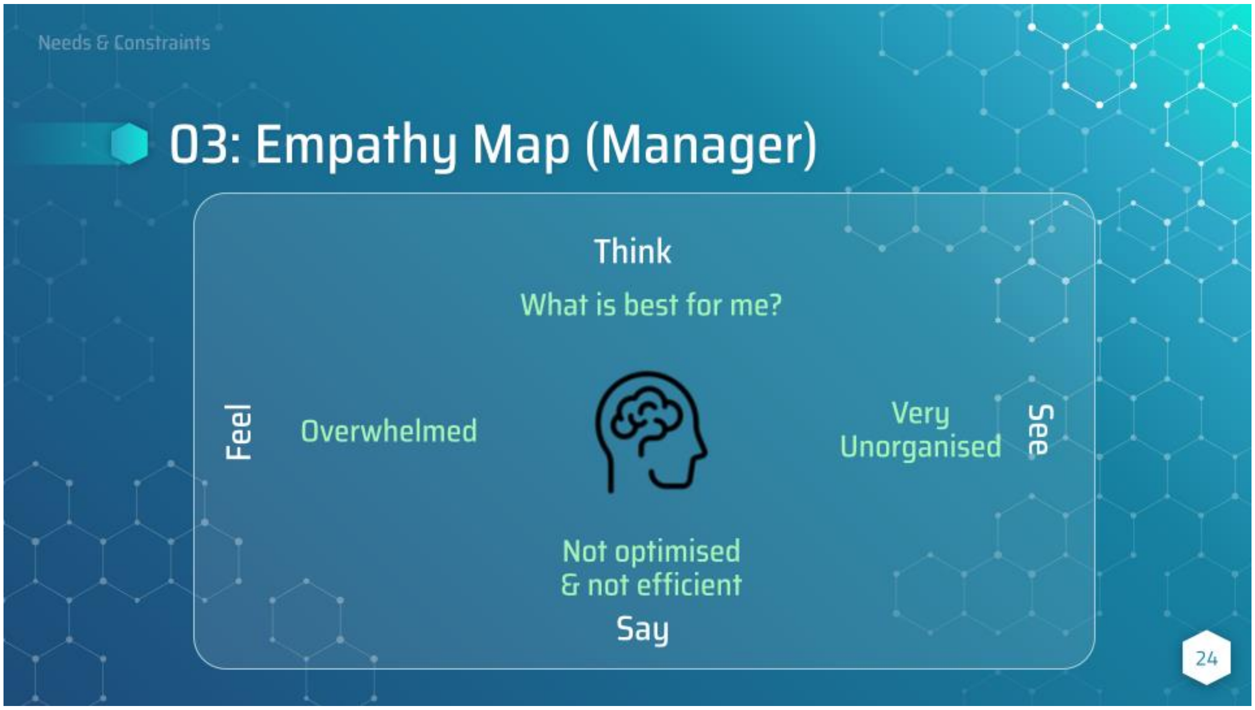
1. **Stakeholders**

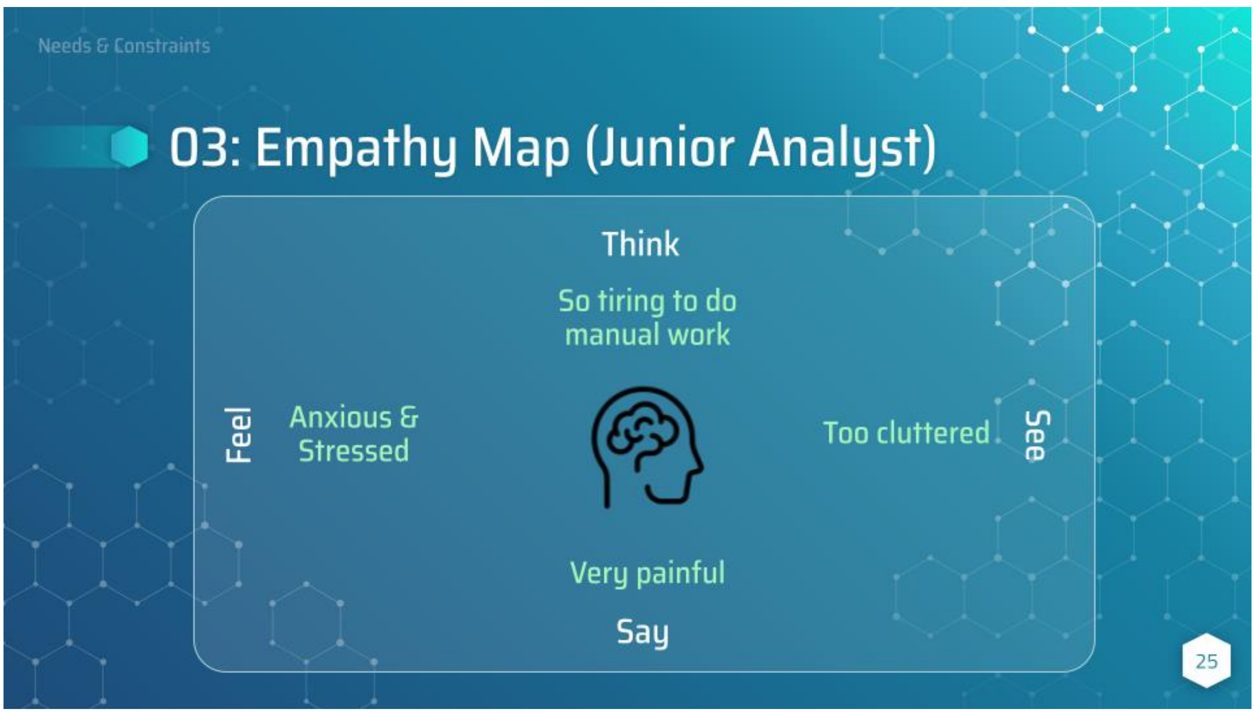


2. **Direct-end Users**



3. Empathy Map





4. Optimization model (formulation):

Binary Integer Program:

Formulae and Definitions:

x_i: The inventory quantity of i^{th} kind of rotable to be stored in warehouse

K: Number of rotables type in a particular ESS level

B: Binary variable; Entry 1 indicates that we need to purchase x_i rotable i, entry means we do not purchase x_i rotable

C: Cost; The cost incurred if we buy x_i of rotable i

λi: $failure\ rate,\ where\ \lambda i = \frac{QPA_i * \text{Fleet Size} * \text{FH}}{MTBUR_i}$

Ti: $Turn\text{-}Around\text{-}Time\ (TAT)$

P: Service level; service level if x_i of rotable i is purchased is calculated using Poisson Distribution

Where,

$$P = \frac{e^{-\lambda_i T_i} (\lambda_i T_i)^{x_i}}{x_i!}$$

A_0: Service level as required for each ESS class (e.g., ESS1 $\geq$ 98%; $ESS2\ \geq 95\%\ etc.$)

C_0: Arbitrary cost limit set by SIAEC

Formulation: (we have only included the primary constraints)

$$Minimize\ Cost = \sum_{r=1}^{n} \sum_{c=1}^{K} C_{rc} B_{rc}$$

$Subjected\ to$:

$$\frac{1}{n} \sum_{r=1}^{n} \sum_{c=1}^{K} P_{rc} B_{rc} \geq\ A_0\ [Average\ Service\ level\ should\ be\ at\ least\ the\ required\ level\ ]$$

$$\sum_{r=1}^{n} \sum_{c=1}^{K} C_{rc} B_{rc} \leq\ C_0\ [Total\ cost\ must\ not\ exceed\ SIAEC's\ budget\ ]$$

$$\sum_{c=1}^{K} B_{rc}\ [Each\ row\ of\ matrix\ B\ should\ sum\ up\ to\ 1\ ]$$

$$B_{rc}\ \in \{0,1\}\ [\ B_{rc}\ is\ binary\ ]$$

(For the constraints, r = 1, 2, .. K and c = 1,2,3,.. to arbitrary large number)

5. Optimization model (Results):

Results

Number	GRP	UPA	MTBR	Estimated # of Removal	Demand During Repair Transit	-1	0	1	2	3	4
6	21-89				0.28	0	75.7%	96.8%	99.7%	100.0%	100.0%
7	21-07				0.05	0	95.5%	99.9%	100.0%	100.0%	100.0%
16	21-13				0.08	0	92.2%	99.7%	100.0%	100.0%	100.0%
19	21-14				0.36	0	69.7%	94.9%	99.4%	99.9%	100.0%
90	21-61				0.08	0	92.2%	99.7%	100.0%	100.0%	100.0%

Figure 1: The original provisioning by part number level

Total investment:

USD 199,836

AVG ESS Level:

98.22%

Optimization: lpsolve() in R programming

Number	GRP	UPA	MTBR	Estimated # of Removal	Demand During Repair Transit	-1	0	1	2	3	4
6	21-89				0.28	0	75.7%	96.8%	99.7%	100.0%	100.0%
7	21-07				0.05	0	95.5%	99.9%	100.0%	100.0%	100.0%
16	21-13				0.08	0	92.2%	99.7%	100.0%	100.0%	100.0%
19	21-14				0.36	0	69.7%	94.9%	99.4%	99.9%	100.0%
90	21-61				0.08	0	92.2%	99.7%	100.0%	100.0%	100.0%

Figure 2: The optimised provisioning by total average ESS Level

Total investment:

USD 106,363

AVG ESS Level:

95.82%