2017

Dynamic passenger recovery model for airline disruption management

Almodad Muendo Mutinda
Faculty of Information Technology (FIT)
Strathmore University

Recommended Citation

Mutinda, A. M. (2017). *Dynamic passenger recovery model for airline disruption management* (Thesis). Strathmore University. Retrieved from http://su-plus.strathmore.edu/handle/11071/5711

DYNAMIC PASSENGER RECOVERY MODEL FOR AIRLINE DISRUPTION MANAGEMENT

Mutinda Almodad Muendo

091778

Submitted in Partial Fulfilment of the Requirement for the Award of a Master of Science Degree in Mobile Telecommunications and Innovation (MSc. MTI)

Faculty of Information Technology
Strathmore University

June, 2017

2017

Dynamic passenger recovery model for airline disruption management

Almodad Muendo Mutinda
Faculty of Information Technology (FIT)
Strathmore University

Recommended Citation

Mutinda, A. M. (2017). *Dynamic passenger recovery model for airline disruption management* (Thesis). Strathmore University. Retrieved from http://su-plus.strathmore.edu/handle/11071/5711

DECLARATION AND APPROVAL

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Mutinda Almodad Muendo

…………………………………………..

…………………………………………

Approval

The dissertation of Mutinda Almodad Muendo was reviewed and approved by the following:

Dr. John Olukuru,

Head of Data Science and Analytics,

@iLab Africa,

Strathmore University

Dr. Joseph Orero,

Dean, Faculty of Information Technology,

Strathmore University

Professor Ruth Kiraka,
Dean, School of Graduate Studies,
Strathmore University

ACKNOWLEDGEMENT

I would like to render my sincere gratitude to those who have made contributions towards this dissertation. First, I thank the Almighty God, the most gracious and merciful for giving me life, strength and wisdom to come up with this dissertation.

Secondly, special thanks to my supervisors Dr. John Olukuru (Strathmore) and Mr. Thomas Omondi (Kenya Airways) for believing in this dissertation and guiding me from the start to this far.

Lastly, I thank the MSC.MTI 2015 classmates and Strathmore University lecturers for their expert guidance and support throughout the research period.

ABSTRACT

Airline operations experience schedule disruptions every day. These schedule disruptions require intervention from the airline operations controllers through schedule recovery. In a hub and spoke airline network model, a disruption such as a flight cancellation can affect passenger itineraries in multiple fight legs, making it hard for airlines to re-accommodate disrupted passengers within a short time period. The current airline recovery solutions do not explicitly consider passenger recovery. This dissertation investigates the passenger recovery process by considering the challenges faced by passengers during a schedule disruption, the current solutions used to recover disrupted passengers and how a suitable solution can be designed, developed, tested and validated to ensure that it solves these challenges.

Data was collected from existing records of flight schedules and passenger bookings. The data collected was used as input to an optimisation model for passenger recovery. Scrum Agile Development methodology was adopted as the software methodology for developing the solution. A proof of concept web application was developed to make passenger recovery easier and reduce operational cost and passenger delay time. An optimisation model was developed based on IBM ILOG CPLEX optimiser to help solve disruptions faster. Testing was conducted by both the developer and a selected sample of airline industry users.

Keywords: passenger recovery, flight schedule, hub and spoke, optimisation.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AMQP – Advanced Message Queuing Protocol

API – Application Programming Interface

ATC – Air Traffic Control

CHT – Charter

DPM – Disrupted Passenger Metric model

ERD – Entity Relationship Diagram

FSC – Full Service

IATA – International Air Transport Association

IOCC – Integrated Operations Control Centre

IROPS – Irregular Operations

JKIA – Jomo Kenyatta International Airport

JSON – JavaScript Object Notation

KQ – Kenya Airways

LLC – Low-Cost

LP – Linear Programming

OPL – Optimisation Programming Language

PDM – Passenger Delay Metric model

POC – Proof of Concept

QoS – Quality of Service

REG – Regional

UAT – User Acceptance Testing

UML – Unified Modelling Language

CHAPTER ONE: INTRODUCTION

1.1 Introduction

This chapter covers the background of the study, research objectives, research questions, scope and significance of the study. There are many terms used in the airline industry that were of interest in this research. There are many terminologies used in the airline industry. Some of these are frequently used in this research and therefore it is good to define them. IATA defines a *flight leg* as the operation of an aircraft from one scheduled departure station to its next scheduled arrival station. Further, they define a *flight* as the operation of one or more flight legs with the same flight designator. When a flight is disrupted, passengers need to be rebooked to other flights as part of passenger recovery. Jafari et al. (2010) define *passenger recovery* as the reassignment of disrupted passengers to alternative itineraries, beginning at the disrupted passenger locations after their available times, and terminating at their destination, or a location nearby.

Most airlines utilise the hub-and-spoke network model to operate their flights. The hub-and-spoke network identifies a number of key major airports (hubs) with a large proportion of all flights scheduled between these and outlying airports (spokes) and very few flights occurring between the spokes (Maher, 2015). This network is characterised by passenger itineraries, a set of flights booked to travel between an origin and destination, generally involving a transit through at least one hub. The choice of a network model has a significant impact on the cost of disruption. There are many costs that incurred by the airline and passengers whenever a disruption occurs. These can be categorised in to three: *soft costs, hard costs and internalised costs.* According to (Eurocontrol, 2010), *soft costs* are costs borne by the airline (such as loss of market share due to passenger dissatisfaction). *Hard costs* on the other hand are costs borne by the airline (measurable, bottom-line costs such as re-booking and compensation) (Eurocontrol, 2010). Finally, *internalised costs* are costs borne by the passenger, not passed on to the airline (for example potential loss of business due to late arrival at meeting; partial loss of social activity (Eurocontrol, 2010).

1.2 Background of the Study

Passengers are of great value to any airline business. When disruptions occur, passengers are severely affected by the recovery decisions. Irregular operations (IROPS) can have a significant impact on airline costs, particularly direct costs. IROPS can also cause serious damage to an airline's image and reputation, especially when spread through social networks. From a passenger perspective, a journey disruption might simply mean any change from the original scheduled itinerary or reserved service. Missing an important business meeting or being late to an important event would be considered a major journey disruption, even if the passenger was impeccably re-accommodated by the airline. Passenger journey disruptions may carry unseen revenue impacts that are not being adequately assessed and quantified. For example, there may be a significant load factor impact when passengers with flexible tickets simply neglect the re-assigned flight provided by the carrier and find alternate transportation.

The cost of delay to airline operations comprises several components. These include the costs of passenger delay to the airline, plus crew and maintenance costs. The total cost is often dominated by the passenger component which can be split into 'hard' costs, such as those due to passenger rebooking, compensation and care, and 'soft' costs (Cook et al., 2009). Hard costs are typically difficult to fully attribute to a given flight due to accounting complications, but are, in theory at least, identifiable deficits in the airline's bottom line. The hard costs are higher as airlines pay more in recovery and care costs, such as meal vouchers and overnight accommodation. 'Soft' costs manifest themselves in several ways. Even with no experience of an airline, a passenger may perceive it to be unpunctual and choose another, instead. Due to a delay on one occasion, a passenger may defect from an unpunctual airline as a result of dissatisfaction (and maybe later come back). A passenger with a flexible ticket may arrive at an airport and decide to take a competitor's on-time flight instead of a delayed flight, on which they were originally booked. 'Soft' costs, exemplified by these types of revenue loss, are rather more difficult to quantify, but may even dominate the hard costs (Cramer & Irrgang, 2007; Cook et al., 2004). The soft costs are higher for longer delays, as passengers are more likely to be dissatisfied as the result of a longer delay than a shorter one.

Bad weather conditions have proved to be a major cause of disruptions especially in Europe. According to Eurocontrol (2010), the eruption of the Icelandic volcano *Eyjafjallajökull* in 2010 resulted in over one hundred thousand flight cancellations and affected more than ten million passengers. The number of complaints in the United Kingdom increased significantly due to the high amount of delays and cancellations; the Commission for Aviation Regulation of Ireland received in 2010 more than double the number of complaints than the previous year (Commission for Aviation Regulation, 2011). The rapid growth of social media in recent years has empowered passengers to share their experiences. This necessitates airlines to offer more to their passengers with regards to disruption of operations. The German airline, Lufthansa has suffered repeated strikes over the years affecting its reputation and seeing it losing customers (Amadeus, 2016). In 2014, Lufthansa apologised to customers through social media for a pilot strike, calling it the longest and hardest labour conflict in the airline's history. In 2015, the airline lost about €100 million to strikes with each day of the disruption costing it €10 to €15 million. The most recent strike in this airline was in November 2016 after pilots boycotted work to push for pay raise. Analysts noted that the airline's repeated strikes had also caused it lasting damage, with consumers opting to book with other airlines (Amadeus, 2016).

In December 2016, Kenya Airways' technical department went on a strike over poor pay, causing flight delays and disgruntled passengers. As Kenya Airways grappled with the immediate disruption, it issued a notice to passengers informing them: "We would like to advise our guests that while we have some delays due to a number of our technicians going on a go-slow, the majority of the flights are on schedule for the rest of the day" (Munguti, 2017). The effects of an airline strike have, however, been found to spread in multiple ways, and can further hurt the airline financially through reputational damage. According to a report by Amadeus (2016), the disruption problem spreads virally because the flight that was cancelled or delayed in one city was supposed to provide the aircraft for a departure from another city. The problems tend to keep growing and it is not long before everyone is unhappy. The estimated cost of disruption to airlines is eight per cent of airline revenue, or $60 billion worldwide.

When disruptions are not managed properly and timely, they will severely affect the airlines performance in terms of revenue, operational efficiency, and passenger satisfaction. In the event of a disruption, airlines can recover operations in a number of ways such as delaying flight departures, cancelling flights, rerouting aircraft, reassigning crews or calling in new crews, and re-accommodating passengers. The objective is to get feasible, cost-minimising plans that allow the airline to recover from the disruptions and their associated delays. In particular, most of the studies in literature have not focused on passenger services and re-assignment, that is, booking disrupted passengers onto new flights.

1.3 Problem Statement

Schedule disruptions require airlines to intervene through the process of recovery; this involves modifications to the planned schedule, aircraft routings, crew pairings and passenger itineraries. The current methods of passenger recovery are manual and therefore it takes a lot of time to recover normal schedules from a disruption. This leads to the airline losing a lot of revenue on passenger compensation and other disruption costs. Re-assigning large numbers of passengers quickly and efficiently is essential in order to protect airline revenues and minimise the negative impact on the passengers. A model that provides such a solution is missing leaving airlines without choice other than spend huge amounts of money to solve disruptions. Passengers on the other hand waste a lot of time waiting for disruptions to be solved thus affecting their whole journey.

1.4 Research Objectives

i. To identify the current methods of passenger recovery and their challenges.
ii. To review the existing solutions for passenger recovery.
iii. To design, develop and test a system for passenger recovery.
iv. To implement and validate the system.

1.5 Research Questions

i. What challenges are faced in the current methods of passenger recovery?
ii. What are the existing solutions for passenger recovery?
iii. How can a passenger recovery system be designed, developed and tested?
iv. Does the system solve the passenger recovery problem?

1.6 Research Justification

Passengers are of high importance to any airline, but their direct costs are not easily quantified. In most cases, they are not usually considered until the last stages of recovery processes. Therefore, it is common for them to be significantly affected by any disruption in the airline. Kenya Airways needed an effective solution that could help them make quick operational decisions during disruptions with minimal effect on passenger itineraries.

1.7 Scope

This research was carried out within the airline industry focusing on disruptions and recovery. A case of Kenya Airways was selected. This was due to the availability and willingness to provide data necessary for this research. The major focus was on dynamic passenger recovery problem as a subset of integrated airline recovery problem.

1.8 Limitations

This research was carried out within Nairobi. Data was collected using a small population because the airline staff worked in shifts of a few at a time. The implementation of the final product was on the CPLEX optimisation studio and the Django web framework. The choice of the optimiser was because the airline had partnered with IBM.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter includes the review of literature consisting of studies conducted worldwide in the related areas. With very little literature about passenger recovery, this study identifies gaps in the research area and possible ways to take knowledge of the gaps in the present research investigation. It provides an overview of the causes of disruption, general airline recovery, the cost due to disruption as well as existing solutions for passenger recovery.

2.2 Common Causes of Disruption

According to Amadeus (2016), there are ten common causes of disruption in an airline. Some of these causes include mechanical and technical problems with aircraft or support systems that take time to resolve, third-party issues such as problems with local transport networks connecting to the airport which for example, can lead to a build-up of late passengers in departures.

Additionally, crew logistics including but not limited to legal measures to protect staff can prevent them from working overtime to tackle disruption and flight crew duty limitations that must be observed. Weather issues such as fog, ice, snow, or heat that can negatively impact infrastructure are not exceptional. Health epidemics like passengers being taken ill can cause delays or the spread of a major viral infection can isolate a country or region.

2.3 General Airline Recovery

There are many ways in which airlines respond to disruptions. These include adjusting the flight schedules by several ways such as rescheduling flight take off, flight cancellation, aircraft rerouting, crew reassignment, and passenger re-assignment. Most research show that the crew recovery problem is a very complex and difficult problem having a significant impact on the operational cost of an airline. Similar to the aircraft recovery problem, several unique methods are suggested in order to recover model runtimes. The crew recovery problem permitting the use of flight delays and cancellations is more complex problem than the comparable fixed flight schedule problems. Lettovsky, Johnson, and Nemhauser (2000) present a crew recovery problem that introduces the use of flight cancellations. In their research, a number of approaches to reduce the computational time of the recovery algorithm are proposed, including a search scheme to identify the undisrupted crew and compact storage for the generated columns. The time-band network was presented by Bard et al. (2001) for the airplane switching problem in reaction to delays and grounding. Barnhart, Kniker, and Lohatepanont (2002) have shown that official government and air transportation methods that are flight-centric do not precisely reveal the passenger experience but underrate disruptions on the passenger trip time as a result of missed connections and flight cancellations. An innovative approach to the crew recovery problem is developed by Abdelghany, Ekollu, Narasimhan, and Abdelghany (2004), partitioning flights by their resource independence. The partitioning process reduces the solution runtimes by defining a series of distinct recovery problems that are more readily solvable than the original problem. A solution by Stojkovi´c and Soumis (2005) introduced flight delays and constructing individual pairings for each crew member. According to Ball, Barnhart, Nemhauser, and Odoni (2007), the objective is mainly to get cost-minimising strategies that enable an airline recover their normal schedules.

The aircraft recovery problem was solved by Eggenberg et al. (2007) using a column generation algorithm grounded on the time-band network. Flight-based metrics excludes the trip interruptions accrued by passengers who are re-booked due to missed connections or cancelled flights. Also, flight-based metrics do not quantify the magnitude of the delay (only the likelihood) and thus fail to provide the consumer with a useful assessment of the impact of a delay (Wang, Sherry, Xu, & Donohue, 2007). Hu et al. (2011) developed an integer programming model using a time-band network. The solutions for a single-fleet aircraft recovery were found using Lingo solver by solving the linear programming slackening and then applying a rounding heuristic to find an integer viable result. The data used for the computation was only associated with twelve aircraft. The model was later extended by Hu et al. (2015) to a multi-fleet passenger transiting and aircraft routing optimisation. The solutions were found directly using CPLEX Solver. Moreover, both studies made the assumption that passenger journeys consist of a one flight leg.

2.4 Aircraft Recovery Problem Including Disrupted Passengers

Airline scheduling is a complex and time-consuming process as there are several factors and multiple constrained resources involved on it. The schedule often faces with disruptions in operation phase. Now the object is to turn to the initial plan quickly and with the minimum cost. Bratu & Barnhart (2006) argue that recovery solutions need to be generated in less than 3 minutes, otherwise the recovery solution can become infeasible. In airline recovery problem, time horizon is smaller than scheduling phase and is called disrupted period or recovery period. Coordinators in an operation control centre usually have to make efforts to recover disruptions in this period and prevent their propagation in the airline schedule.

Figure 2.1 presents flights of a small network that consists of seven flights (numbered from f1 to f7) and three aircrafts (A/C1–A/C3). The horizontal axis is time and the vertical is aircraft. Each flight is represented using a thick line where start and end of the line represent the departure and arrival times and airports, respectively. The letters in the parentheses under the line show the itinerary of passengers (P1, P2, P3, and P4). As shown in Figure 2.1, flight f1 departs from airport G and arrives at airport H. Upon arrival of flight f1, aircraft A/C1 connects to flight f2 and passengers in itinerary P1 get to their next flight (f4).

Now suppose that flights f1 and f3 are delayed as shown in Figure 2.2. Therefore one or more of the downline flights that use aircrafts out of flights f1 and f3 and passengers who have other flights in their itineraries after f1 and f3 would be affected.

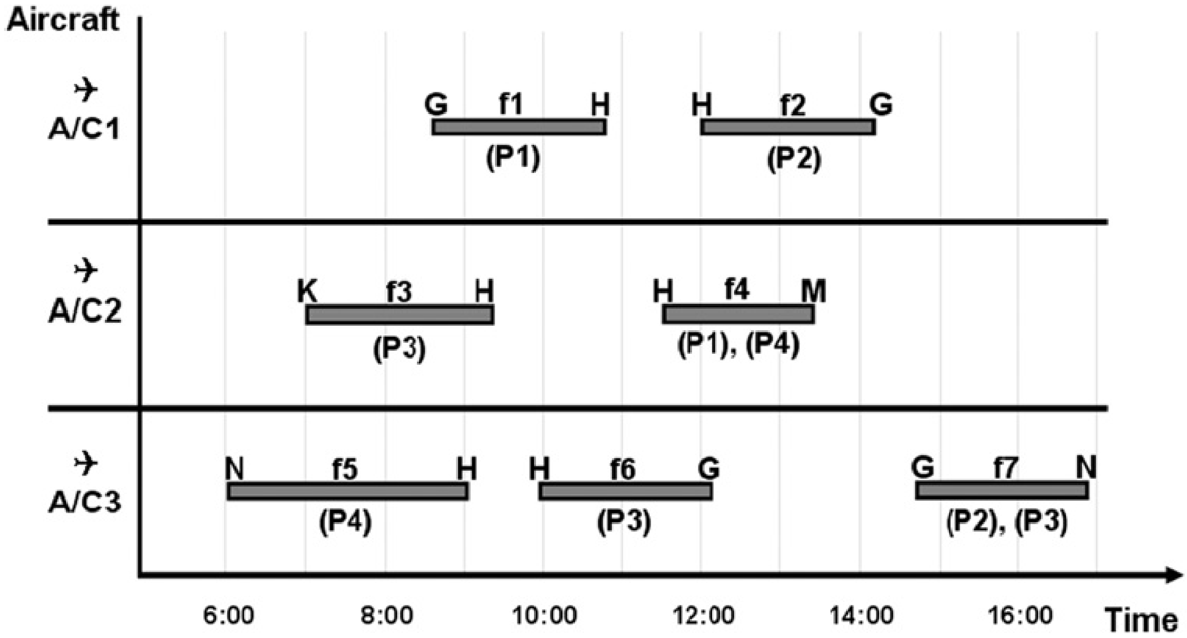


Figure 2. 1 Flights, Aircrafts and Passengers during Normal Operations. Source (Jafari & Zegordi, 2010)

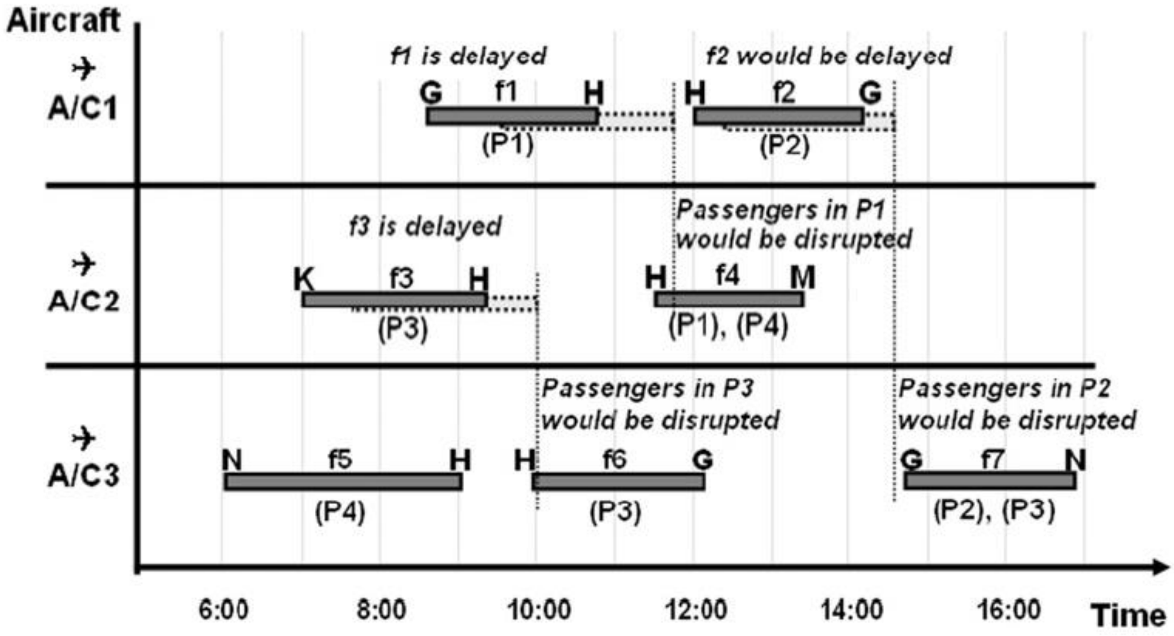


Figure 2. 2 Flights, Aircrafts and Passengers during Irregular Operations. Source (Jafari & Zegordi, 2010)

Passengers in itinerary P1 have two flights f1 and f4. Based on delay in arrival of f1, passengers in P1 arrive after departure of f4. Hence, they cannot get to their destination (M) and it means passengers in P1 are disrupted. Aircraft A/C1 arrives late, so f2 will fly as soon as A/C1 becomes ready. Therefore flight f2 would be delayed. As a result, passengers in itinerary P2 will miss their next flight (f7) because they do not have enough time to get out of arrival gate and go to the departure gate. Also as late arrival of f3, passengers in itinerary P3 will miss their next two flights f6 and f7 so they become disrupted. This case demonstrates the downstream of a flight disruption from the view of aircrafts and passengers and the importance of integrating aircraft and passengers while recovering a state of irregular operations.

2.5 Existing Solutions for Passenger Recovery

A passenger’s trip on-time performance (OTP) is a major performance indicator of the Quality of Service (QoS) provided by any air transportation system. QoS has been associated with productivity, airline profitability, customer satisfaction and customer loyalty (Heskett, Jones, Loveman, Sasser, & Schlesinger, 1994). According to Bratu and Barnhart (2006), low priority is given to passengers according to most literature on disruption management. The main difficulty in passenger recovery is the reassignment of disturbed passengers to other itineraries without delaying their schedule.

There are a few studies on passenger recovery discussed that are of interest in this research. Lettovsky (1997) integrated the costs associated with rebooking passengers in to the cost of flight cancellation, meals and hotel costs and estimated cost of losing the goodwill of passengers. Barnhart et al. (2002) designed the problem by like a multi-commodity network flow problem. Modelling approaches are presented by Clarke (2005) for re-accommodating disrupted passengers (who might be interrupted by schedule changes as a result of concerns such as income management operations). Bratu and Barnhart (2005) applied a dynamic heuristic known as the Passenger Delay Calculator (PDC). PDC allows some passenger recovery policies (such as first-disrupted-first-recovered or frequent flyers first) to be enforced.

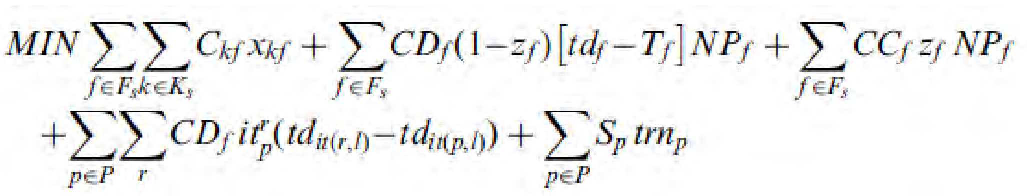
Two models, Passenger Delay Metric model (PDM) and the Disrupted Passenger Metric model (DPM) were proposed by Bratu and Barnhart (2006), and were grounded on a time-band network as presented in (Bard et al., 2001). In DPM, only were considered passenger recovery costs along with approximated delay costs were considered in the DPM. PDM on the other hand explicitly modelled passenger recovery options and disruptions in order to compute the delay costs with better accuracy. DPM was used to analyse three dissimilar disruption scenarios since PDM has not been proven suitable for the operations in real-world. A large neighbourhood search heuristic was developed by Bisaillon et al. (2011) by linking passenger reassignment and aircraft routing to solve the airline recovery problem with the main objective being to minimise the costs of operation and impact on passengers. There are three core phases of the heuristic which include are construction, repair and improvement phase. In each phase, a higher priority is given to the aircraft routes optimisation, and then based on optimised aircraft routes, passenger itineraries are reallocated. Extra steps were added to each phase by Sinclair et al. (2014) in order to improve the heuristic. A mathematical model that considers both passenger and aircraft problems was derived by Arıkan et al. (2013) by overlaying the passenger route network on top of the aircraft network. This problem was solved using CPLEX and was expressed as a mixed-integer nonlinear program. Likewise, a formulation was proposed by Chan et al. (2013) to integrate passenger and aircraft recovery with an objective of minimising airline operation cost and the sum of delay cost on passengers. In their formulation, airline operations costs comprised of an inconvenience cost of transferring passengers to alternative airlines and the cost of late flight arrivals. Nonetheless, techniques to solve the problem were not given.

The major shortcoming of aforementioned studies is that passenger itineraries are not explicitly considered in their models apart from Bratu and Barnhart (2006) which focuses on passengers, but integrate flight departure rescheduling, cancellation decisions and relax restrictive functional restrictions. This implies that passengers are given the last priority and their re-accommodation is always done after aircraft and crew recovery. As it is declared by some research like Dorndorf, Jaehn, Lin, Ma, and Pesch (2007) and Kohl, Larsen, Larsen, Ross, and Tiourine (2004), it appears that there are still challenges of a combined crew, passenger and aircraft recovery model hence opening gaps for more research. In the KQ context, when disruptions occur, disrupted passengers should be reallocated to alternative itineraries that are available with the same end points, without affecting other passengers who are not disrupted.

2.5.1 Mathematical Model

Jafari & Zegordi (2010) developed a simultaneous recovery model for aircraft and passengers whose general idea was based on (Abdelghany et al., 2004) and (Bratu & Barnhart, 2006). However, they employed *aircraft rotations* and *passengers' itineraries* instead of flights. Aircraft rotation is the sequence of flight legs that are flown by an aircraft beginning and ending at the same station. The model includes flight recovery, aircraft recovery and passenger recovery problems but it does not include crew recovery, route planning, gate assignment, ground staff and catering planning problems. This model assumes that the maximum delay allowed for a single flight, minimum time required for each flight and aircraft to turnaround, and minimum connection time is needed for passengers to leave a flight to arrive to the next flight, are fixed. The mathematical formulation of this model whose objective function is to minimize the total cost associated to recovering all flights, aircrafts, and passengers in the recovery scope is shown in equation 2.1.

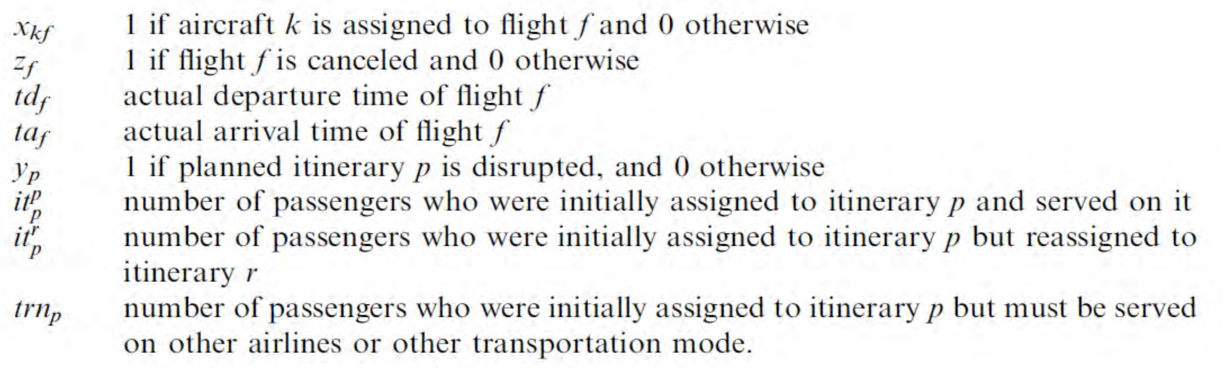
Equation 2.1: Mathematical Formulation. Source (Jafari & Zegordi, 2010)



$$MIN \sum_{f\in F_s}\sum_{k\in K_s} C_{kf}\,x_{kf} + \sum_{f\in F_s} CD_f(1-z_f)\left[td_f - T_f\right]NP_f + \sum_{f\in F_s} CC_f\, z_f\, NP_f$$
$$+\sum_{p\in P}\sum_{r} CD_f\, it_p^r(td_{it(r,l)} - td_{it(p,l)}) + \sum_{p\in P} S_p\, trn_p$$

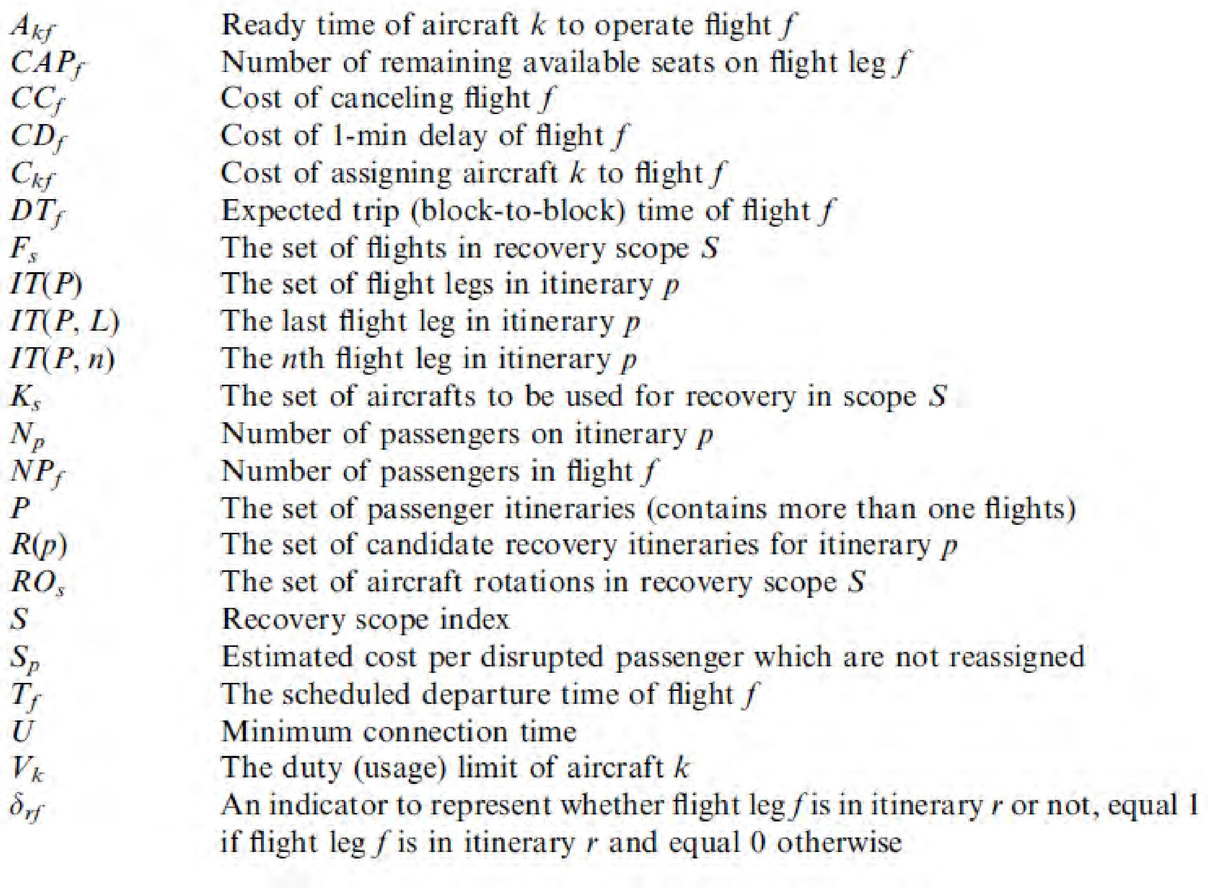
Where,

Decision variables:



x_{kf}	1 if aircraft k is assigned to flight f and 0 otherwise
z_f	1 if flight f is canceled and 0 otherwise
td_f	actual departure time of flight f
ta_f	actual arrival time of flight f
y_p	1 if planned itinerary p is disrupted, and 0 otherwise
it_p^p	number of passengers who were initially assigned to itinerary p and served on it
it_p^r	number of passengers who were initially assigned to itinerary p but reassigned to itinerary r
trn_p	number of passengers who were initially assigned to itinerary p but must be served on other airlines or other transportation mode.

Parameters:



A_{kf}	Ready time of aircraft k to operate flight f
CAP_f	Number of remaining available seats on flight leg f
CC_f	Cost of canceling flight f
CD_f	Cost of 1-min delay of flight f
C_{kf}	Cost of assigning aircraft k to flight f
DT_f	Expected trip (block-to-block) time of flight f
F_s	The set of flights in recovery scope S
$IT(P)$	The set of flight legs in itinerary p
$IT(P, L)$	The last flight leg in itinerary p
$IT(P, n)$	The nth flight leg in itinerary p
K_s	The set of aircrafts to be used for recovery in scope S
N_p	Number of passengers on itinerary p
NP_f	Number of passengers in flight f
P	The set of passenger itineraries (contains more than one flights)
$R(p)$	The set of candidate recovery itineraries for itinerary p
RO_s	The set of aircraft rotations in recovery scope S
S	Recovery scope index
S_p	Estimated cost per disrupted passenger which are not reassigned
T_f	The scheduled departure time of flight f
U	Minimum connection time
V_k	The duty (usage) limit of aircraft k
δ_{rf}	An indicator to represent whether flight leg f is in itinerary r or not, equal 1 if flight leg f is in itinerary r and equal 0 otherwise

The model is subject to 17 constraints, some of which have been considered in this study. The model was solved using LINGO solver but solving the model was difficult as the size of the problem increased.

The framework of simultaneous aircraft and passenger recovery model is presented in figure 2.3.

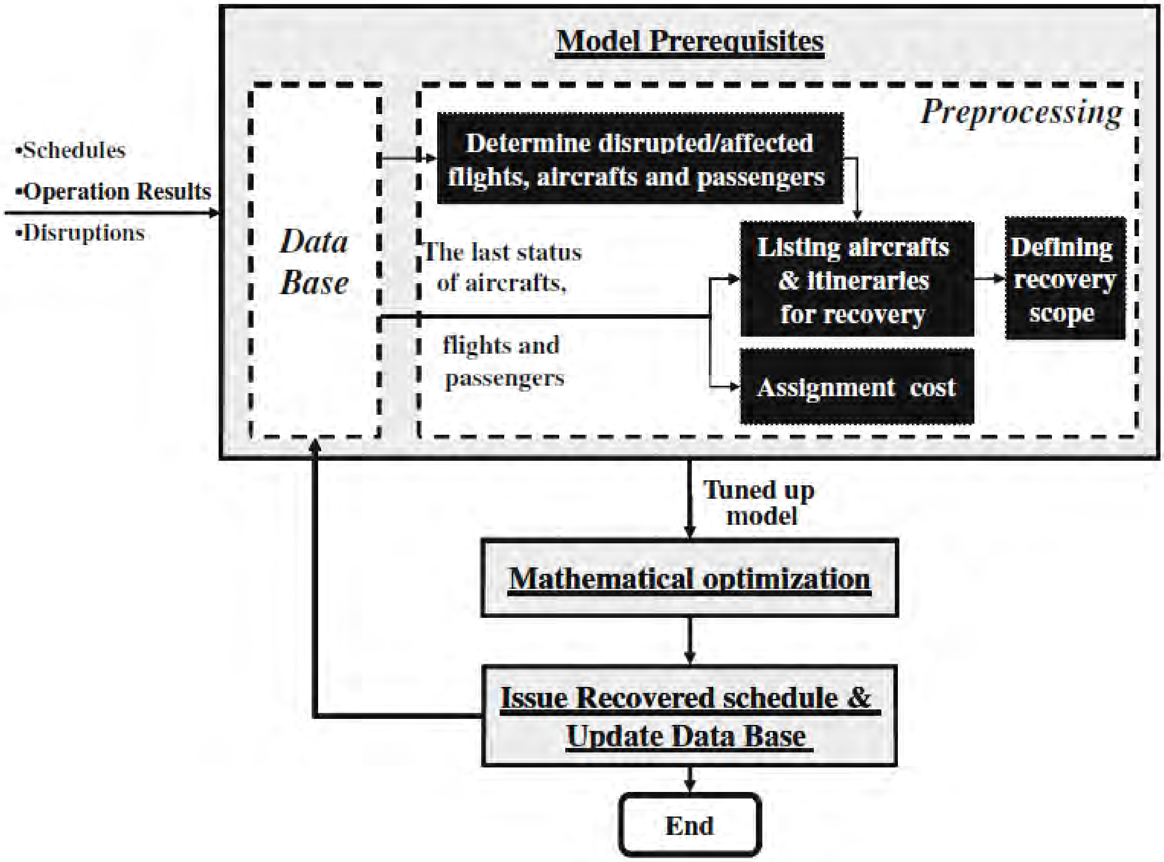


Figure 2. 3 Simultaneous Aircraft and Passenger Recovery Model Framework. Source (Jafari & Zegordi, 2010)

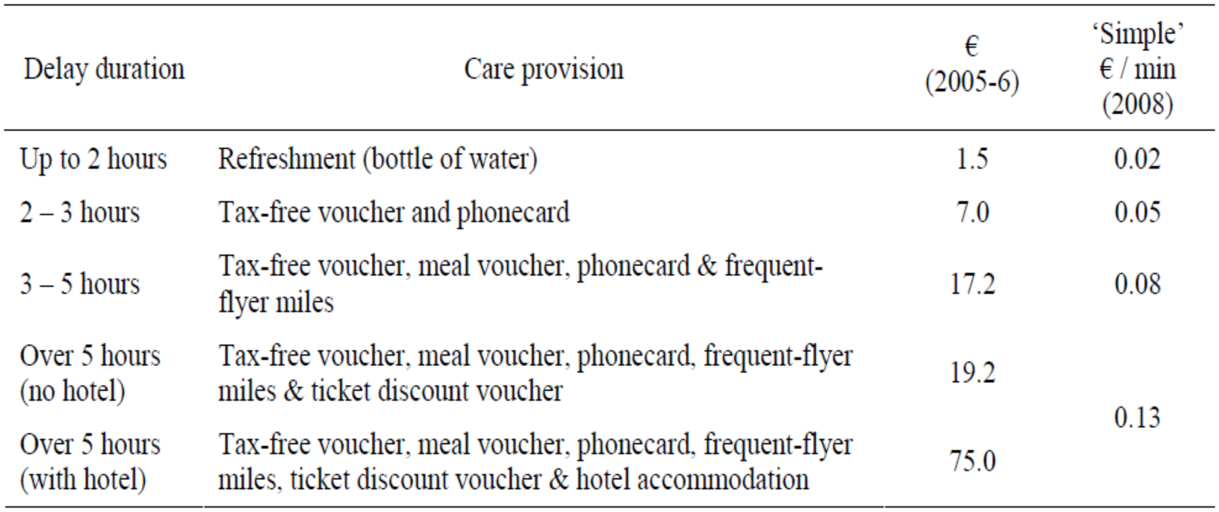
2.6 Costs Due to Disruption

Regularly, flights are for example, cancelled or delayed due to reasons such as bad weather, mechanical failures, security problems, or airport congestion. Rosenberger, Johnson, and Nemhauser (2003) account that 75% of disruptions as a result of weather conditions. The impact of disruptions on the economy is quite significant. Commonly, low priority is given to passengers according to most literature on disruption management. According to Bratu and Barnhart (2006), operational decision making is rarely driven by passenger perturbations. Regularly, crew and aircraft are first recovered, with a need to respect aircraft maintenance requirements especially for aircraft in critical conditions which require immediate maintenance. Kohl et al. (2007) suggest that the efficacious operation of an airline relies on the synchronised actions of all supporting functions. However, each group typically operates under its own directive, with its own budget and performance measures. Ball et al. (2007) refer to a research by Air Transport Association which accounts that delays cost airlines and passengers about $65 billion in the year 2000. When disruptions occur, airlines must react quickly to recover and minimise their impact. The widespread use of hub-and-spoke networks means that a disruption at a hub airport can have far-reaching effects quickly.

2.6.1 Care Costs by Delay Duration

Cook et al. (2009) argue that assuming a typical airport operation from 0630hours to 2230hours (sixteen hours), it could be estimated that of all five-hour delays, approximately one-third would delay passengers later than 2200, such that overnight accommodation would be required/supplied. This gives a simplified, combined estimate for the 'over 5 hours' category, of a certain amount. Increasing each of these costs by inflation and dividing by the number of minutes gives an initial estimate of costs per minute as illustrated in table 2.1.

Table 2. 1 Average Care Costs per Delayed Passenger



Delay duration	Care provision	€ (2005-6)	'Simple' € / min (2008)
Up to 2 hours	Refreshment (bottle of water)	1.5	0.02
2 – 3 hours	Tax-free voucher and phonecard	7.0	0.05
3 – 5 hours	Tax-free voucher, meal voucher, phonecard & frequent-flyer miles	17.2	0.08
Over 5 hours (no hotel)	Tax-free voucher, meal voucher, phonecard, frequent-flyer miles & ticket discount voucher	19.2	0.13
Over 5 hours (with hotel)	Tax-free voucher, meal voucher, phonecard, frequent-flyer miles, ticket discount voucher & hotel accommodation	75.0	

Source: Cook et al. (2009)

2.6.2 Hard Costs and Soft Costs

Hard costs include provision, compensation and transfer fees. Provision costs are generated due to departing delay due to the care of duty. According to Delgado, Martin, Blanch, and Cristóbal (2016), these costs are variable as a function of the airline model (full service (FSC), low-cost (LLC), charter (CHT) or regional (REG)) and the passenger fare type (premium or standard).

Compensation costs are based on Regulation 261 scheme on arrival delay (European Commission, 2004). Not all passengers seek compensation, for instance due to lack of awareness of their entitlement. It is estimated that 11% of passengers currently apply for compensation according to (University of Westminster, 2015). If a passenger misses its connection, an optimisation can be done to re-allocate the passenger to a following flight considering alliances and passenger fares types. This might lead to a cost to the airline in terms of transfer costs.

For the soft costs, the total arrival delay is considered within an estimation of the propagation of the delay. Soft cost computations are based on the cost of delay reported in (University of Westminster, 2015).

2.7 Technologies Used

The following technologies were used in most of the literature and were of interest in this research.

2.7.1 IBM ILOG CPLEX Optimisation Studio

CPLEX Optimisation Studio is an analytical decision support toolkit for rapid development and deployment of optimisation models using mathematical and constraint programming. It combines an integrated development environment (IDE) with the powerful Optimisation Programming Language (OPL) and high-performance ILOG CPLEX optimiser solvers (IBM, 2016a).

The Optimisation Programming Language (OPL) provides a natural representation of optimisation models, requiring far less effort than general-purpose programming languages. According to IBM (2016a), OPL has advanced data types designed for the special needs of optimisation problems, and it fully supports linear and quadratic objectives and constraints, as well as real and integer decision variables. It also supports both mathematical and constraint programming techniques.

A mathematical model that considers both passenger and aircraft problems was derived by Arıkan et al. (2013) by overlaying the passenger route network on top of the aircraft network. This problem was solved using CPLEX and was expressed as a mixed-integer nonlinear program.

Advantages of CPLEX

CPLEX provides the following advantages as outlined by (IBM, 2016b):

i. It provides high-performance optimisation engines for optimising business decisions.
ii. It makes it easy to develop and deploy optimisation models quickly by using flexible interfaces and prebuilt deployment scenarios.
iii. It provides capability to create real-world applications that can significantly improve business outcomes.
iv. Solving very large, real-world optimisation problems is possible.

2.7.2 Heuristic Algorithms

A heuristic algorithm is a problem-solving method that uses incomplete information to derive a potentially inaccurate or imprecise solution. Bratu and Barnhart (2005) applied a dynamic heuristic known as the Passenger Delay Calculator (PDC). PDC allows some passenger recovery policies (such as first-disrupted-first-recovered or frequent flyers first) to be enforced. Heuristics are intended to gain computational performance or conceptual simplicity, potentially at the cost of accuracy or precision. In optimised search, heuristic algorithms refine each iterative result until they find the result closest to the solution. Figure 2.4 illustrates the heuristic process.

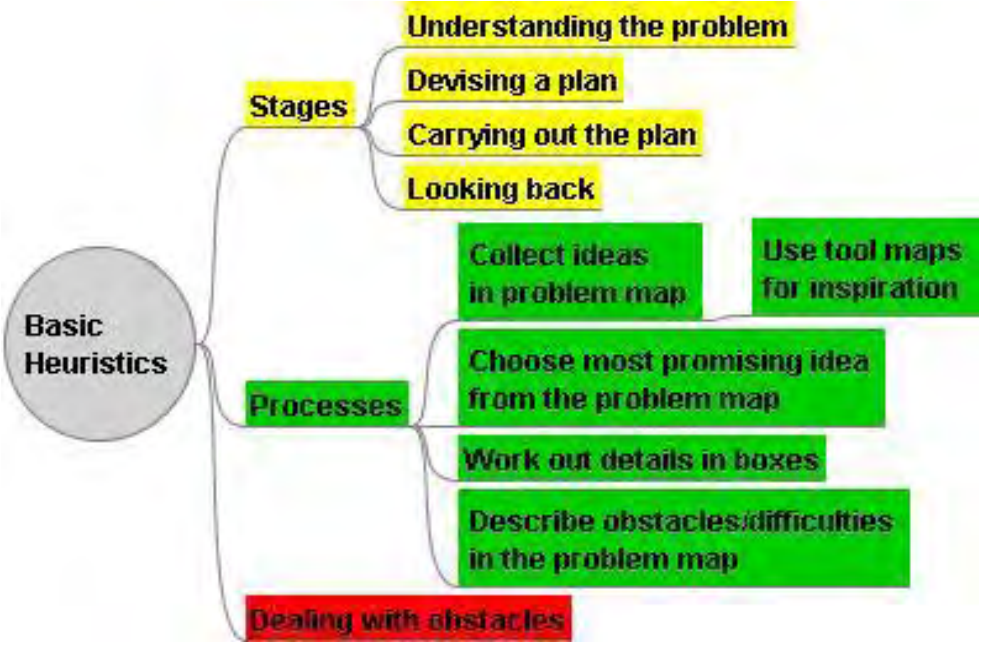


Figure 2. 4 Basic Heuristics Adapted from (Kashyap, 2011)

A large neighbourhood search heuristic was developed by Bisaillon et al. (2011) by linking passenger reassignment and aircraft routing to solve the airline recovery problem with the main objective being to minimise the costs of operation and impact on passengers. There are three core phases of the heuristic which include: construction, repair and improvement phase. In each phase, a higher priority is given to the aircraft routes optimisation, and then based on optimised aircraft routes, passenger itineraries are reallocated. Extra steps were added to each phase by Sinclair et al. (2014) in order to improve the heuristic.

2.7.3 Time-Band Network

Two models, Passenger Delay Metric model (PDM) and the Disrupted Passenger Metric model (DPM) were proposed by Bratu and Barnhart (2006), and were grounded on a time-band network as presented in (Bard et al., 2001). In DPM, only were considered passenger recovery costs along with approximated delay costs were considered in the DPM. PDM on the other hand explicitly modelled passenger recovery options and disruptions in order to compute the delay costs with better accuracy. DPM was used to analyse three dissimilar disruption scenarios since PDM has not been proven suitable for the operations in real-world.

2.8 Gaps and Limitations in Current Solutions

Firstly, passenger involvement is lacking as passengers are not considered when making recovery decisions and are therefore adversely affected during a disruption of normal operations. This has a negative revenue impact on the disrupted passengers and the airline. A passenger recovery model can help operations managers to generate feasible solutions within an acceptable time during a disruption.

Secondly, there is no technological way of notifying passengers of disruptions and recovery decisions affecting their scheduled itineraries. The current methods are manual and annoying to most passengers who end up being delaying for hours without prior notification of such delay. Having an efficient method of notifying passengers of disruptions can lead to better customer service and improved revenues for the airline.

A review of existing literature has shown that only the study by Jafari and Zegordi (2010) tries to solve the passenger recovery problem. However, their study focusses more on aircraft recovery. This research implemented part of the common parameters identified by (Jafari &

Zegordi, 2010). Many studies have assumed that passenger itineraries consist of only one flight leg. The consideration of multiple flight legs for passenger itineraries was a major contribution of this research.

The use of CPLEX optimisation studio takes more computation time but produces optimal solutions unlike heuristics which generate near-optimal solutions within a short computation time. However, there is model approach that uses both CPLEX and heuristics to generate optimal solutions within short computation times. This was another contribution of this research.

2.9 Summary

The current methods for passenger recovery are slow, unreliable and expensive. The proposal to come up with a web based decision support tool is justified given the expensive and undependable techniques that are used currently. A system that considers passengers during decision making will restore passenger trust and lead to significant increase in airline revenues.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter discusses the research methodology used in line with the research questions. It is organized into the following major sections: software development methodology, system analysis, system design, system implementation, system testing and evaluation.

3.2 Research Design

The purpose of this research was to develop a model for passenger recovery during disruptions of normal airline operations. Qualitative techniques were used to get the limitations of current systems, users' expectations of the new system and to see the number of people who would like to use the new system or think it is a good idea. The findings of this research led to the design, development, and deployment and testing of a model to help in the passenger recovery process and a proof of concept web application to test the model.

3.3 Scrum Agile Software Development Methodology

According to Larman (2004), Agile Methods comprise of a subsection of evolutionary and iterative methods and are grounded on repetitive improvement and adaptable development practices. Every iteration is a mini-project which is self-contained with activities that cover requirements analysis, planning, design and development, testing and retrospection (Boehm, 2007).

The rationale of adapting short iterations to make it possible for new information and response from iterations N and earlier result to the enhancement of iteration N + 1.

Boehm (2007) argue that end users adaptively specifies their requirements for subsequent releases based on their observation of the growing product, instead of assumptions at the beginning of the project. Iteration length is filled by choosing scope for every iteration. Instead of increasing the iteration length to fit the selected scope, the scope is condensed to fit the iteration length.

Agile methodology was used in this research because of the following reasons as highlighted by (Sultanía, 2015):

- Focus on customer as a priority.
- Requirement change adaptation at any stage.
- Frequent delivery of working software.

- Business and developing team members work together.
- Work is defined for the specified time.

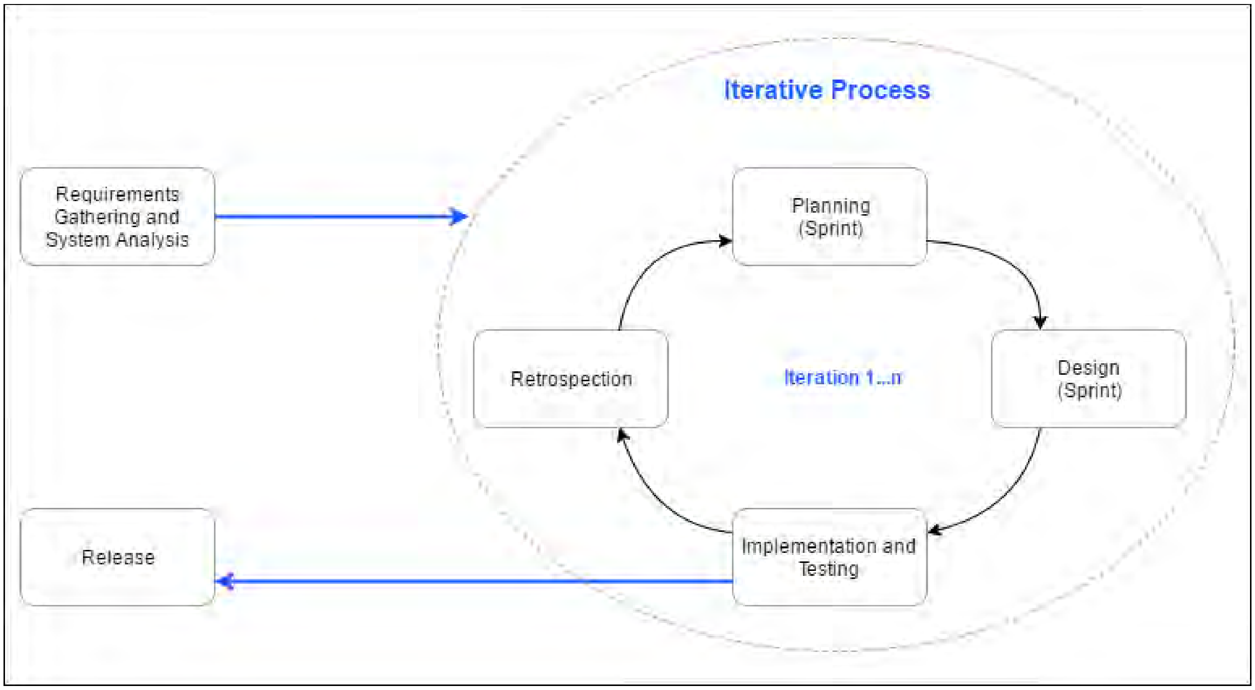


Figure 3. 1 Agile Software Development Methodology with Scrum Adapted from (Boehm, 2007)

As shown in figure 3.1, Agile projects proceed in an iterative fashion where new features are integrated to extend the capabilities of the software. That is, each sprint delivers user-desired, working, and tested features. Each iteration generally consists of four distinct phases: *planning*, *design*, *implementation* and *testing*, and *retrospection*.

3.3.1 Requirements Gathering and System Analysis

In this phase, high-level requirements are gathered as well as the scope of the project. This allows developers to quickly begin coding in order to find out what works even quicker (Boehm, 2007). This phase was used to gather information about the decision variables, constraints and information necessary to build an optimisation model for passenger recovery.

Data on disruption management was collected from existing records through document reviews. The respondents who helped in system analysis included passenger operations staff at KQ Integrated Operations Control Centre (IOCC), duty managers at IOCC, head of passenger services at KQ and the head of airport security at Jomo Kenyatta International Airport (JKIA). Appendix D shows the system analysis interview guide.

i. Location of the Study

This study was carried out at KQ focusing mainly on passenger itineraries and flight disruptions. In order to come up with efficient ways of recovering from irregular operations, the IOCC provided data on the current methods of recovery.

ii. Target Population

The target population is the group of elements to which the researcher wants to make inference (Fricker, 2013). The target population performed two functions. First, it provided data and secondly it tested the model using the final Proof of Concept (POC). The target population comprised of 7 employees at IOCC, 2 duty managers at IOCC, 1 head of passenger services at KQ, 1 head of security at JKIA and 1 employee at Air Traffic Control (ATC). This resulted to a target of 12 respondents in total.

In addition, the total number of KQ aircraft and the number of flights per aircraft per day was considered. A sample was selected from the aircraft and flight population and used as input in the optimisation model for passenger recovery.

3.3.2 Planning

This is the first phase of the iterative process and its core purpose is to decide and document which new features are to be added to the software, or what changes to existing features need to be made (Boehm, 2007). This phase comes to an end when all stakeholders agree upon the features to be implemented within the given time and resource constraints. Planning helped to identify the resources that are needed to implement the model.

3.3.3 Design

According to Boehm (2007), this phase involves modelling and development of one or more features agreed upon between the various stakeholders during the planning phase.

i. Design

Unified Modelling Language (UML) as defined by Dennis, Wixom, and Tegarden (2012) was used as the modelling language to model design diagrams and to offer clarifications on user requirements. The tools for modelling included use case diagram, sequence diagram, context diagram and entity relationship diagram.

Use case diagram and its corresponding use case descriptions were used to model the system functionality. The system functionality was identified and partitioned using the use cases and that made it easy to separate the system into actors and use cases (Dennis et al., 2012). The use cases were represented as a text that describes the action the user is effecting on the system

Sequence diagram was used to show information passing between the main entities of the system to model the system flow according to (Dennis et al., 2012). It showed how objects interact with each other sequentially.

A Context Diagram as highlighted by (Le Vie & Donald, 2000) was used to define the boundary of the system and its interactions with the critical elements in its environment. It gives a single diagram that has the system of interest at the centre, with no details of its interior structure or function, surrounded by those elements in its environment with which it interacts.

Entity relationship diagram (ERD) showing the tables, their attributes and their relationships Dennis, Wixom, and Tegarden (2012), was used to model the database. This enabled the researcher to create different objects with actual real life relationships. A database schema showing the fields, data types and their descriptions was used to model the tables, triggers and views.

3.3.4 Implementation and Testing

i. Implementation

This is the actual development of the system based on the design produced in the design subsection. The system consisted of an optimisation model based on linear programming and a proof of concept which was a web based application. Data was stored in a central MySQL database.

a) Optimisation Model

Linear programming (LP), involves minimising or maximising a linear objective function subject to linear constraints (Tamba & Joelianto, 2016). The constraints may be equalities or inequalities. LP was used to develop the model based on the decision variables identified from the study and subject to predefined constraints. The decision variables were the Boolean variables which determine the reassignment of passengers in the event of a disruption.

There was one objective function of the optimisation model. This was to minimise the total cost of recovering all disrupted passengers considering constraints in the recovery scope. A detailed explanation of the mathematical formulation will be covered in chapter four. This will include the decision variables, parameters, indices, objective function and constraints.

b) Web Application

The web application was developed as a proof of concept to test the optimisation model. It was developed using the Django framework explained by Holovaty, Kaplan-Moss, and Gilmore (2008) and was hosted on the Apache HTTP server (Bowen & Coar, 2007).

c) Database

MySQL relational database management system explained by Ramakrishnan (2000) was used to design the database. This is because it is open source, secure and has a huge online development support community.

d) RabbitMQ

The Python library for RabbitMQ open source message broker software as highlighted by Videla and Williams (2012) was used to queue input data files using the Advanced Message Queuing Protocol (AMQP) that RabbitMQ implements.

ii. Testing

In this phase, the testing team decides if the software is correct and complete (Boehm, 2007). Completed features are removed from the list of features needing another planning sprint, and incomplete features are again candidates for future iterations. The system went through the following types of testing:

i. Compatibility Testing

This was to test the compatibility of the web application on different web browsers (Firefox, Internet Explorer, Chrome, Opera and Edge) running in different operating systems (Windows and Ubuntu). This was done by eight employees of KQ by running the application in their different computers.

ii. Load Testing

This was done to measure the amount of time the model takes to process multiple requests simultaneously and produce feasible solutions. Load testing was done through experiments by the researcher.

iii. Integration Testing

This was conducted to ensure that the various system modules work as expected after their integration. This was done through experiments.

iv. User Acceptance Testing (UAT)

User acceptance testing to measure user satisfaction and feedback was done to help in validating the system. UAT was conducted by eight employees of KQ.

3.3.5 Retrospection

In this phase, the development team meets to reflect on the last iteration and discuss those tasks, techniques, and team interactions that worked and those that need improvement (Boehm, 2007). The final system was evaluated by the selected industry users, to establish whether it is valid, and if the research objectives stated in chapter one were met. This was essential as it showed if the system helped in passenger recovery.

3.4 Summary

This chapter has described the methods and processes that were used to collect data and the methodologies that were used to answer the research questions. It has also helped to decide on the target population were data was collected from and to test the final proof of concept and validate it.

CHAPTER FOUR: REQUIREMENT ANALYSIS, ARCHITECTURE AND DESIGN

4.1 Introduction

This chapter covers the system analysis, architecture and design of the proposed system. Design diagrams were drawn showing the design and architecture of the system. The design was done using Unified Modelling Language (UML) diagrams which include use case diagram, data flow diagram, context diagram, system sequence diagram and entity relationship diagram.

4.2 Requirement Analysis

System requirements were collected through document reviews and content analysis. These helped derive the functional and non-functional requirements of the system. However, assumptions were made based on interactions with operation controllers and observations. These assumptions are: If the airline schedule is subjected to any source of irregularity at one or more airports, which results in delaying or cancelling some flights, information on these affected flights is made available to project all potential downline disruptions in the scope due to the introduced delays or cancellations. If a flight leg disruption is projected, remaining affected flights legs in the route of this flight are considered disrupted.

4.2.1 Functional Requirements

Functional requirements defines the functionality that a system or one of its subsystems must perform successfully. They include:

i. **Create Account**

The system should allow the administrator to create user accounts and assign them permissions. It should provide an interface to enter the required information.

ii. **Login**

All users must be authenticated using their username and password for them to access the system.

iii. **View Flight Schedule**

All authenticated users should be able to view flight schedule.

iv. **Upload Flight Schedule**

All authenticated users should be able to upload new flight schedule.

v. **View Disruptions**

All authenticated users should be able to view previous disruptions and their solutions.

vi. **Recover Passengers**

All authenticated users should be able to run the optimiser and recover passenger itineraries when there is a disruption.

vii. **View Passenger Itineraries**

All authenticated users should be able to view passengers and their itineraries.

viii. **View Recovery Reports**

The duty managers should be able to view passenger recovery reports

ix. **Manage System Configuration**

Authenticated system administrator should be able to perform administrative duties.

4.2.2 Non Functional Requirements

These requirements do not affect the primary functionality of the system but they enhance the system quality. They include:

i. **Performance** - The system should perform its functions with a fast and reasonable response time.

ii. **Usability**- The system's user interface should be self-explanatory and easy to use.

iii. **Security** - The system should restrict its access only to authenticated and authorized users with a clear audit trail of user activity.

iv. **Integrity**- The data stored in the system should not be modified by unauthorised users.

v. **Scalability** - The system should be able to grow while maintaining its quality and without extra coding.

vi. **Reliability and availability** - The system should be fault tolerant to enhance reliability and should be available to perform user tasks anytime.

4.3 System Architecture

The Client Server Architecture was adopted for the development of the proof of concept. This was based on the requirements gathered. The architecture consists of four major components, namely: web client, optimiser, file watcher service, queue server, database server and SMS service. The web client is a web portal that users use to perform optimisation. The optimiser is called when a user wants to solve a disruption and recover passengers via the web client. The optimiser stores its solution(s) in files. The file watcher service detects the new files and publishes them to a specific queue. The queue server passes the queued files to a consumer for reads the data and store it in the database. The consumer then invokes the SMS service which notifies the affected passengers of their new flight schedules and itineraries. Figure 4.1 shows the system architecture.

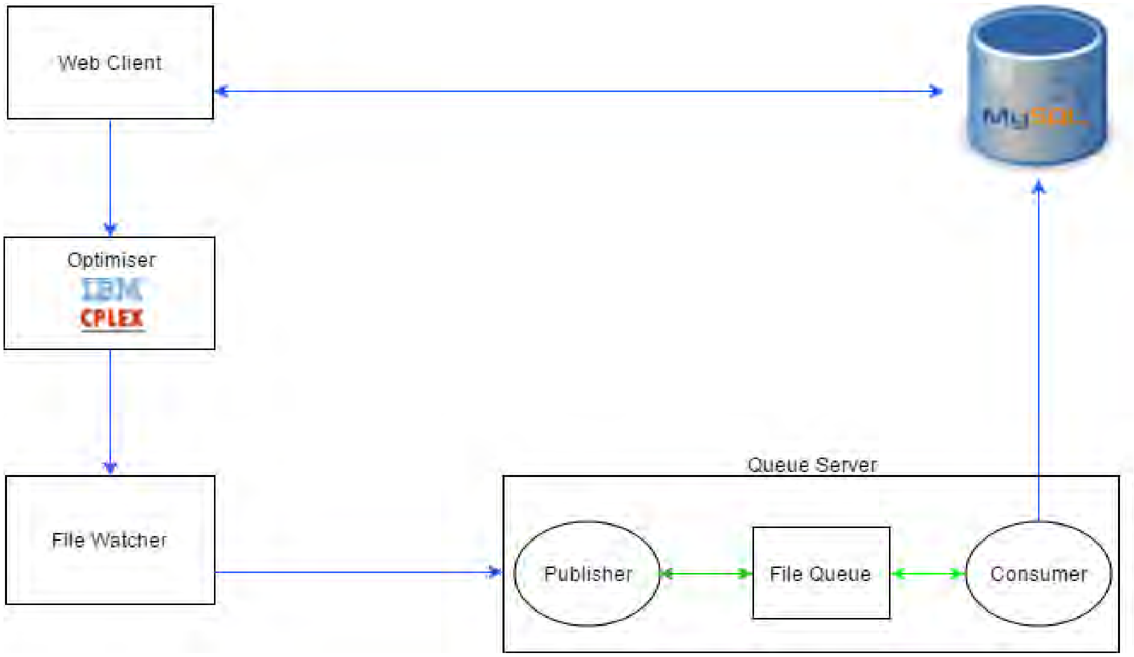


Figure 4. 1 System Architecture

4.4 Optimisation Model

This section describes the passenger recovery model in more detail. A linear programming model was implemented in order to achieve the objectives of this research. Figure

4.2 shows the architecture of the model.

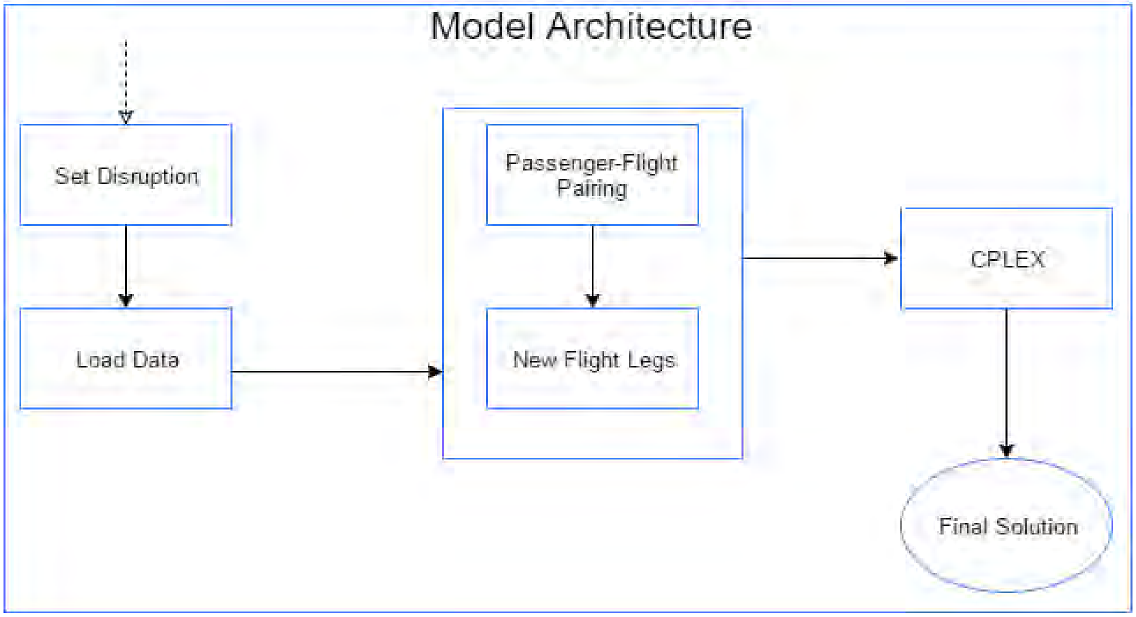


Figure 4. 2 Model Architecture

4.4.1 Mathematical formulation

The mathematical formulation of this model is based on the model by (Jafari & Zegordi, 2010) but with more variable and parameters specific to passenger recovery. This is a linear programming model which is expressed in a mathematical formulation. The mathematical formulation consists of four major elements which are: the objective function, decision variables, constraints and parameters. These elements are discussed in detail and the relevant mathematical formulae given.

a) Decision variables, parameters and indices

The decision variables are the Boolean variables which determine the reassignment of passengers in the event of a disruption. Their values are therefore, either 1 or 0 and they are as given below:

Decision Variables:

Xw_f = 1 if aircraft w is assigned to flight f and 0 otherwise

C_f = 1 if flight f is cancelled and 0 otherwise

Y_j = 1 if planned itinerary j is disrupted, and 0 otherwise

$IT_{k,j}$ = number of passengers who were initially assigned to itinerary j but reassigned to itinerary k

NK_j = number of passengers who were initially assigned to itinerary j but must be served on other airlines or other transportation mode.

The parameters of this model are as highlighted below:

Parameters:

Np_f = Total number of passengers on itinerary j

NPe_f = Number of economy class passengers in flight f

NPb_f = Number of business class passengers in flight f

BC_m = Business class multiplier

P = The set of passenger itineraries

Sj = Estimated cost per disrupted passenger which are not reassigned

NR_{cr} = Number of required crew to operate flight f

NA_{cr} = Number of available crew to operate flight f

CAP_f = Number of remaining available seats on flight leg f

CC_f = Cost of cancelling flight f

CT_f = Cost of one minute delay of flight f

Cw_f = Cost of assigning aircraft w to flight f

F = The set of flights in recovery scope S

FL = The set of flight legs in itinerary j

$IT(P, L)$ = The last flight leg in itinerary j

$IT(P, n)$ = The nth flight leg in itinerary j

DS_c = The set of disruption costs

RO_n = The set of recovery options

K_s = The set of aircrafts to be used for recovery in scope S

$Ttrip_f$ = Expected trip (end-to-end) time of flight f

Td_f = The scheduled departure time of flight f

Dtj = The departure time of itinerary j in flight f

Dtk = The departure time of itinerary k in flight f

There are indices used in this model to identify flights, passengers and itineraries. The indices are described below:

Indices:

f = flight index

m = Business class multiplier index

j = Itinerary index

s = Scope index

b) Objective Function and Constraints

The objective function of this model is to minimise the total cost of reassigning all disrupted passengers subject to the constraints in the recovery scope. It is the summation of aircrafts assignment cost, total delay cost, cancellation cost and other costs associated with disrupted passengers such as meals and accommodation. This results in equation 4.1 which represents the objective function.

The first term in the objective function is to recover open positions of each disrupted flight by using the most efficient aircrafts in the system. The second and third terms promote reliable operations by minimising flight delay and cancellation, respectively. The fourth and fifth terms recover disrupted passengers through reassigning them to the earliest available itinerary or transport them to the destination by any other means, partner airlines or an alternative means of transport.

$$Minimise\left(\sum_{f=0}^{Fs}\sum_{k=0}^{Ks}Cwf.Xwf+\sum_{f=0}^{Fs}CTf(1-Cf)[\mathrm{Ttripf}-Tdf]Npf\right.$$
$$+\sum_{f=0}^{Fs}(CCf.Cf.NPef)+\ BCm(CCf.Cf.NPbf)$$
$$\left.+\sum_{j=0}^{P}\sum_{k}CTf.\mathrm{ITkj}(\mathrm{Dtk}-Dtj)+\sum_{j=0}^{P}Sj.Nkj\right) \quad (4.1)$$

The objective function is subject to constraints which act as control functions. Equation 4.2 represents the first constraint which ensures that a passenger cannot be assigned to a cancelled flight.

$$\sum_{p\in NPf}(Fip\,.Yj)-Cf\ =1 \qquad \forall i\ \in F \qquad (4.2)$$

The second constraint ensures that a passenger cannot be assigned to a flight leg with zero seats remaining. This is depicted in equation 4.3.

$$\sum_{f\in F}(FLi\,.CAPf)>0 \qquad \forall i\ \in FL \qquad (4.3)$$

The equation 4.4 illustrates the constraint that a flight can only be operated if the number of required crew to operate it is available.

$$\sum_{f\in F}(FLi.NAcr-FLi.NRcr)\geq 0 \qquad \forall i\ \in FL \qquad (4.4)$$

4.5 System Design

The system design was presented using the UML design diagrams. The following sections illustrate the interaction between different components of the system and how data flow is modelled in the system. The diagrams include use case diagram, data flow diagram, context diagram, system sequence diagram and entity relationship diagram.

4.5.1 Use Case Diagram

A use case diagram gives a visual representation the interaction between actors of the system and the system (Dennis et al., 2012). An actor can be a person or another system. The following are the actors of the system:

Operations Staff - These are the employees working at the IOCC who deal with disruptions and passenger recovery.

Duty Manager – This is the operations manager in a given duty period such as 12 hours.

System Administrator – This is the overall system administrator who has the highest system permissions.

Head of Passenger Services – This is the person in charge of all passenger services in the airline. There main interaction with the system is when viewing reports about passenger recovery.

The following are the main processes that the system can perform.

i. *Manage Users* – The primary actor is the System Administrator who performs CRUD operations on users.
ii. *Manage Disruption Costs* – The primary actors are the Duty Managers who define the cancellation and delay cost matrices.
iii. *Recover Passengers* – The Operations Staff are the major actors and run the disruption solver in order to recover passengers.
iv. *Upload Flight Schedule* – The primary actors are the Operations Staff.
v. *View Recovery Solutions* – This allows the Operations Staff to view a pool of recovery solutions generated by the optimiser.
vi. *View Disruptions* – The primary actors are the Operations Staff who can view records of previous disruptions.
vii. *Manage Recovery Options* – The Duty Manager is the primary actor and is able to create and amend recovery options.

viii. *Manage Settings* – The Duty Manager is the primary actor and manages the various parameters used as input in the optimiser.

ix. *Generate Reports* – The Head of Passenger Services is the primary actor and generates and views the reports after a disruption has occurred.

Figure 4.3 shows the use case diagram of how different actors use the system.

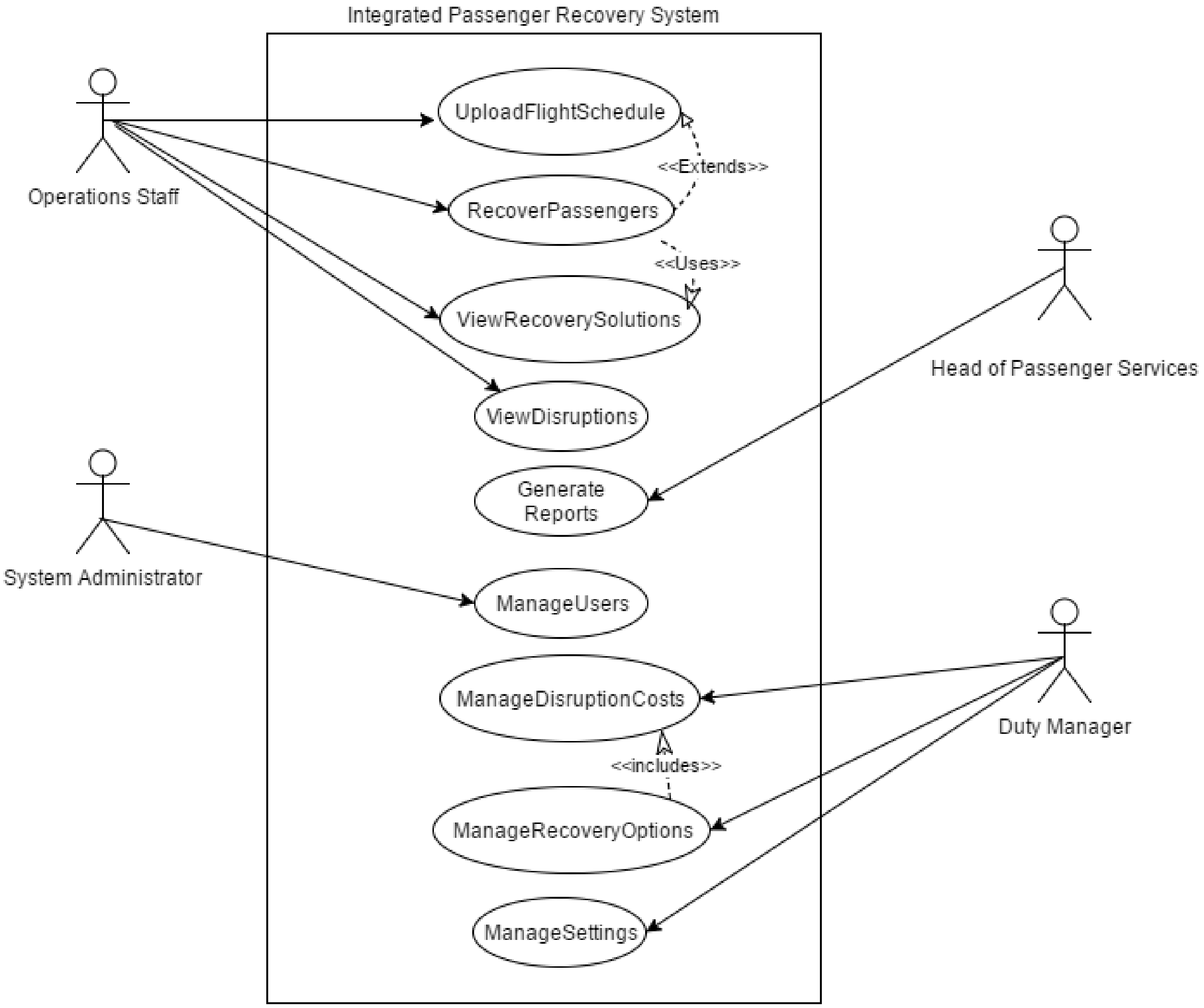


Figure 4. 3 Use Case Diagram

Use Case Descriptions

Table 4.1 shows the description the various use cases.

Table 4. 1 Use Case Descriptions

USE CASE	**DESCRIPTION**
UC1 – Upload Flight Schedule	**Primary actors:** Operations Staff. **Stakeholder:** **Precondition:** Is authenticated. Has a csv file with flight schedule data. **Post condition:** New flight schedule will be created. **Success:** Schedule uploaded successfully.
UC2 – Recover Passengers	**Primary actors:** Staff **Stakeholder:** IOCC Department. **Precondition:** A disruption has occurred. **Post condition:** Recovery solutions will be displayed on the screen. Affected passengers will be notified. **Success:** Disruption solved successfully.
UC3 – View Recovery Solutions	**Primary actors:** Staff **Stakeholder:** IOCC Department **Precondition:** User is authenticated.

	A successful recovery has been done. **Post condition:** User clicks on a disruption. **Success:** Available solutions are displayed on the screen.
UC4 – View Disruptions	**Primary actors:** Staff **Stakeholder:** IOCC Department **Precondition:** User is authenticated. **Post condition:** **Success:** Previous disruptions are displayed on the screen.
UC5 – Manage Users	**Primary actors:** System Administrator **Stakeholder:** ICT Department **Precondition:** User is authenticated. User has administrative permissions. **Post condition:** **Success:** User is able to add, edit, search and delete user accounts.
UC6 – Manage Disruption Costs	**Primary actors:** Duty Manager **Stakeholder:** IOCC Department

	Precondition: User is authenticated. User has managerial permissions. **Post condition:** **Success:** User is able to add, edit, search and delete disruption costs.
UC7 – Manage Recovery Options	**Primary actors:** Duty Manager **Stakeholder:** IOCC Department **Precondition:** User is authenticated. **Post condition:** **Success:** User is able to add, edit, search and delete recovery options.
UC8 – Manage Settings	**Primary actors:** Staff **Stakeholder:** IOCC Department **Precondition:** User is authenticated. **Post condition:** **Success:** User is able to add, edit, search and delete parameter settings.
UC9 – Generate Reports	**Primary actors:** Head of Passenger Services **Stakeholder:** Passenger Services Department

	Precondition: User is authenticated. **Post condition:** **Success:** User is able to generate reports.

4.5.2 System Sequence Diagram

This diagram shows the sequential interaction between users and the various components of the system according to (Dennis et al., 2012). It shows the request and response messages that users exchange with the system. Figure 4.4 shows the sequence diagram of how different users interact with the system.

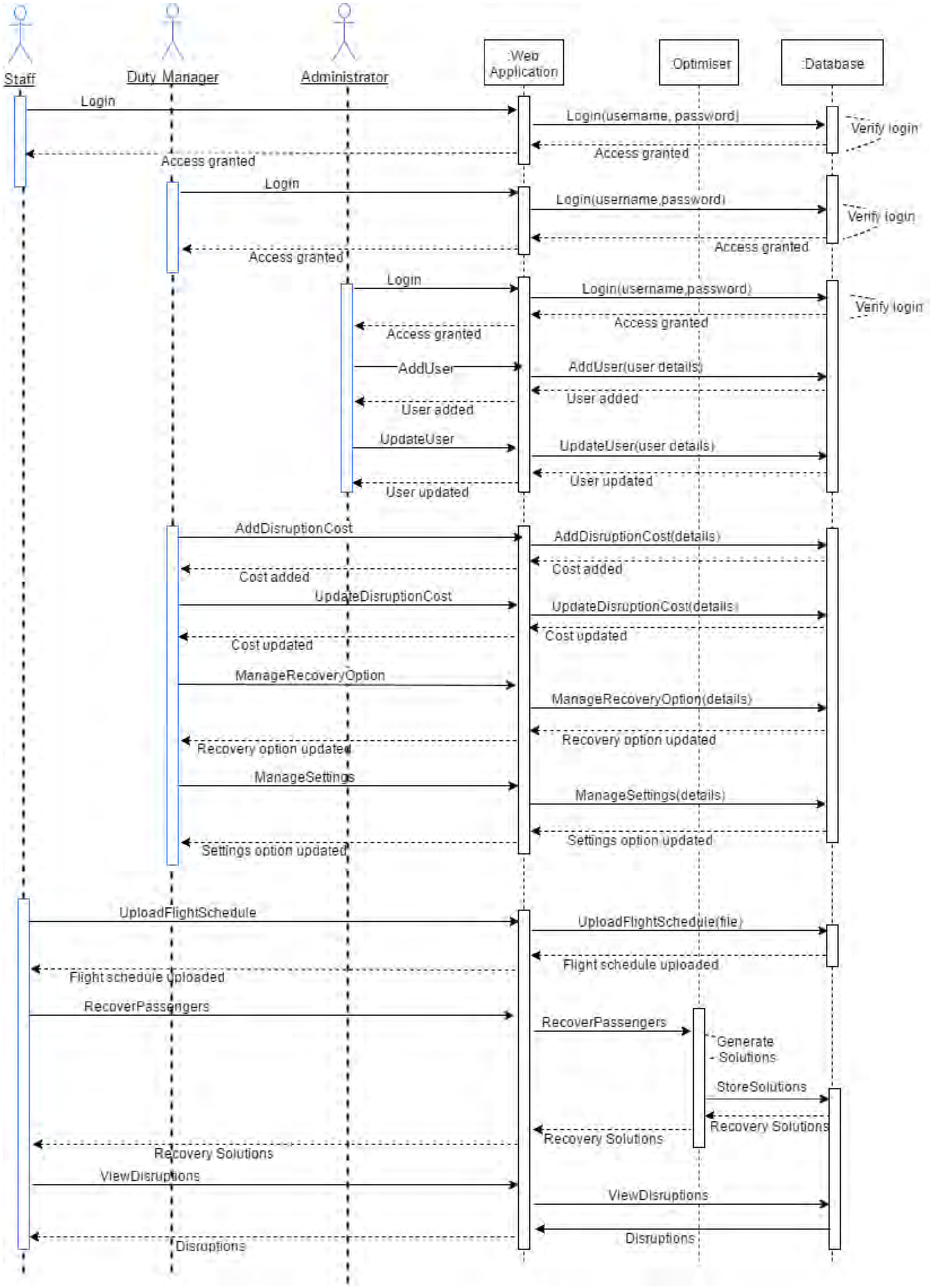


Figure 4. 4 System Sequence Diagram

4.5.3 Context Diagram

A Context Diagram is a single picture that has the system of interest at the center, with no details of its interior structure or function, surrounded by those elements in its environment with which it interacts (Le Vie & Donald, 2000). It defines the boundary of the system of interest and its interactions with the critical elements in its environment. Figure 4.5 shows the context diagram.

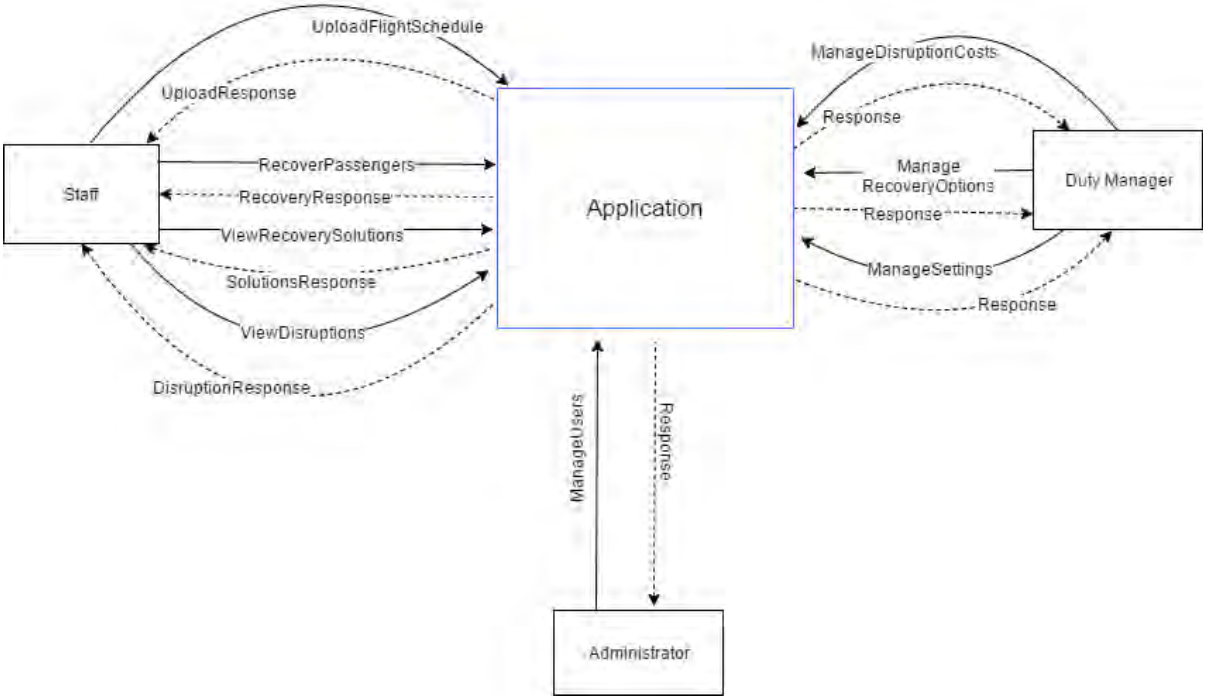


Figure 4. 5 Context Diagram

4.5.5 Database Schema

a) Entity Relationship Diagram

Entity Relationship Diagram is a data modelling tool that helps in the organisation of data into entities with relationships defined between them (Dennis, Wixom & Tegarden, 2012). The entities have attributes associated to them. Figure 4.6 illustrates how different entities are related in the database.

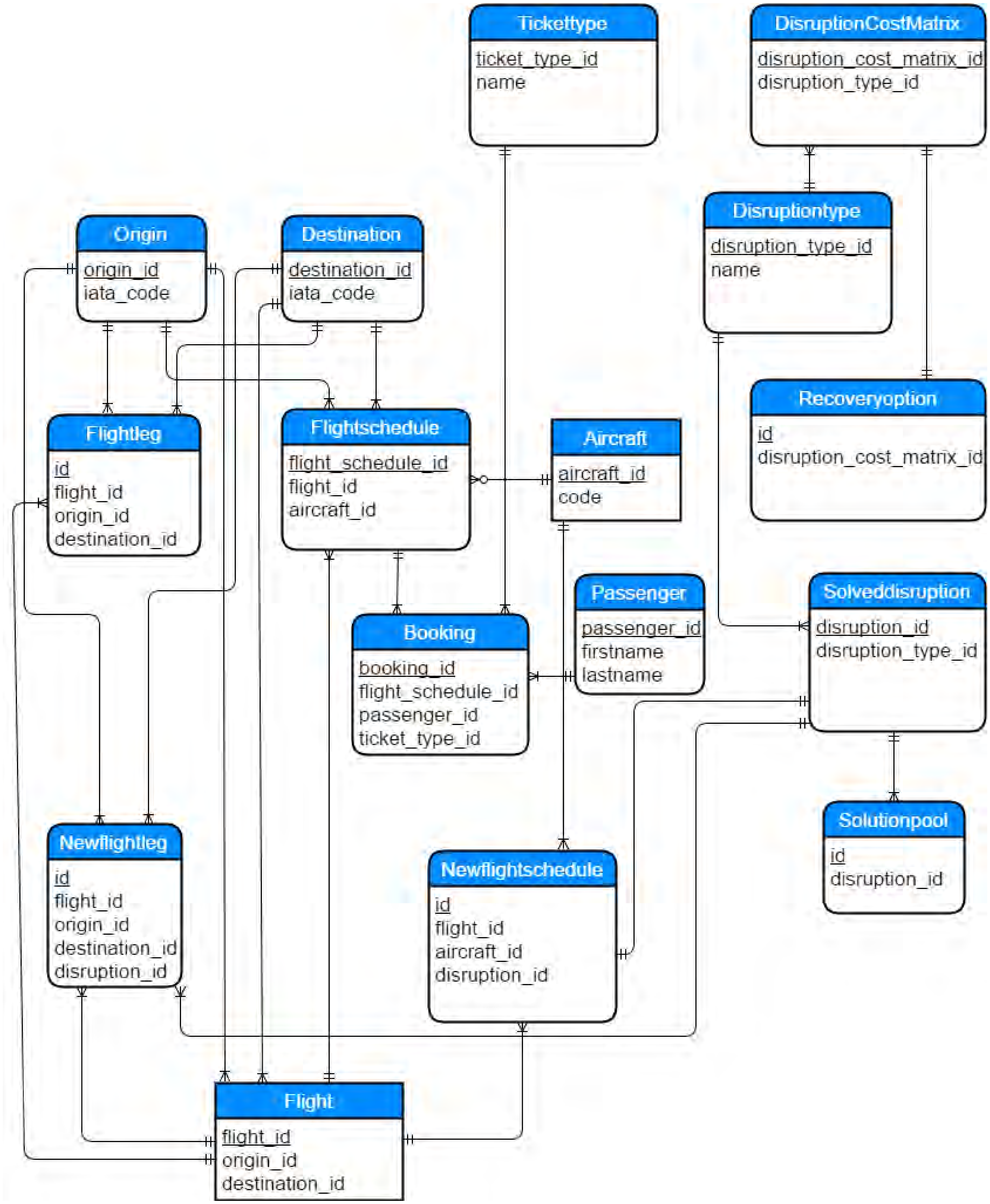


Figure 4. 6 Entity Relationship Diagram

The database tables are explained in the following sections. The attributes, datatypes, indices and a short description are given.

i. Setting

This is the table that holds the general parameters that apply to all recovery sections, that is: crew, aircraft and passenger recovery. Table 4.2 shows the settings relation.

Table 4. 2 Setting Table

Field	**Datatype**	**Index**	**Description**
id	INT(11)	PRIMARY KEY	This is a unique identifier.
temporary_passport_cost	DOUBLE		Temporary passports are issued if hotel accommodation is given.
business_class_multiplier	INT(11)		A factor to determine the business class ticket cost.
maximum_delay_minutes	INT(11)		Maximum flight delay time before passengers can be compensated.

ii. NewFlightLeg

This table stores the new flight legs generated by CPLEX after optimisation. The description is shown in table 4.3.

Table 4. 3 New Flight Leg

Field	**Datatype**	**Index**	**Description**
id	INT(11)	PRIMARY KEY	A unique identifier.
date_created	DATETIME		Date the record was created.
destination_id	INT(11)	FOREIGN KEY	Flight destination.
origin_id	INT(11)	FOREIGN KEY	Flight origin.
flight_id	INT(11)	FOREIGN KEY	Flight unique identifier.
disruption_id	INT(11)	FOREIGN KEY	Disruption unique identifier.

iii. Aircraft

This table store data about available aircraft. This data is used as input for the optimisation model. Table 4.4 shows the description of aircraft table.

Table 4. 4 Aircraft Table

Field	**Datatype**	**Index**	**Description**
aircraft_id	INT(11)	PRIMARY KEY	Unique identifier.
code	VARCHAR(20)		A unique code to identify the aircraft.
description	VARCHAR(50)		The actual aircraft name and model.
economyclass	INT(11)		Capacity of economy class.
businessclass	INT(11)		Capacity of business class.
ticketrefund	INT(11)		Refundable amount per ticket.
date_created	DATETIME		Date the record was created.
minimum_crew	INT(11)		Minimum crew required to operate a flight on this aircraft.

iv. SolvedDisruption

This is the table used to store the objective values of CPLEX solutions based on the disruption type. The description is as shown in table 4.5.

Table 4. 5 Solved Disruption

Field	**Datatype**	**Index**	**Description**
disruption_id	INT(11)	PRIMARY KEY	Unique identifier.
average_cost	DOUBLE		The average cost of disruption.
disruption_type_id	INT(11)	FOREIGN KEY	The type of disruption.
date_created	DATETIME		Date the record was created.

v. Passenger

This table stores information about passengers. Table 4.6 shows its description.

Table 4. 6 Passenger

Field	**Datatype**	**Index**	**Description**
passenger_id	INT(11)	PRIMARY KEY	Unique identifier.
firstname	VARCHAR(30)		The first name of a passenger.
lastname	VARCHAR(30)		The last name of a passenger.
phone	VARCHAR(20)		Passenger mobile number.
gender	VARCHAR(10)		Passenger gender
date_created	DATETIME		Date the record was created.

vi. Destination

This table is used to store data about all destinations. The specific fields, datatypes and indices are shown in table 4.7.

Table 4. 7 Destinations

Field	**Datatype**	**Index**	**Description**
destination_id	INT(11)	PRIMARY KEY	A unique identifier
iata_code	VARCHAR(5)		A three-letter code designating each airport.
icao_code	VARCHAR(5)		A four-character alphanumeric *code* designating each airport.
fullname	VARCHAR(30)		Actual name of city or country.
date_created	DATETIME		Date the record was created.

vii. Origin

This table is used to store data about all origins. The specific fields, datatypes and indices are shown in table 4.8.

Table 4. 8 Origins

Field	**Datatype**	**Index**	**Description**
origin_id	INT(11)	PRIMARY KEY	A unique identifier
iata_code	VARCHAR(5)		A three-letter code designating each airport.
icao_code	VARCHAR(5)		A four-character alphanumeric *code* designating each airport.
fullname	VARCHAR(30)		Actual name of city or country.
date_created	DATETIME		Date the record was created.

viii. SolutionPool

There can be more than one solution produced by CPLEX. The solution pool generated by the optimiser is stored in this table. The specific fields, datatypes and indices are shown in table 4.9.

Table 4. 9 Solution Pool

Field	**Datatype**	**Index**	**Description**
id	INT(11)	PRIMARY KEY	A unique identifier of records.
objective_value	DOUBLE		The value of the objective function of the optimisation model (in this case the cost).
disruption_id	INT(11)	FOREIGN KEY	A unique identifier of the associated disruption.
date_created	DATETIME		Date the record was created.

ix. FlightLeg

This table stores the flight legs of the original flight schedule. Table 4.10 shows the description of this table.

Table 4. 10 Flight Leg

Field	Datatype	Index	Description
id	INT(11)	PRIMARY KEY	A unique identifier.
flight_id	INT(11)	FOREIGN KEY	Unique identifier of a specific flight.
origin_id	INT(11)	FOREIGN KEY	Unique identifier of an origin airport.
destination_id	INT(11)	FOREIGN KEY	Unique identifier of a destination airport.
date_created	DATETIME		Date the record was created.

x. RecoveryOption

Recovery options can be meals, hotel accommodation or even monetary compensation. These are stored in this table. The description of this relation is shown in table 4.11.

Table 4. 11 Recovery Option

Field	Datatype	Index	Description
id	INT(11)	PRIMARY KEY	A unique identifier.
name	VARCHAR(50)		Option name.
amount	DOUBLE		Amount per passenger.
disruption_cost_matrix_id	INT(11)	FOREIGN KEY	Compensation per passenger.
date_created	DATETIME		Date the record was created.

xi. Booking

This table keeps the booking records. The description is shown in table 4.12.

Table 4. 12 Booking

Field	**Datatype**	**Index**	**Description**
booking_id	INT(11)	PRIMARY KEY	Unique identifier.
flight_schedule_id	INT(11)	FOREIGN KEY	Unique identifier of a specific flight.
passenger_id	INT(11)	FOREIGN KEY	Unique identifier of a passenger.
ticket_type_id	INT(11)	FOREIGN KEY	Unique identifier of a ticket type.
booking_amount	DOUBLE		The cost of the booked flight.
date_created	DATETIME		Date the record was created.

xii. DisruptionType

Disruption types include: delay, cancellation, change of aircraft and airport unavailability among others. These are stored in this table. The description of the table is shown in table 4.13.

Table 4. 13 Disruption Type

Field	**Datatype**	**Index**	**Description**
disruption_type_id	INT(11)	PRIMARY KEY	A unique identifier
name	VARCHAR(40)		Name of the disruption type.
date_created	DATETIME		Date the record was created.

xiii. Flight

This table is used to store static flight information. The description is shown in table 4.14.

Table 4. 14 Flight

Field	Datatype	Index	Description
flight_id	INT(11)	PRIMARY KEY	A unique identifier
origin_id	INT(11)	FOREIGN KEY	Unique identifier of the origin airport.
destination_id	INT(11)	FOREIGN KEY	Unique identifier of the destination airport.
flight_number	VARCHAR(20)		A descriptive alphanumeric number to identify the flight.
date_created	DATETIME		Date the record was created

xiv. DisruptionCostMatrix

Disruptions have different costs depending on the amount of time. The cost mapping is stored in this table. Table 4.15 shows the description of this table.

Table 4. 14 Disruption Cost Matrix

Field	Datatype	Index	Description
disruption_cost_matrix_id	INT(11)	PRIMARY KEY	A unique identifier.
minutes	INT(11)		Duration in minutes.
amount	DOUBLE		Compensation cost per passenger for the specific duration.
disruption_type_id	INT(11)	FOREIGN KEY	The type of disruption such as delay or cancellation.
date_created	DATETIME		Date the record was created.

xv. FlightSchedule

This table is used to store the original flight schedule. The description is shown in table 4.16.

Table 4. 15 Flight Schedule

Field	Datatype	Index	Description
flight_schedule_id	INT(11)	PRIMARY KEY	A unique identifier.
flight_id	INT(11)	FOREIGN KEY	Unique identifier of a flight.
std	DATETIME		Scheduled departure time.
sta	DATETIME		Scheduled arrival time.
aircraft_id	INT(11)	FOREIGN KEY	Aircraft to operate the flight.
date_created	DATETIME		Date the record was created.

xvi. TicketType

This table stores the different ticket types. The description is shown in table 4.17.

Table 4. 16 Ticket Type

Field	Datatype	Index	Description
ticket_type_id	INT(11)	PRIMARY KEY	A unique identifier.
name	VARCHAR(30)		Name of the ticket type.
date_created	DATETIME		Date the record was created.

xvii. NewFlightSchedule

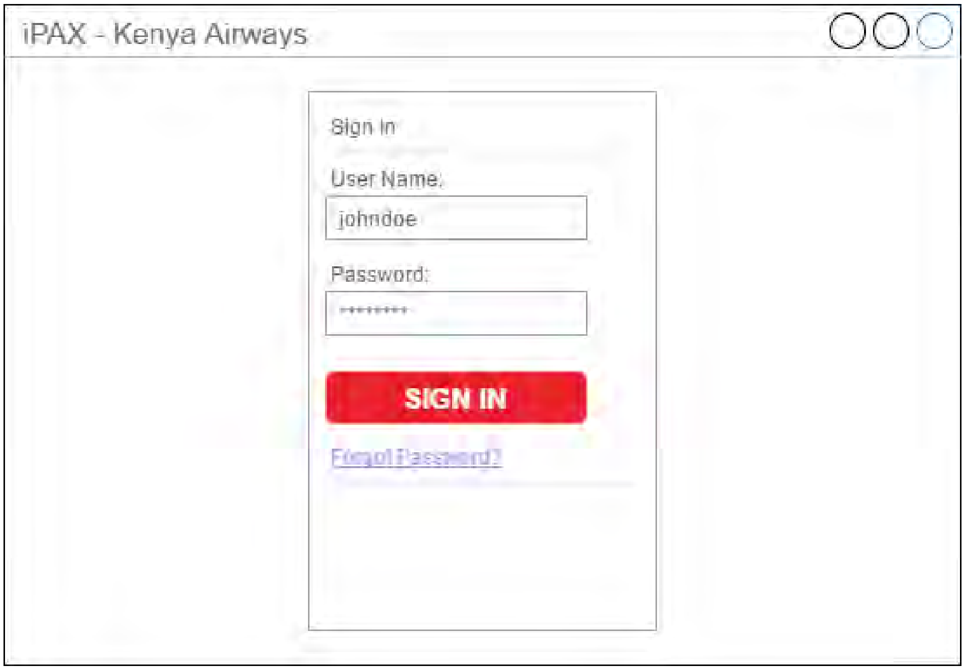
When a disruption occurs, a new flight schedule is generated which is stored in this table. The description is shown in table 4.18.

Table 4. 17 New Flight Schedule

Field	**Datatype**	**Index**	**Description**
id	INT(11)	PRIMARY KEY	A unique identifier.
flight_id	INT(11)	FOREIGN KEY	Unique identifier of a flight.
std	DATETIME		New scheduled departure time.
sta	DATETIME		New scheduled arrival time.
aircraft_id	INT(11)	FOREIGN KEY	New aircraft to operate the flight.
disruption_id	INT(11)	INT(11)	A unique identifier of the disruption.
date_created	DATETIME		Date the record was created.

4.5.6 Wireframes

This section presents the anticipated appearance of the final application. The user opens the web application and logins in. After successful login they are redirected to the dashboard page with a navigation menu to access other system features. If user is an administrator they are redirected to the administration portal.



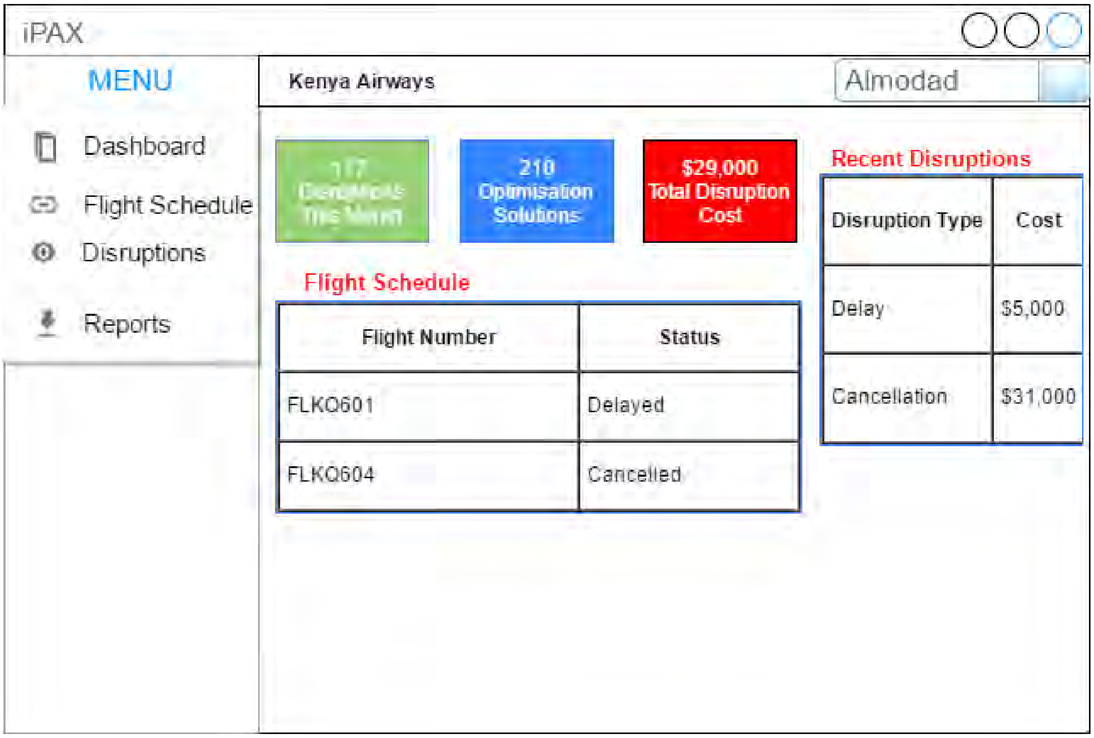
User Name: johndoe

Password: ********

Figure 4. 7 Login Screen

User Name: johndoe

Figure 4. 8 Reset Password



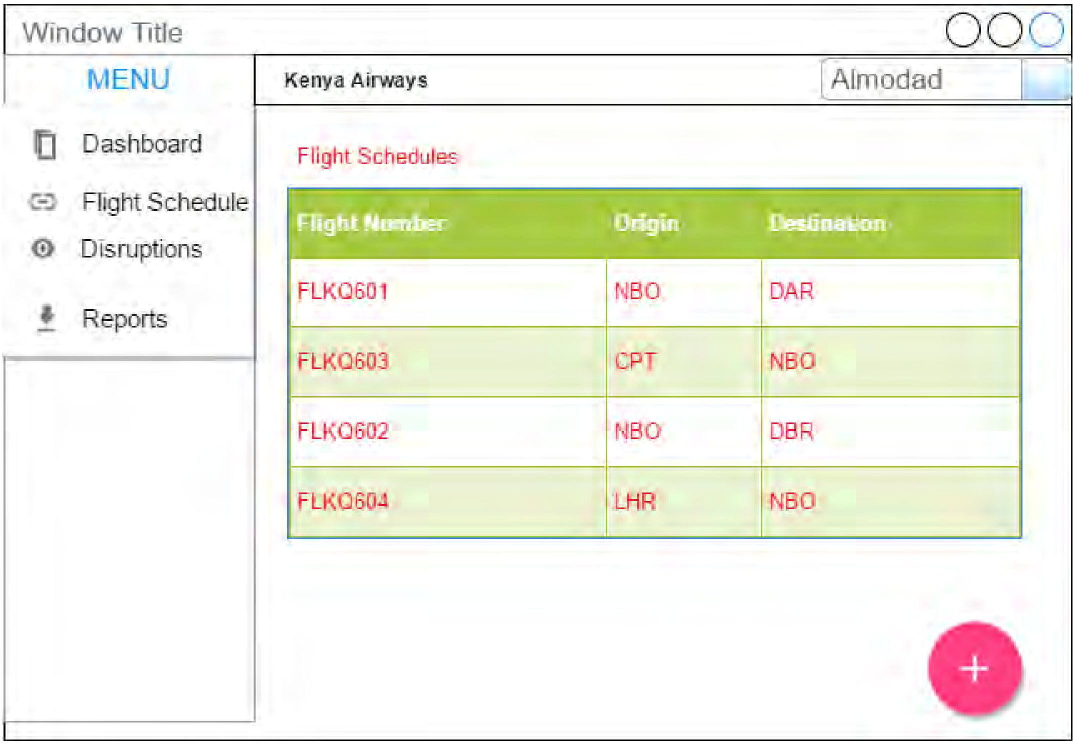
Flight Schedule

Flight Number	Status
FLKO601	Delayed
FLKO604	Cancelled

Recent Disruptions

Disruption Type	Cost
Delay	$5,000
Cancellation	$31,000

Figure 4. 9 Dashboard Screen



Flight Number	Origin	Destination
FLKQ601	NBO	DAR
FLKQ603	CPT	NBO
FLKQ602	NBO	DBR
FLKQ604	LHR	NBO

Figure 4. 10 Flight Schedule Screen

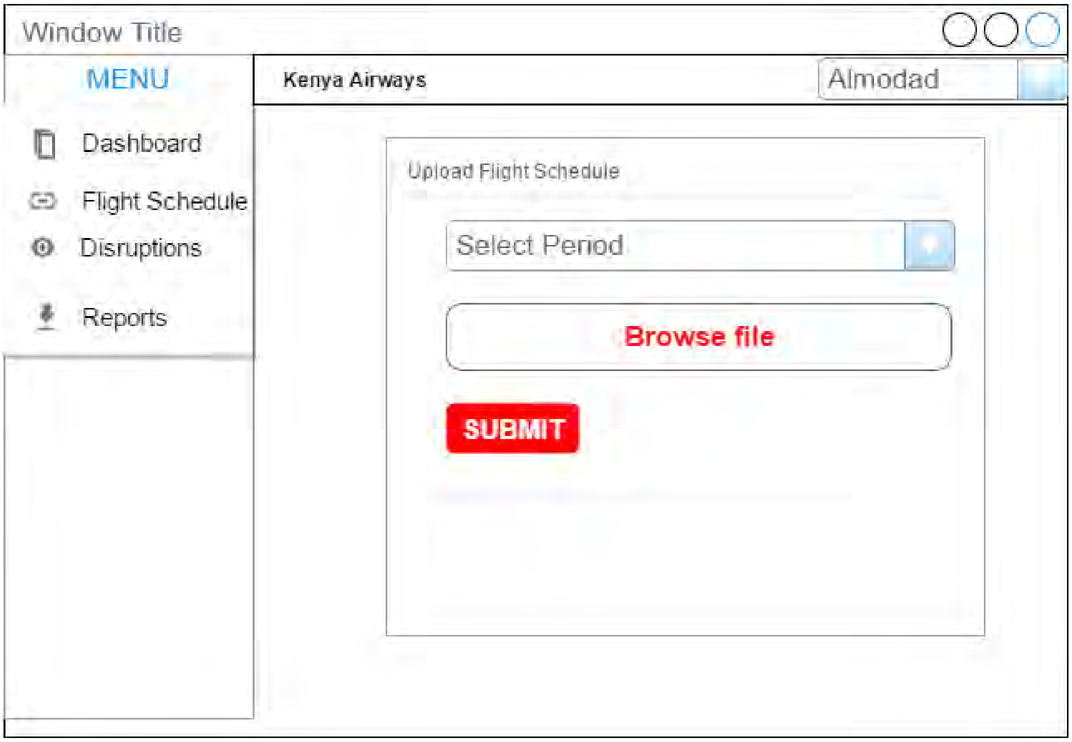
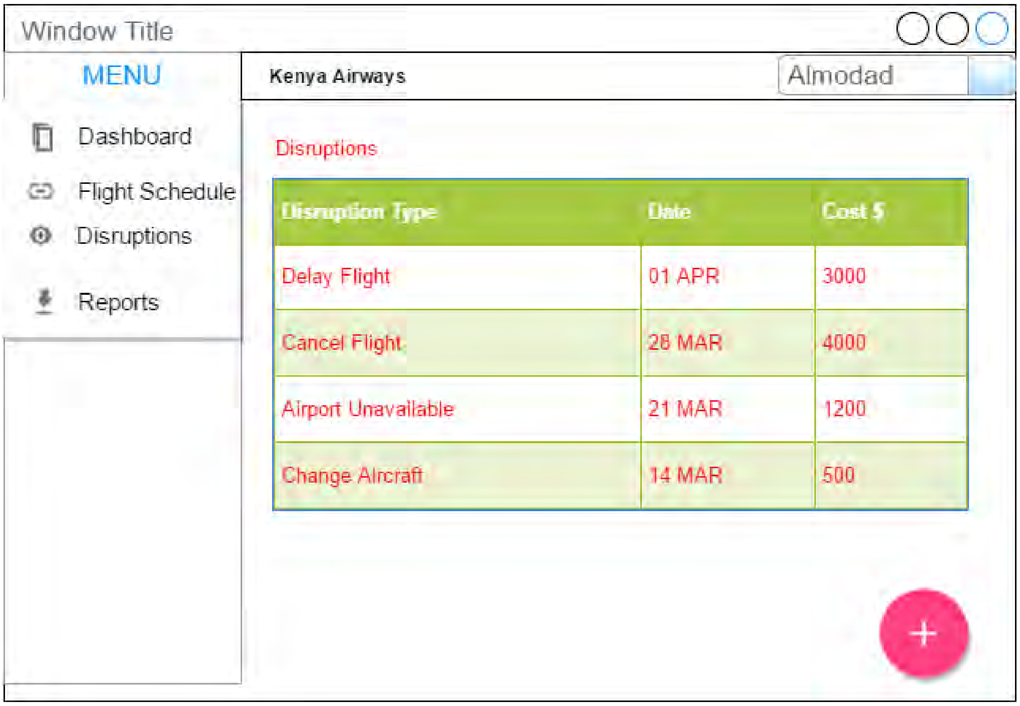


Figure 4. 11 Upload Flight Schedule



Disruptions

Disruption Type	Date	Cost $
Delay Flight	01 APR	3000
Cancel Flight	28 MAR	4000
Airport Unavailable	21 MAR	1200
Change Aircraft	14 MAR	500

Figure 4. 12 Disruptions Screen

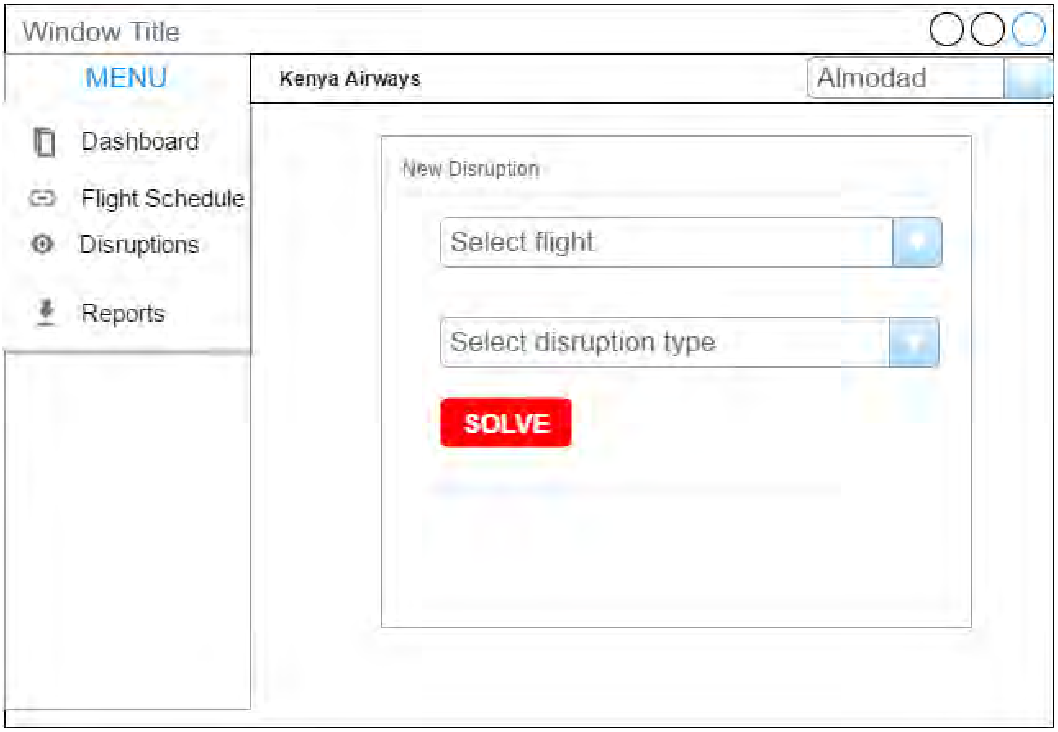
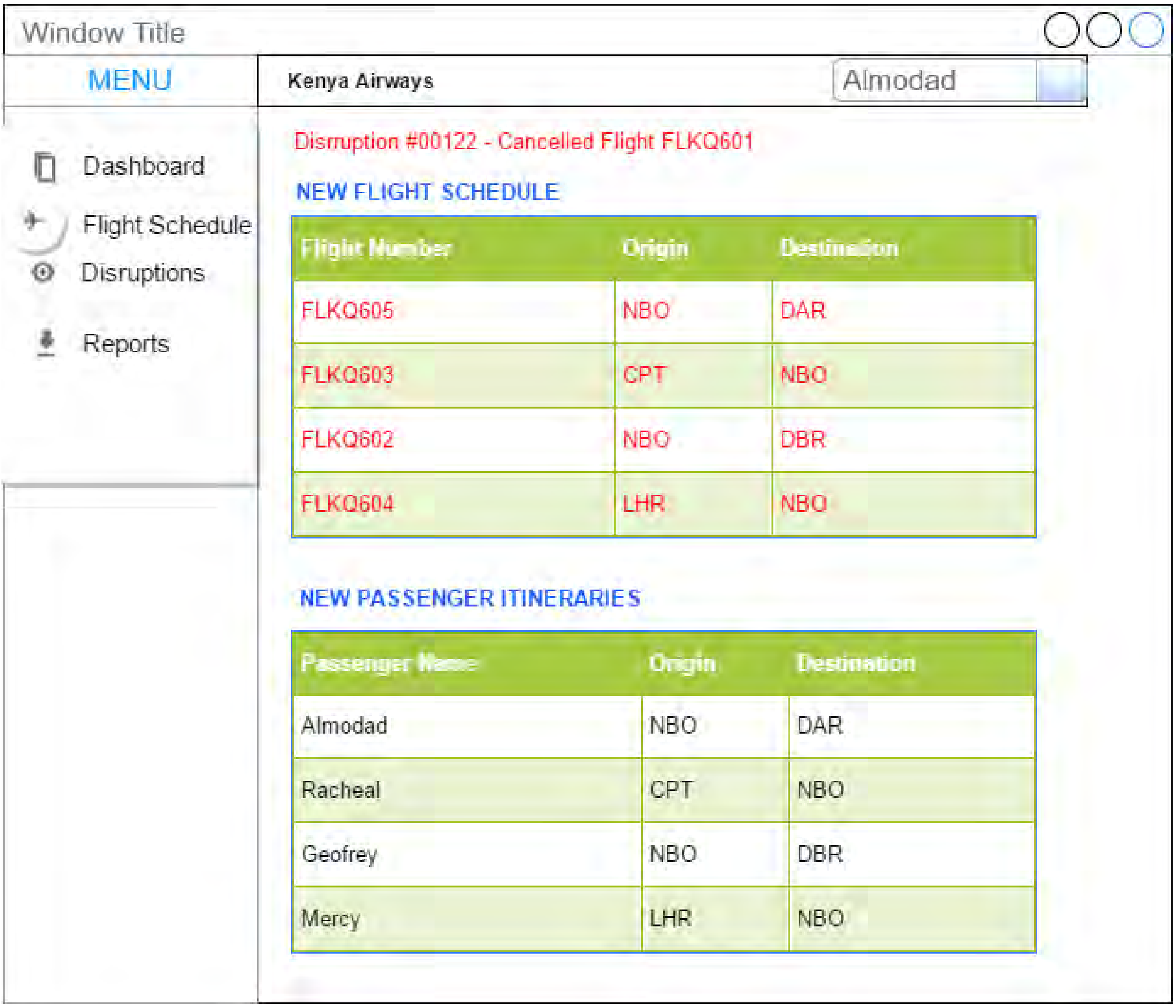


Figure 4. 13 Add Disruption Screen



Kenya Airways

Almodad

Disruption #00122 - Cancelled Flight FLKQ601

NEW FLIGHT SCHEDULE

Flight Number	Origin	Destination
FLKQ605	NBO	DAR
FLKQ603	CPT	NBO
FLKQ602	NBO	DBR
FLKQ604	LHR	NBO

NEW PASSENGER ITINERARIES

Passenger Name	Origin	Destination
Almodad	NBO	DAR
Racheal	CPT	NBO
Geofrey	NBO	DBR
Mercy	LHR	NBO

Figure 4. 14 Optimisation Results Screen

4.6 Summary

This chapter presented an analysis of the system requirements and highlighted the proposed system architecture. System design diagrams were drawn using UML to further illustrate the various system. Some of these diagrams include: *system sequence diagram, context diagram, use case diagram and entity relationship diagram.* The mathematical recovery model was also discussed in more detail where the decision variables, parameters, constraints and the objective function were explained.

CHAPTER FIVE: SYSTEM IMPLEMENTATION AND TESTING

5.1 Introduction

This chapter explains how the design presented in chapter four was implemented. Both client side and administrator side of the web portal are presented. The functional requirements are clearly shown in the design output presented with screenshots in this chapter. The procedure to carry out system testing is also given.

5.2 System Development

Scrum Agile Development methodology as discussed in chapter three was used to design and implement the proof of concept.

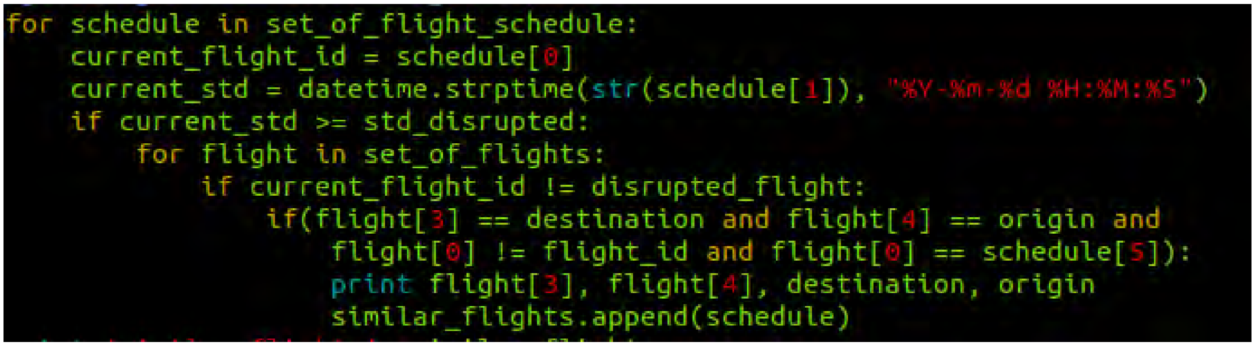
5.2.1 Optimisation model

The optimisation model, which is the core of this implantation was developed using the Python API for the CPLEX optimization studio. The output of the model is converted and stored in CSV files. An Observer Programming Pattern was implemented using Py-notify to monitor any new files generated by CPLEX. RabbitMQ message broker is used to queue the new CPLEX output file for processing and storage in the database. Different algorithms are used to perform several checks and pairings before invoking the optimiser.

Algorithm 1: Check Available Flights in the Disrupted Flight Route

---

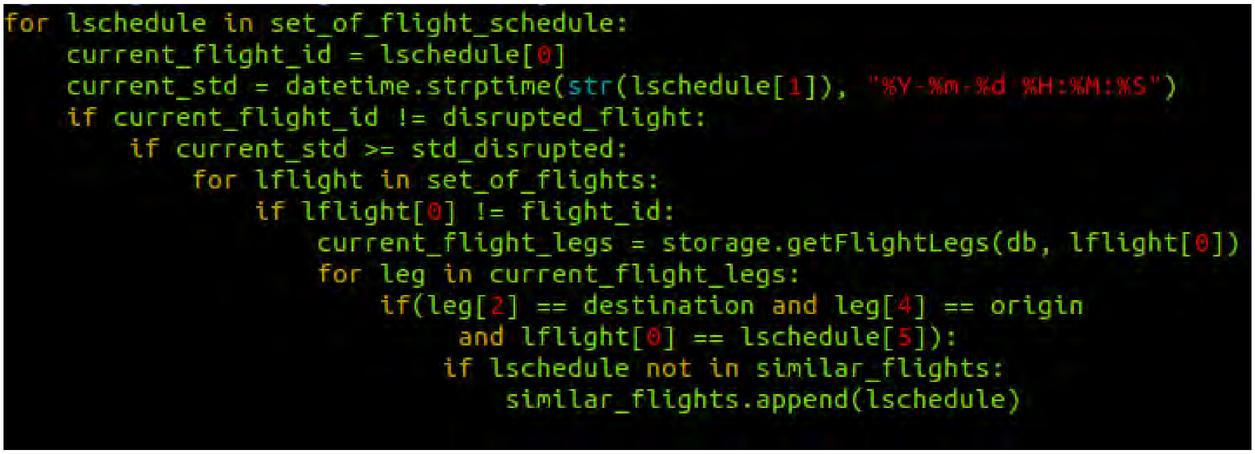
This algorithm checks for all upcoming flights in the flight schedule whose origin and destinations are similar to the disrupted flight and stores them in a list.



```
for schedule in set_of_flight_schedule:
    current_flight_id = schedule[0]
    current_std = datetime.strptime(str(schedule[1]), "%Y-%m-%d %H:%M:%S")
    if current_std >= std_disrupted:
        for flight in set_of_flights:
            if current_flight_id != disrupted_flight:
                if(flight[3] == destination and flight[4] == origin and
                    flight[0] != flight_id and flight[0] == schedule[5]):
                    print flight[3], flight[4], destination, origin
                    similar_flights.append(schedule)
```

Algorithm 2: Check Available Flights Legs Similar to the Disrupted Flight

This algorithm checks for flight legs in other flights. The flight legs must much the origin and destination of the disrupted flight and must be for flights coming after the disrupted flight.



```
for lschedule in set_of_flight_schedule:
    current_flight_id = lschedule[0]
    current_std = datetime.strptime(str(lschedule[1]), "%Y-%m-%d %H:%M:%S")
    if current_flight_id != disrupted_flight:
        if current_std >= std_disrupted:
            for lflight in set_of_flights:
                if lflight[0] != flight_id:
                    current_flight_legs = storage.getFlightLegs(db, lflight[0])
                    for leg in current_flight_legs:
                        if(leg[2] == destination and leg[4] == origin
                             and lflight[0] == lschedule[5]):
                            if lschedule not in similar_flights:
                                similar_flights.append(lschedule)
```

Algorithm 3: Generate Passenger-Flight Pairings

This algorithm generates pairings of passengers to flights based on seat availability and passenger type.

```
if total_business-business > 0 and len(business_passengers) > 0:
    #assign business seats
    if total_business-business >= business_pax:
        for passenger in pax_set:
            for bpax in business_passengers:
                if passenger[0] == bpax:
                    reassigned_business_passengers[passenger[0]] = sflight
                    business_passengers.remove(bpax)
    else:
        length = total_business-business
        for passenger in pax_set:
            for bpax in business_passengers:
                if passenger[0] == bpax:
                    if length > 0:
                        reassigned_business_passengers[passenger[0]] = sflight
                        business_passengers.remove(bpax)
                        length -= 1
else:
    print 'No remaining business class seats'
if total_economy-economy > 0 and len(economy_passengers) > 0:
    #assign economy seats
    if total_economy-economy >= economy_pax:
        for passenger in pax_set:
            for epax in economy_passengers:
                print 'pax', passenger[0], 'epax', epax
                if passenger[0] == epax:
                    reassigned_economy_passengers[passenger[0]] = sflight
                    economy_passengers.remove(epax)
    else:
        length = total_economy-economy
        for passenger in pax_set:
            for epax in economy_passengers:
                print 'pax', passenger[0], 'epax', epax
                if passenger[0] == epax:
                    if length > 0:
                        reassigned_economy_passengers[passenger[0]] = sflight
                        economy_passengers.remove(epax)
                        length -= 1
```

Algorithm 4: Passenger Notifications

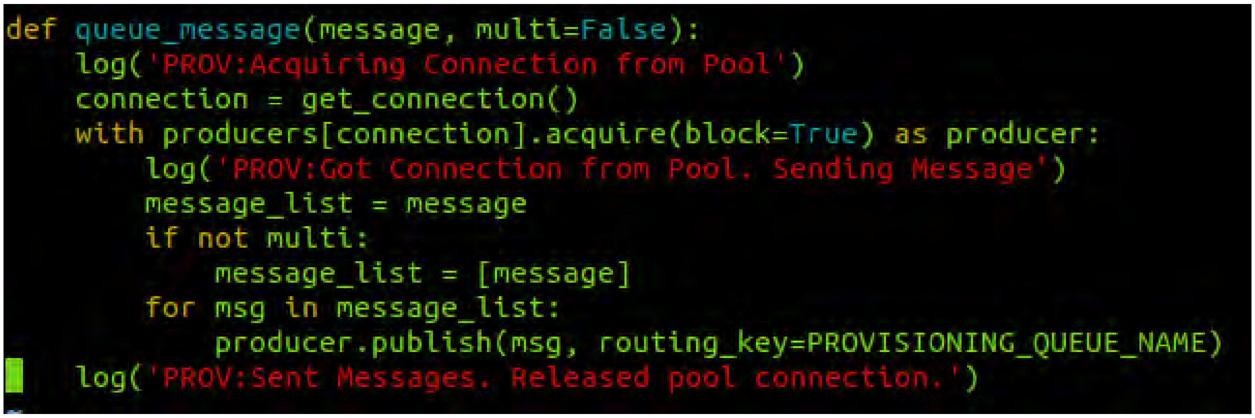
This algorithm notifies passengers affected by a given disruption. It received a disruption identifier from the queue consumer and retrieves the notifications associated with that disruption from a database.

```
def notify_passenger(record):
    """Send SMS to passengers
    """
    status = False
    #response = None
    disruption_id = int(record[0])
    LOG.info('processing disruption_id|{}'.format(disruption_id))
    try:
        s = storage.Storage()

        notifications = s.getNotifications(config.Config().getConnection(), disruption_id)
        for note in notifications:
            message = note[1]
            paxid = note[5]
            paxdetails = s.getPassengerDetails(config.Config().getConnection(), paxid)
            phone = paxdetails[0][3]
            sms.sendsms(phone, message)
            print message
    except Exception, e:
        print str(e)
        response = 'error'
        status = False
        print response
        return (status, response)
    else:
        response = 'SUCCESS'
        status = True
        return (status, response)
```

Algorithm 5: Message Queueing

This algorithm is used to queue files containing the disruption information. The files are handled as messages and transported using the AMQP.



```
def queue_message(message, multi=False):
    log('PROV:Acquiring Connection from Pool')
    connection = get_connection()
    with producers[connection].acquire(block=True) as producer:
        log('PROV:Got Connection from Pool. Sending Message')
        message_list = message
        if not multi:
            message_list = [message]
        for msg in message_list:
            producer.publish(msg, routing_key=PROVISIONING_QUEUE_NAME)
    log('PROV:Sent Messages. Released pool connection.')
```

5.2.2 Web Application

The web application was implemented as a proof of concept. Python Django framework was used to develop the web application. MySQL database was used to store data generated by the optimisation model as solutions to disruption problems. The parameters of the optimisation model are also stored in the database.

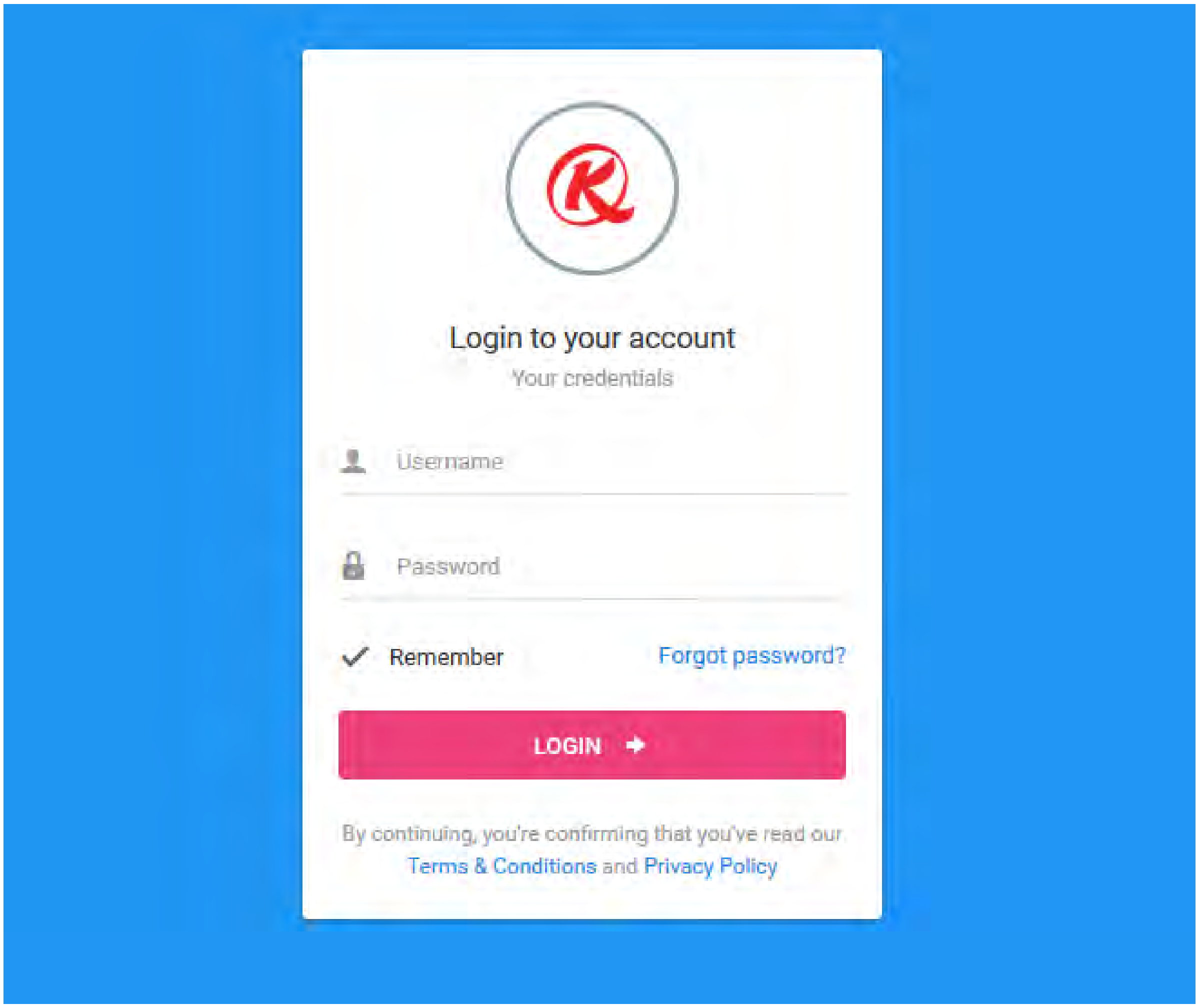
5.3 System Implementation

The web application makes invokes the optimisation model and displays data stored in the database. There are two major user types: normal users and administrators.

The major system components of the web application are:

5.3.1 Login

Users of the web application are required to login using a username and password. The username and password are authenticated and based on their validity, access is either granted or denied. Upon successful login, the user is redirected to the appropriate page based on their user group thus preventing unauthorised access. Figure 5.1 shows the login page.



Login to your account

Your credentials

Username

Password

✓ Remember

Forgot password?

LOGIN

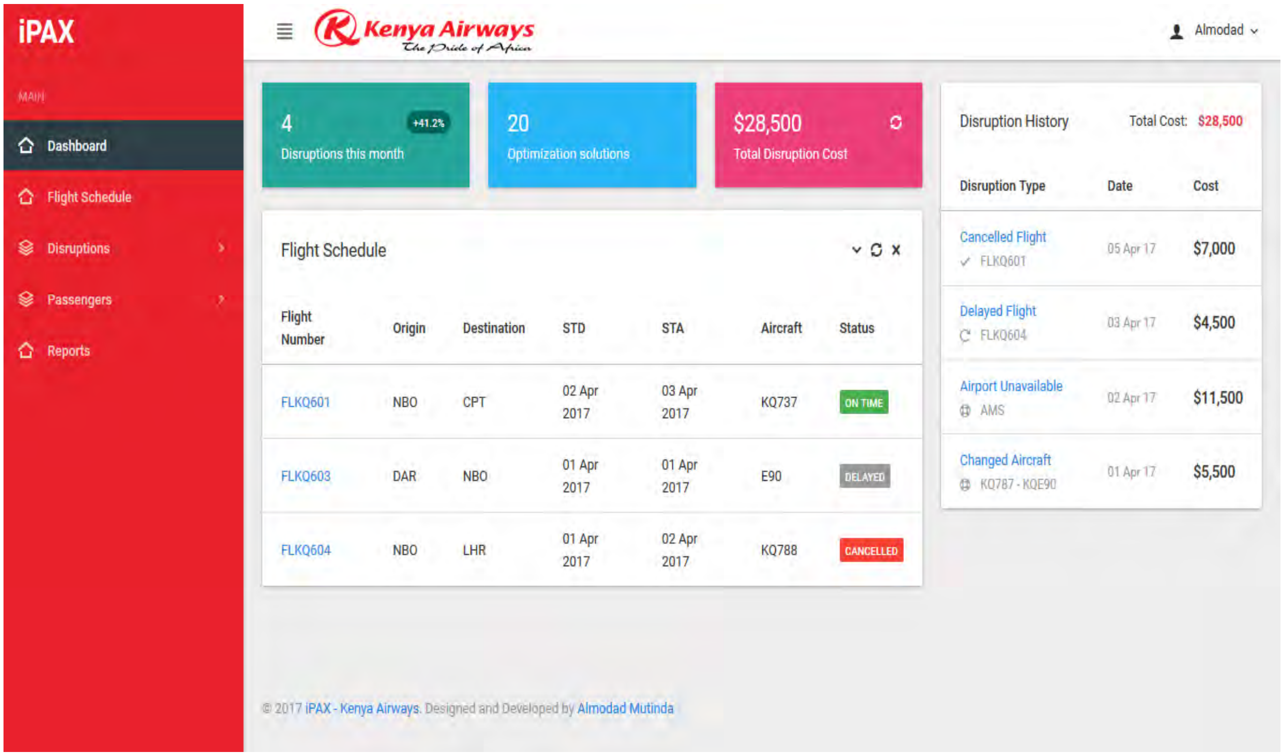
By continuing, you're confirming that you've read our Terms & Conditions and Privacy Policy

Figure 5. 1 Login Page

All user login attempts are recorded to be able to detect suspicious attempts and enforce the necessary password policies.

5.3.2 User Home Page

This is the page where users are redirected after a successful login. It shows a dashboard with a few summaries, the flight schedules and recent disruptions. Figure 5.2 shows the home page.



4 +41.2%
Disruptions this month

20
Optimization solutions

$28,500
Total Disruption Cost

Flight Schedule

Flight Number	Origin	Destination	STD	STA	Aircraft	Status
FLKQ601	NBO	CPT	02 Apr 2017	03 Apr 2017	KQ737	ON TIME
FLKQ603	DAR	NBO	01 Apr 2017	01 Apr 2017	E90	DELAYED
FLKQ604	NBO	LHR	01 Apr 2017	02 Apr 2017	KQ788	CANCELLED

Disruption History Total Cost: $28,500

Disruption Type	Date	Cost
Cancelled Flight ✓ FLKQ601	05 Apr 17	$7,000
Delayed Flight FLKQ604	03 Apr 17	$4,500
Airport Unavailable AMS	02 Apr 17	$11,500
Changed Aircraft KQ787 - KQE90	01 Apr 17	$5,500

© 2017 iPAX - Kenya Airways. Designed and Developed by Almodad Mutinda

Figure 5. 2 User Home Page

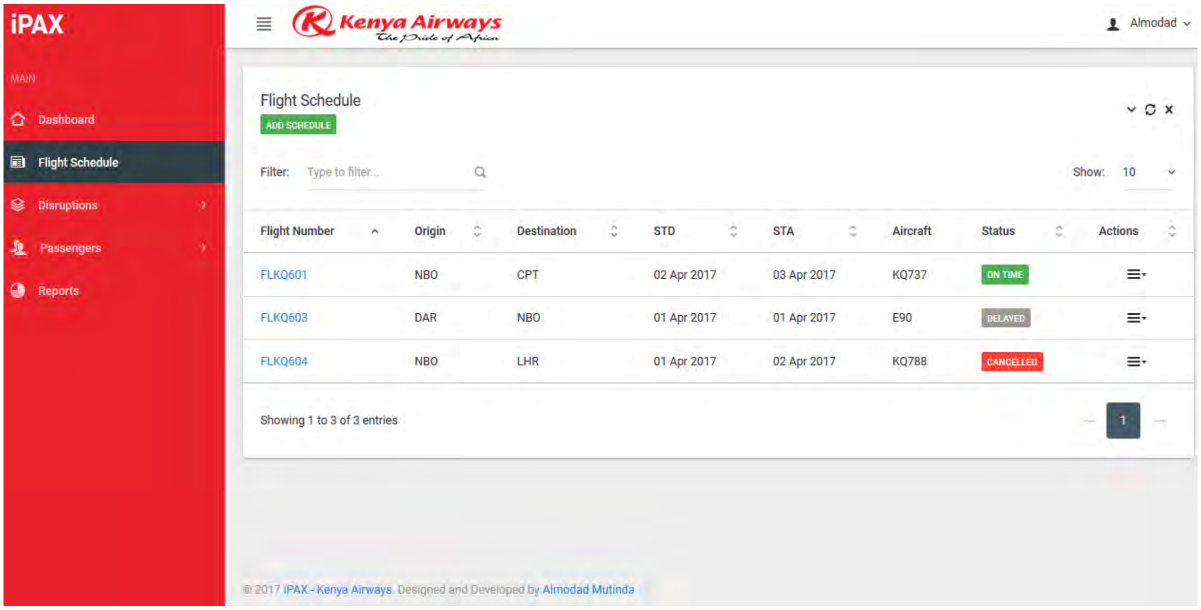
The flight number in the flight schedule table is a clickable link that shows more details about the specific flight. The details include flight legs and passenger information.

5.3.2 Flight Schedule Page

A flight schedule a list of flights that is used to plan and allocate work appropriately among airline crew. The key fields of the schedule are:

i. Flight number
ii. Origin airport
iii. Destination airport
iv. Scheduled departure time
v. Scheduled arrival time
vi. Aircraft

This page presents a detailed table of all available flights. The table has searching and sorting features. Pagination is also incorporated to avoid too much scrolling. A link is provided to add a new flight schedule. Figures 5.3 shows the flight schedule.



Flight Schedule

Filter: Type to filter...

Show: 10

Flight Number	Origin	Destination	STD	STA	Aircraft	Status	Actions
FLKQ601	NBO	CPT	02 Apr 2017	03 Apr 2017	KQ737	ON TIME	≡
FLKQ603	DAR	NBO	01 Apr 2017	01 Apr 2017	E90	DELAYED	≡
FLKQ604	NBO	LHR	01 Apr 2017	02 Apr 2017	KQ788	CANCELLED	≡

Showing 1 to 3 of 3 entries

© 2017 iPAX - Kenya Airways. Designed and Developed by Almodad Mutinda

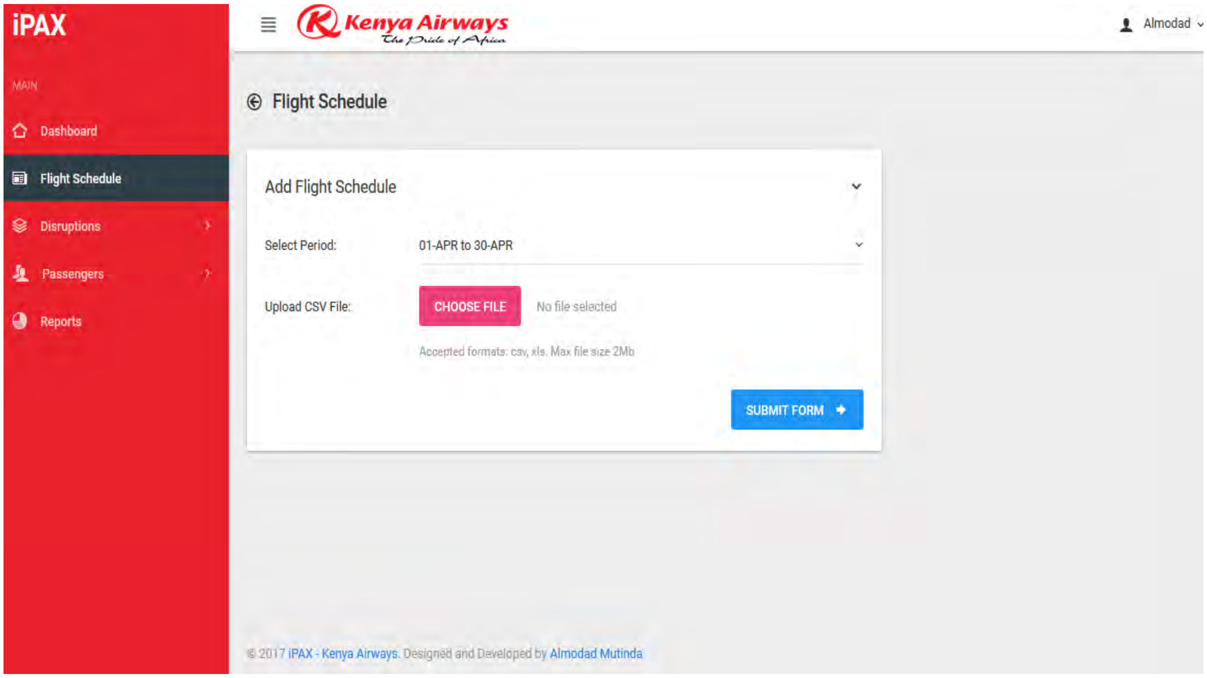
Figure 5. 3 Flight Schedule Page

5.3.3 Add Flight Schedule Page

The flight schedule is as described in section 5.3.2. However, the airline has to add the original schedule to be used as input when there is a disruption. To add the schedule, two fields should be filled. These are:

i. Schedule period
ii. A CSV file containing the flight information

The schedule information is generated based on the passengers' flight booking records. Figure 5.4 shows the form for adding a flight schedule.



Add Flight Schedule

Select Period: 01-APR to 30-APR

Upload CSV File: No file selected

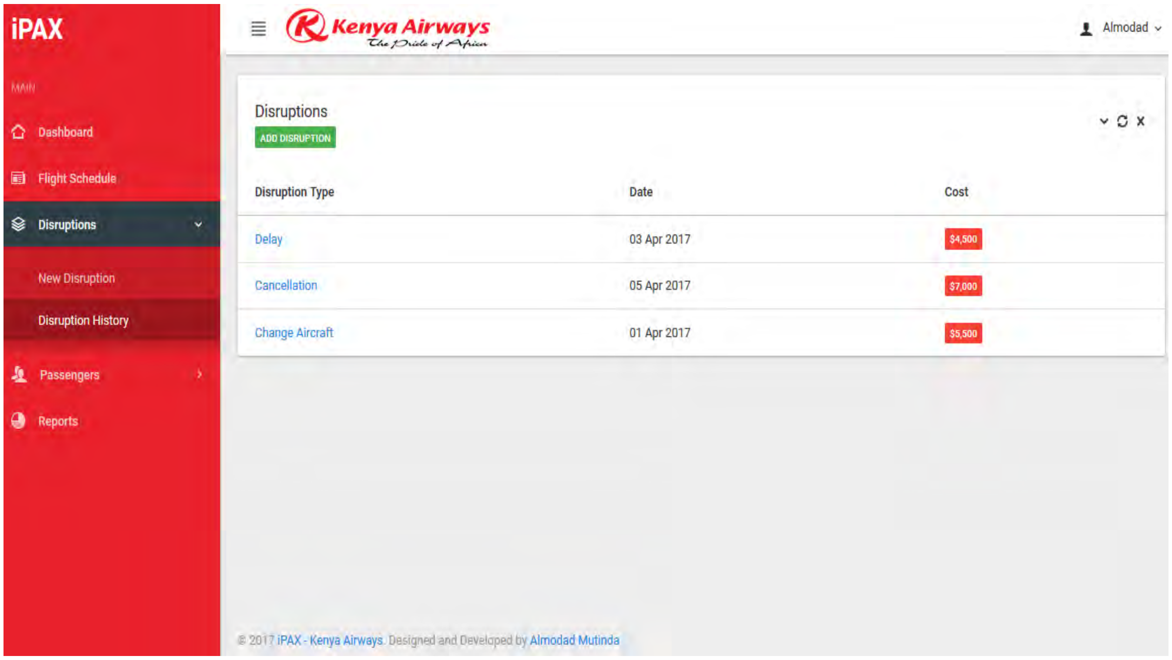
Accepted formats: csv, xls. Max file size 2Mb

© 2017 iPAX - Kenya Airways. Designed and Developed by Almodad Mutinda

Figure 5. 4 Add Flight Schedule

5.3.4 Disruptions Page

A disruption is any activity that negatively affects the planned schedule thus making it infeasible. Disruptions occur every day and airlines suffer huge loses of money and reputation due to these disruptions. Figure 5.5 shows a list of previous disruptions and provides a link to add a new disruption. On clicking the disruption type, one is able to view more details about it.



Disruptions

Disruption Type	Date	Cost
Delay	03 Apr 2017	$4,500
Cancellation	05 Apr 2017	$7,000
Change Aircraft	01 Apr 2017	$5,500

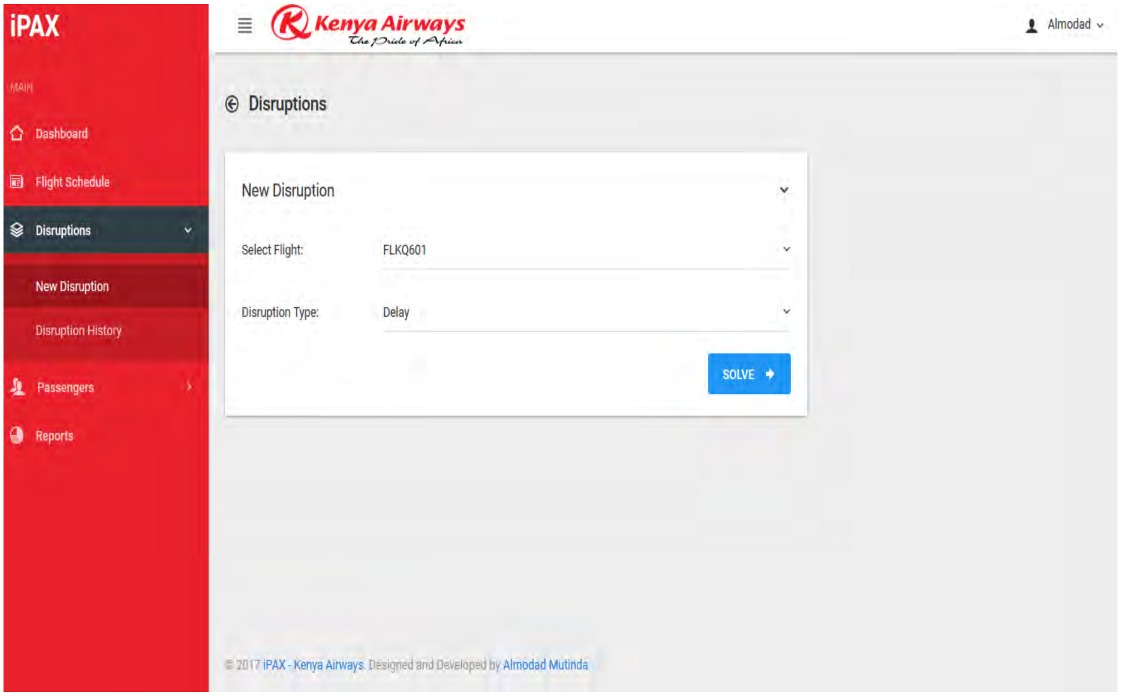
© 2017 iPAX - Kenya Airways. Designed and Developed by Almodad Mutinda

Figure 5. 5 Disruption page

5.3.5 Add Disruption Page

When a disruption occurs, it need to be solved quickly and optimally to recover the original schedule. An optimiser implemented with CPLEX is used to quickly solve disruptions given the necessary parameters. The component in figure 5.6 is used to record a new disruption when it occurs and invoke the optimiser. This form has two mandatory fields which are:

i. The disrupted flight
ii. The disruption type



Select Flight: FLKQ601

Disruption Type: Delay

© 2017 iPAX - Kenya Airways. Designed and Developed by Almodad Mutinda

Figure 5. 6 Add Disruption

5.3.6 Administration Portal

The parameters used to define input to the optimiser needs to be very dynamic. This is so because many disruptions are unpredictable and they require different parameter sets to solve. The administration component of the web application is primarily used for configuring the model parameters and user management. The following are some of the functionalities that can be achieved through the portal:

i. Manage Recovery Options
ii. Manage Parameter Settings
iii. Manage Disruption Types
iv. Manage Disruption Costs
v. Manage Flight Legs
vi. User Management
vii. Manage Aircraft (for testing)
viii. Manage Destinations (for testing)

Figure 5.7 shows the administration portal home page.

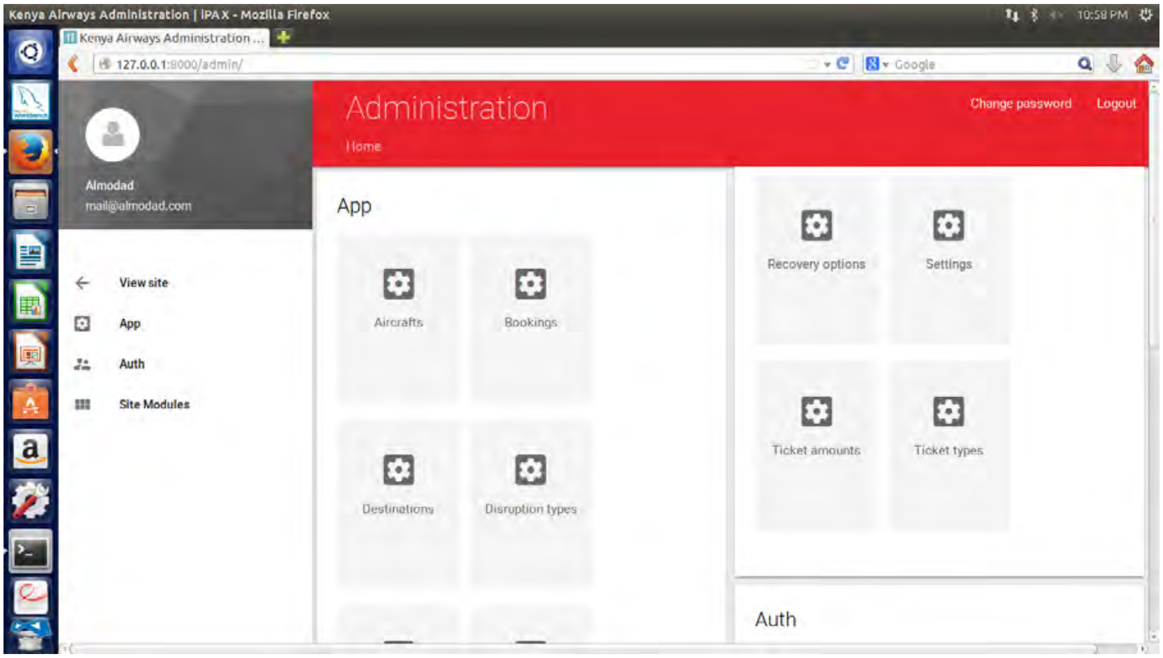


Figure 5. 7 Administration Portal Home Page

Disruptions can be of many types. These can be easily configured through the administration portal as depicted in figure 5.8.

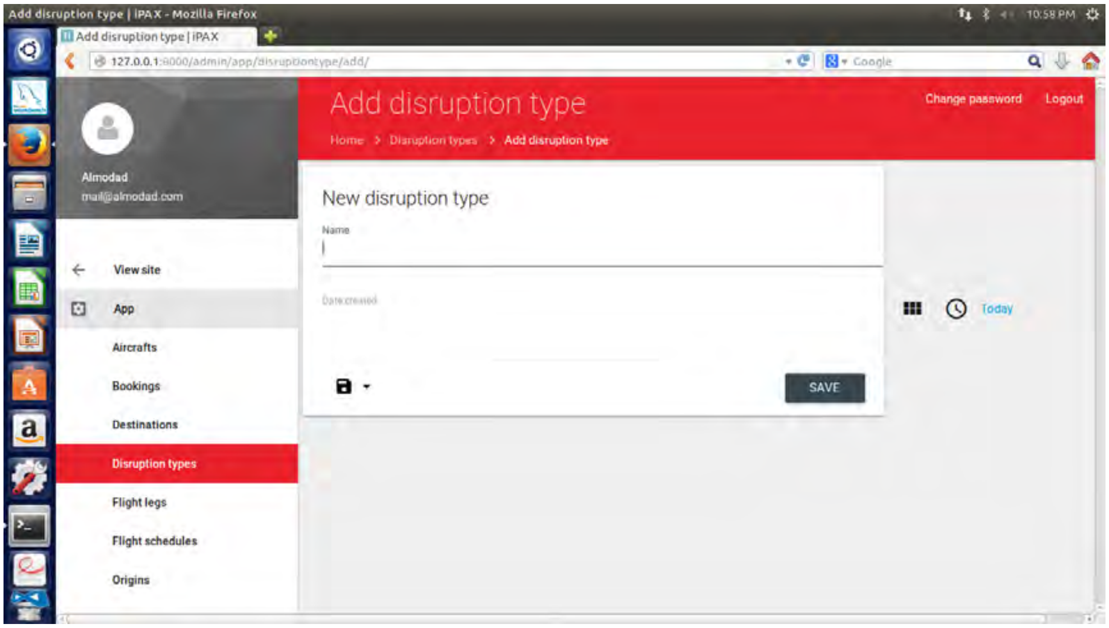
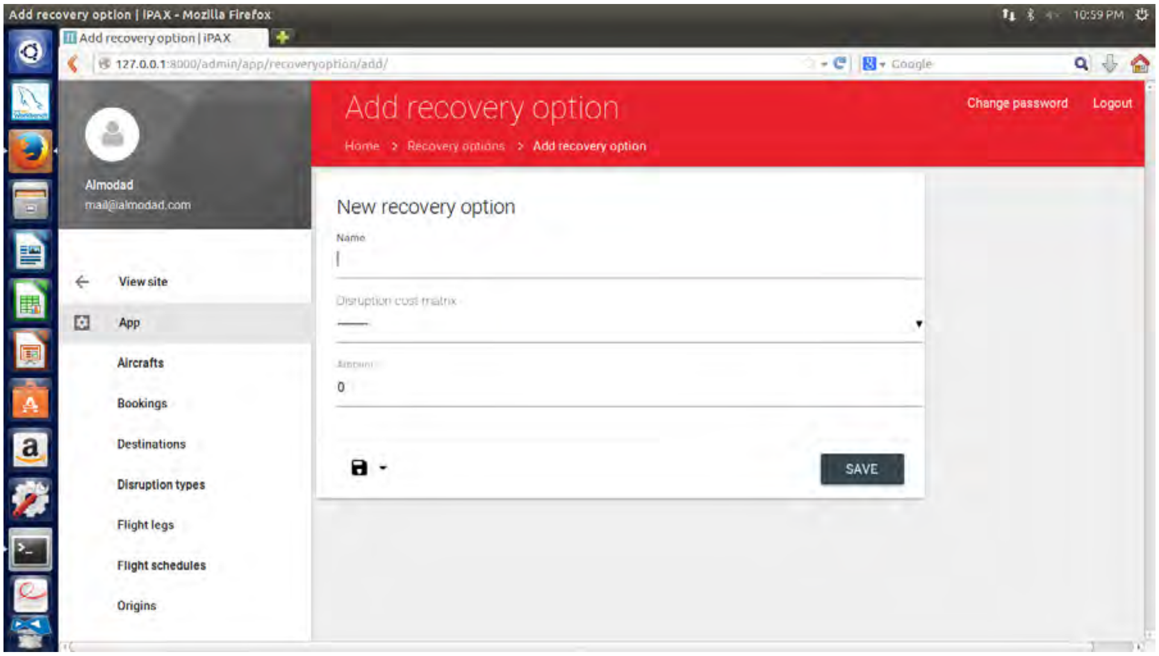


Figure 5. 8 Disruption Types

When a disruption occurs, there can be more than one recovery option. These options have different cost matrices associated to them and can greatly impact airline revenues. The cost represents the cost per disrupted passenger who was assigned that recovery option. These options can be configured in the administration portal as shown in figure 5.9.



New recovery option

Name

Disruption cost matrix: ———

Amount: 0

Figure 5. 9 Recovery Options

A flight leg is the operation of an aircraft from one scheduled departure station to its next scheduled arrival station. Each flight has at least one flight leg. This means that a flight can have one or many flight legs. Flight legs are used as inputs in the optimiser to be able to generate new optimal flight schedules. Figure 5.10 shows the form used to add a flight leg.

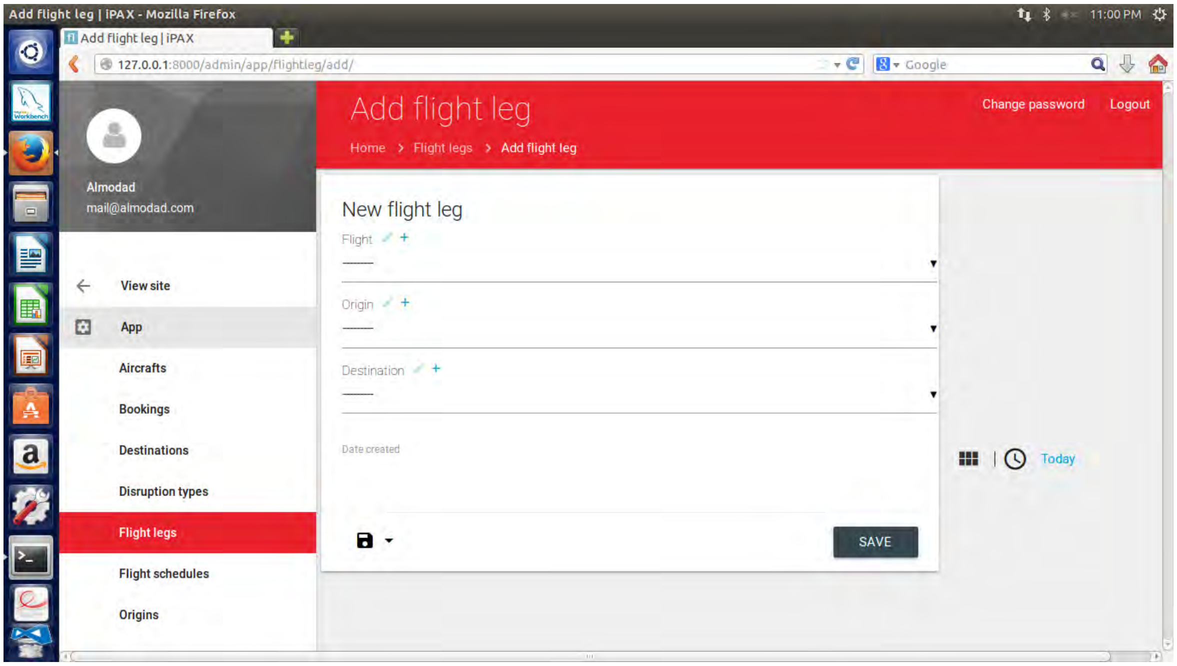
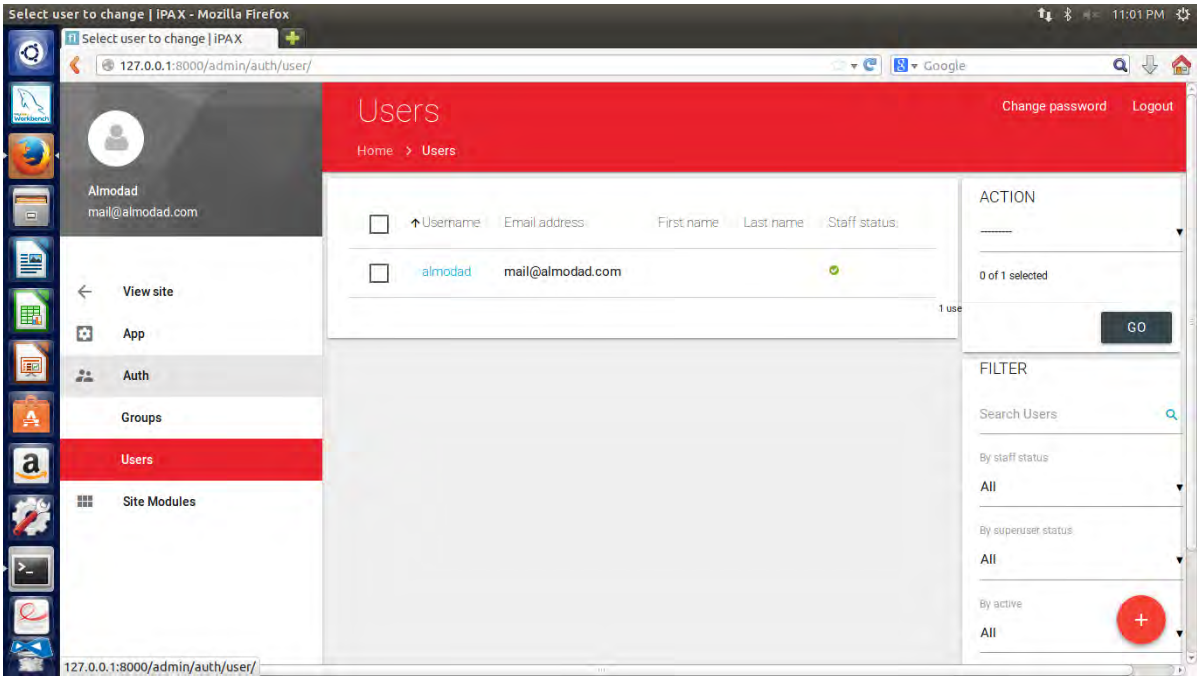


Figure 5. 10 Flight Legs

User management enables the system administrator to create, view, update and delete user in the system. This component also provides a simpler way of managing user permissions to control access to system resources. Figure 5.11 shows the user management page.



☐	Username	Email address	First name	Last name	Staff status
☐	almodad	mail@almodad.com			✓

Figure 5. 11 User Management

5.4 System Testing

This section outlines the tests carried out on the model and the web application. Four types of tests were done; compatibility testing, load testing, integration testing and user testing which included functionality testing and acceptance testing.

5.4.1 Compatibility Testing

Compatibility testing was mainly done for web browsers. This was in order to ensure that the system was compatible with most common web browsers. Table 5.1 shows results of the web browser compatibility test.

Table 5. 1 Web Browser Compatibility Testing

Web Browser	Compatibility
Mozilla Firefox 10.0 and up	**compatible**
Google Chrome	**compatible**
Internet Explorer 7 and up	**compatible**
Opera Mini	**Compatible**
Microsoft Edge	**compatible**

5.4.2 Load Testing

This testing was done on the optimisation model by running it and measuring the time taken to solve disruptions given different sets of parameters and constraints. Load testing was done to measure how long the model takes to produce a feasible solution. This was a very key consideration because the output of the model was used to make decisions in a real world environment. Queueing was tested to measure the significance of number of workers on the processing time.

5.4.3 Integration Testing

System Integration Testing (SIT) was done after all system modules were integrated. This testing was done to detect any bugs as a result of integration before user testing could be done. Interoperability between the different modules was tested to confirm that they worked as expected while maintaining high cohesion.

5.4.4 Functional Testing

Functional testing was done against the functional and non-functional requirements of the system to determine whether the system was functionally acceptable. The major use cases were tested to find out the success or failure of the specific system components. Results expected for different scenarios were set and then the actual results identified to either successful or unsuccessful. Table 5.2 shows the upload flight schedule test case.

Table 5. 2 Upload Flight Schedule Test Case

Test Case ID	1
Test Case	Upload Flight Schedule
Description	User selects the schedule period and then selects a CSV file containing the flight schedule and submits the form
Utilized use case	UploadFlightSchedule
Results	Schedule uploaded successfully
Pass/Fail	Pass

The test case for recovering passengers is shown in table 5.3.

Table 5. 3 Recover Passengers Test Case

Test Case ID	2
Test Case	Recover Passengers
Description	User specifies the disruption type and the disrupted flight.
Utilized use case	RecoverPassengers
Results	Passengers reallocated successfully.
Pass/Fail	Pass

In table 5.4, the test case for viewing recovery solutions is provided.

Table 5. 4 View Recovery Solutions Test Case

Test Case ID	3
Test Case	View Recovery Solutions
Description	User clicks on Recovery Options button.
Utilized use case	ViewRecoverySolutions
Results	Recovery solutions displayed on the user's screen.
Pass/Fail	Pass

Disruptions are shown in table 5.5.

Table 5. 5 View Disruptions Test Case

Test Case ID	4
Test Case	View Disruptions
Description	User clicks on the Disruptions button.
Utilized use case	ViewDisruptions
Results	Previous disruptions displayed on the user's screen.
Pass/Fail	Pass

Table 5.6 shows the user management test case.

Table 5. 6 Manage Users Test Case

Test Case ID	5
Test Case	Manage Users
Description	User clicks on the Users menu option.
Utilized use case	ManageUsers
Results	Registered users displayed on the user's screen.
Pass/Fail	Pass

The disruption costs can be managed as shown in table 5.7.

Table 5. 7 Manage Disruption Costs Test Case

Test Case ID	6
Test Case	Manage Disruption Costs
Description	User clicks on the Disruption Costs menu option.
Utilized use case	ManageDisruptionCosts
Results	User able to add, edit and delete disruption costs
Pass/Fail	Pass

Table 5.8 shows the results of manage recovery options test case.

Table 5. 8 Manage Recovery Options Test Case

Test Case ID	7
Test Case	Manage Recovery Options
Description	User clicks on the Recovery Options menu option.
Utilized use case	ManageRecoveryOptions
Results	User able to add, edit and delete recovery option.
Pass/Fail	Pass

Each parameter used in the model is configured in settings page. Table 5.9 shows the test case for this.

Table 5. 9 Manage Settings Test Case

Test Case ID	8
Test Case	Manage Settings
Description	User clicks on the Settings menu option.
Utilized use case	ManageSettings
Results	User able to add, edit and delete parameter settings.
Pass/Fail	Pass

Table 5. 10 Generate Reports Test Case

Test Case ID	9
Test Case	Generate Reports
Description	User clicks on the Reports menu.
Utilized use case	GenerateReports
Results	User able to generate reports.
Pass/Fail	Pass

5.4.5 User Testing

System users were involved in testing through oral interviews.

i. User Acceptance Testing (UAT)

The User Acceptance Testing (UAT) was done to ascertain that the system performed as per user expectations. Eight respondents were involved in UAT. The interview guide is provided in Appendix C. User responses are presented using charts. Figure 5.12 shows the results of UAT.

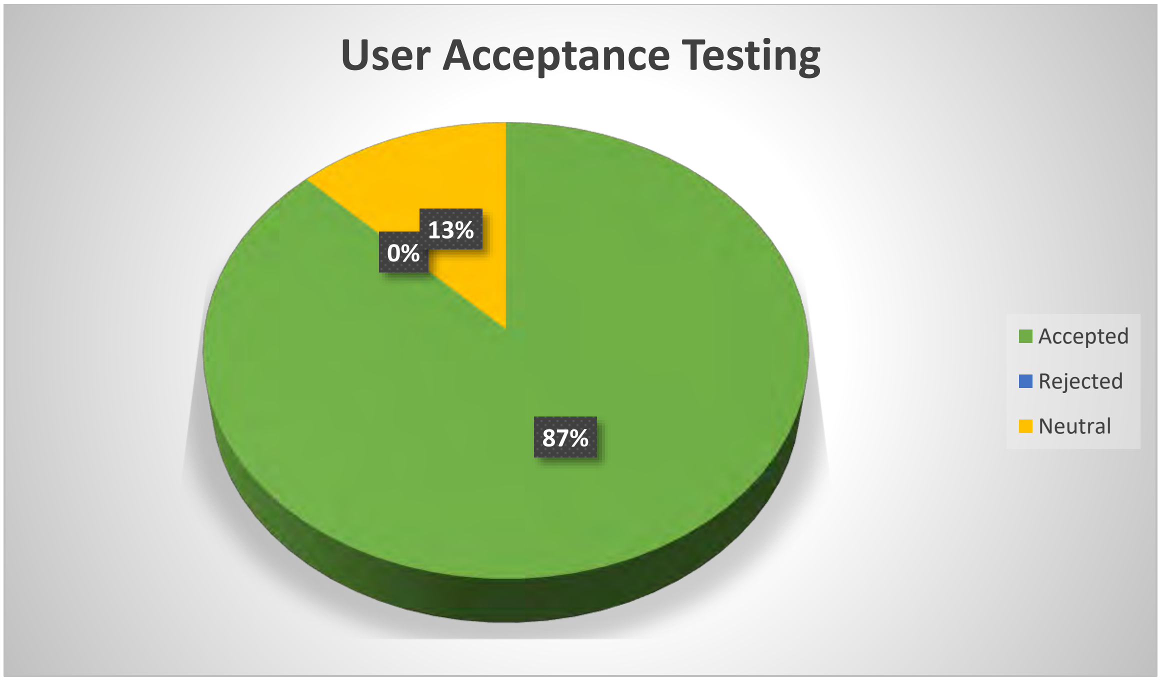


Figure 5. 12 User Acceptance Testing

87% of the respondents accepted the system while only 13% remained neutral. None of them rejected the system.

5.5 Summary

System implementation was done where an optimisation model and a web application were developed based on the functional and non-functional requirements formulated in chapter 4. The various design concepts were considered during implementation. The objectives of this research were met and research questions answered by developing and testing the system. Functional testing helped to test the system functionalities while UAT was done to validate whether the system was useful to the target users.

CHAPTER SIX: DISCUSSION OF RESULTS

6.1 Introduction

The purpose of this research was to come up with an optimisation model for passenger recovery in airline operations. This chapter aims at finding out if the research objectives have been met and compare the developed system with the current system at KQ to identify its advantages and the unique improvements it has brought.

6.2 Discussions

This research was conducted within Kenya Airways with the aim of identifying the problems faced during disruptions and develop a technical solution to those problems. Preliminary discussions with the director of operations showed that there was a problem with passenger recovery whenever a disruption occurs. This was echoed by the head of passenger services, the head of security and the staff at the IOCC department at Kenya Airways.

Discussion in Relation to Research Objectives

The research objectives in section 1.4 were achieved as discussed in this section. The first research objective in Section 1.4 was to identify the current methods of passenger recovery and their challenges. A discussion with Kenya Airways IOCC staff in charge of passenger recovery revealed that they manually reallocate passengers whenever a disruption occurs. This process takes a lot of time which leads to high costs for compensating disrupted passengers. An analysis of the current system was done using the guide in Appendix D. It was found out that the current passenger recovery methods are largely manual.

The second research objective was to review existing solutions for passenger recovery. A review of literature as explained in section 2.3 shows that very little research has been done on passenger recovery with many researchers concentrating on aircraft and crew recovery. Section 2.4 and 2.5 looks into the existing passenger recovery models in the airline industry in order to identify the gaps. It was found out that most solutions do not provide a way of notifying passengers of the disruption decisions affecting their itineraries. This gap has been solved by implementing automated notifications whenever a decision has been made. Section 2.6 reviews the costs as a result of disruptions. This objective forms the basis for this research.

The third objective was to design, develop and test a system for passenger recovery. Currently there is no system for passenger recovery. During disruptions, the operation controllers manually check through the passenger bookings and the flight schedule to find available slots and allocate disrupted passengers. This objective was achieved in chapter four and five. The system architecture is shown in section 4.3 while the optimisation model is explained in section 4.4. This includes the mathematical formulation and the model architecture. Design was done based on the user requirements and section 4.4 shows the various designs. Development or implementation was achieved using various technologies. Implementation output is presented in section 5.3. System testing was done in section 5.4 to ensure that the system worked as expected. UAT was done where 87% of the users accepted the system while only 13% remained neutral and none of them rejected the system.

The fourth and last research objective was to implement and validate the system. System implementation was done in section 5.3 where the various components of the system were explained. Implementation of the model was done using CPLEX optimisation studio. A web application was developed to provide an interface for users to interact with the model. The web application was developed using Django web framework. Validation of the system was done to test the model computation time and reliability in providing feasible solutions. This was done with the head of OCC and the head of passenger services at Kenya Airways.

6.3 Advantages of the Developed Solution

The passenger recovery system outweighs the current manual system in many ways which include:

i. The system automatically notifies disrupted passengers through SMS.
ii. The system is able to automatically reassign passengers to new itineraries based on the parameters outlined in section 4.4.1.
iii. The optimisation model parameters are dynamically managed based on the current disruption.
iv. The system is scalable and new features can be easily plugged in.

6.4 Limitations of the Developed Solution

The developed system was not without limitations. The following were the limitations.

i. The optimisation model was developed based CPLEX optimiser whose computation time is very high compared to other optimisation concepts such as heuristics.

6.5 Summary

This chapter discussed how the research objectives were met. A comparison of the developed solution and the existing system is provided with advantages and limitations of the developed solution clearly outlined. It can be concluded that the research objectives were met as per the requirements outlined in chapter four. The next chapter gives the conclusions, recommendations and future work.

CHAPTER SEVEN: CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

7.1 Conclusions

This section provides an explanation of how research objectives outlined in chapter one were met based on literature reviewed and data analysis. The first objective was to identify the current methods of passenger recovery and their challenges. A discussion with Kenya Airways IOCC staff in charge of passenger recovery revealed that they manually reallocate passengers whenever a disruption occurs. This process takes a lot of time which could be used for other things. The second objective was to review existing solutions for passenger recovery, Section 2.4 and 2.5 looks into the solutions for passenger recovery in the airline industry in order to identify the gaps. Section 2.6 reviews the costs as a result of disruptions. The most common costs are hotel accommodation, meals, temporary passports, security and ticket refund costs.

The third objective was to design, develop and test a system for passenger recovery. This was explained in chapter four. The system architecture is shown in section 4.3 while the optimisation model is explained in section 4.4. This includes the mathematical formulation and the model architecture. Design was done based on the user requirements and section 4.4 shows the various designs. System testing was done in section 5.4 to ensure that the system worked as expected.

The fourth and last research objective was to implement and validate the system. System implementation was done in section 5.3 where the various components of the system were explained. Validation of the system was done through UAT where 87% of the users accepted the system while only 13% remained neutral and none of them rejected the system.

A review of literature as explained in section 2.3 shows that very little research has been done on passenger recovery with many researchers concentrating on aircraft and crew recovery. Section 2.2 outlines the common causes of disruption some of which are airline specific. Simultaneous aircraft and passenger recovery was reviewed in section 2.4 where the study focusses on recovering aircraft and then disrupted passengers after aircraft recovery. This means that passengers in this model are given the last priority and therefore suffer the effects of long disruptions. Section 2.5 covers the existing solutions for passenger recovery and therefore answers the second objective which is to review existing solutions for passenger recovery. Optimisation has been extensively used for solving disruptions to recover the original schedules for crew, aircraft and passengers. The third and fourth objectives are achieved in section 4.3 and section 5.3 respectively. Section 4.4 explains the mathematical optimisation model developed in this research. The model concept was based on the mathematical model reviewed under section 2.4.

7.2 Recommendations

This section outlines the recommendations. The system was developed as per the requirements explained in chapter four. However, a few improvements are recommended.

i. Integration with aircraft and crew recovery systems should be done to improve the quality of solutions.
ii. Implement the solution using heuristics to reduce computation time.
iii. Add the ability to predict the impact of future disruptions by creating disruption scenarios.

7.3 Future work

The passenger recovery problem is complex and therefore it can be looked at in different ways. The following areas could be researched more in the future either to extend this work or as a new topic.

i. Explore other optimisation tools to compare with CPLEX which has been utilised in this research.
ii. Develop and fully integrate this model with crew and aircraft models to form a complete disruption management solution.
iii. Explore quadratic functions and mixed integer programming in developing a model and how they compare with linear programming in terms of solution runtimes and quality.

REFERENCES

Abdelghany, A., Ekollu, G., Narasimhan, R., & Abdelghany, K. (2004). A proactive crew recovery decision support tool for commercial airlines during irregular operations. Annals of Operations Research.

Amadeus. (2016). Shaping the future of Airline Disruption Management (IROPS). Travel Technology Research Ltd.

Ball, M., Barnhart, C., Nemhauser, G., & Odoni, A. (2007). Air transportation: irregular operations and control: Handbooks in Operations Research & Management Science. Transportation, North-Holland, Amsterdam, Elsevier.

Bard, J., Yu, G., & Argüello, M. (2001). Optimizing aircraft routings in response to groundings and delays.

Barnhart, C., Kniker, T., & Lohatepanont, M. (2002). Itinerary-based airline fleet assignment. Transportation Science 36.

Bisaillon, S., Cordeau, J., Laporte, G., & Pasin, F. (2011). A large neighborhood search heuristic for the aircraft and passenger recovery problem.

Boehm, B. (2007). A survey of agile development methodologies. *Laurie Williams*. Retrieved from http://www.smaele.nl/documents/Williams-AgileMethods-2007.pdf

Bowen, R., & Coar, K. (2007). *Apache Cookbook: Solutions and Examples for Apache Administration*. O'Reilly Media, Inc. Retrieved from http://books.google.com/books?hl=en&lr=&id=RJhfgczxKvUC&oi=fnd&pg=PR5&dq=%22it+especially+remarkable:+it+includes+extensibility+by+design.+In+short,+if%22+%22it+so+that+it+does.+Dozens+of+extensions+(called+modules)+are+included+as+part%22+&ots=ircajXLjU5&sig=FCfCQYE5Z6j4_6ulJLpE2gLgNbc

Bratu, S., & Barnhart, C. (2005). An analysis of passenger delays using flight operations and passenger booking data. Air Traffic Control Quarterly 13.

Bratu, S., & Barnhart, C. (2006). Flight operations recovery: new approaches considering passenger recovery.

Chan, F., Chung, S., Chow, J., & Wong, C. (2013). An optimization approach to integrated aircraft and passenger recovery. Proceedings of the Institute of Industrial Engineers Asian Conference, Taipei.

Clarke, M. (2005). Passenger reaccommodation a higher level of customer service. AGIFORS Airline Operations Study Group Meeting, Mainz, Germany.

Commission for Aviation Regulation. (2011). Report on Passenger Rights Complaints in 2012.

Cook, A., Tanner, G., Lawes, A., & Jovanović. (2009, June 27). THE COST OF DELAY TO AIR TRANSPORT IN EUROPE – QUANTIFICATION AND MANAGEMENT. 13th Air Transport Research Society (ATRS) World Conference, Abu Dhabi.

Cook, A., Tanner, & Anderson, S. (2004). Evaluating the true cost to airlines of one minute of airborne or ground delay. University of Westminster, for EUROCONTROL Performance Review Commission, Brussels.

Cramer, K., & Irrgang, M. (2007). Disruption Costing Methodology: ValSim – visualization and simulation of airline operations (Boeing Commercial Airplanes). AGIFORS (Airline Group of the International Federation of Operational Research Societies) Airline Operations Conference, Denver.

Delgado, L., Martin, J., Blanch, A., & Cristóbal, S. (2016). Hub operations: delay recovery based on cost optimisation. Retrieved from http://sesarinnovationdays.eu/files/2016/Papers/SIDs_2016_paper_63.pdf

Dennis, A., Wixom, B. H., & Tegarden, D. P. (2012). *Systems analysis design, UML version 2.0: an object oriented approach* (4th ed). Hoboken, NJ: John Wiley & Sons.

Dorndorf, U., Jaehn, F., Lin, C., Ma, H., & Pesch, E. (2007). Disruption management in flight gate scheduling. Statistica Neerlandica 61.

Eggenberg, N., Bierlaire, M., & Salani, M. (2007). A Column Generation Algorithm for Disrupted Airline Schedules. Technical Report, Ecole Polytechnique Federale de Lausanne.

EUROCONTROL. (2010). Ash-cloud of April and May 2010: Impact on Air Traffic. STATFOR/Doc294 V1.0.

European Commission. (2004). establishing common rules on compensation and assistance to passengers in the event of denied boarding and of cancellation or long delay of flights, and repealing Regulation. Regulation (EC) No 261/2004 of the European Parliament and of the Council.

Heskett, J., Jones, T., Loveman, G., Sasser, W., & Schlesinger, L. (1994). Putting the Service-Profit Chain to Work. Harvard Business Review.

Holovaty, A., Kaplan-Moss, J., & Gilmore, J. (2008). *The definitive guide to Django: Web development done right*. Berkeley, Calif. : New York: Apress ; Distributed by Springer-Verlag.

Hu, Y., Xu, B., Bard, J. ., Chi, H., & Gao, M. (2015). Optimization of multi-fleet aircraft routing considering passenger transiting under airline disruption.

Hu, Y., Xu, B., Gao, M., & Chi, H. (2011). The study on the aircraft recovery with consideration of passenger transiting. In: Proceedings of the Fifth International Conference on Management Science and Engineering Management, Macau.

IBM. (2016a). IBM Modeling with OPL. Retrieved 11 January 2017, from https://www-01.ibm.com/software/commerce/optimization/modeling/

IBM. (2016b, January 1). IBM ILOG CPLEX Optimization Studio. Retrieved 11 January 2017, from http://www.ibm.com/software/products/en/ibmilogcpleoptistud

Jafari, N., & Zegordi, S. . (2010). Simultaneous Recovery Model for Aircraft and Passengers. Journal of the Franklin Institute.

Kashyap, V. (2011). Heuristic Algorithm. Retrieved 13 January 2017, from http://sookshum.blogspot.com/2011/09/heuristic-algorithm.html

Kohl, N., Larsen, A., Larsen, J., Ross, A., & Tiourine, S. (2004). Airline disruption management—perspectives, experiences and outlook. CEC Contract.

Kohl, N., Larsen, A., Larsen, J., Ross, A., & Tiourine, S. (2007). Airline Disruption Management: Perspectives, Experiences and Outlook. Journal of Air Transport Management, Vol. 13.

Le Vie, D. S., & Donald, S. (2000). Understanding data flow diagrams. In *ANNUAL CONFERENCE-SOCIETY FOR TECHNICAL COMMUNICATION* (Vol. 47, pp. 396–401).

Lettovsky, L. (1997). Airline operations recovery: an optimization approach. Georgia Institute of Technology.

Lettovsky, L., Johnson, E., & Nemhauser, G. (2000). Airline crew recovery. Transportation Science.

Maher, S. . (2015). A novel passenger recovery approach for the integrated airline recovery problem. Computers & Operations Research.

Munguti, Q. (2017). Airline service disruptions hurt carriers as travellers explore alternative flights. Retrieved 11 January 2017, from http://www.businessdailyafrica.com/Airline-service-disruptions-hurt-carriers/1248928-3482528-41sc31/

Ramakrishnan, R. (2000). Database Management Systems. Retrieved from https://works.bepress.com/madraky/19/download/

Rosenberger, J. ., Johnson, E. ., & Nemhauser, G. . (2003). Rerouting aircraft for airline recovery. Transportation Science.

Sinclair, K., Cordeau, J., & Laporte, G. (2014). Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem.

Stojkovi´c, M., & Soumis, F. (2005). The operational flight and multi-crew scheduling problem. Yugoslav Journal of Operations Research.

Sultanía, A. K. (2015). Developing software product and test automation software using Agile methodology. In *Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on* (pp. 1–4). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/7060120/

Tamba, T. A., & Joelianto, E. (2016). A linear programming approach to network utility maximization. In *Instrumentation, Control and Automation (ICA), 2016 International Conference on* (pp. 48–53). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/7811474/

University of Westminster. (2015). The cost of passenger delay to airlines in Europe. University of Westminster.

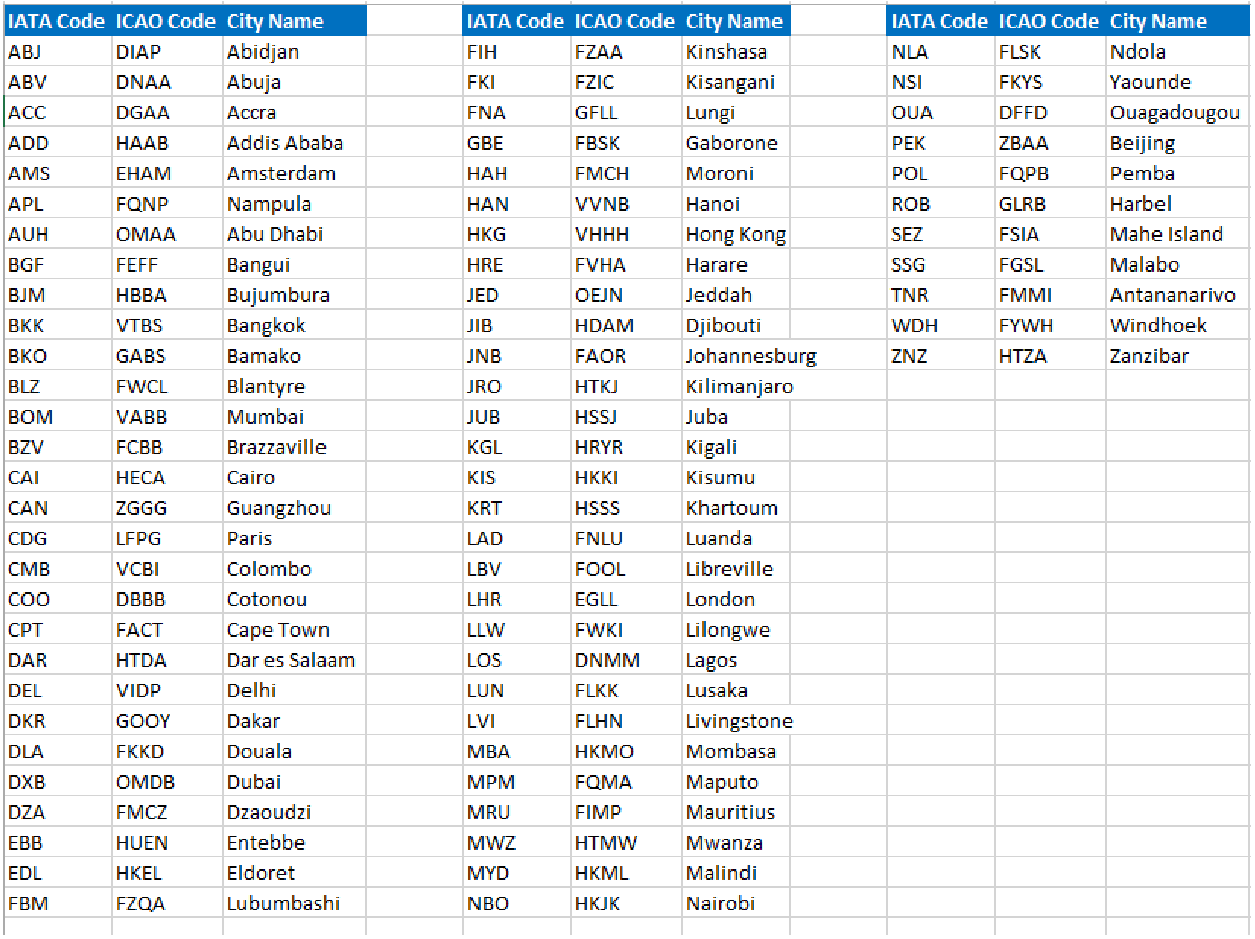
Videla, A., & Williams, J. J. W. (2012). *RabbitMQ in Action: DISTRIBUTED MESSAGING FOR EVERYONE*. SHELTER ISLAND: MANNING.

Wang, D., Sherry, L., Xu, N., & Donohue, G. (2007). System Analysis of Flight and Passenger Trip Delays in the National Airspace System. The 26th Digital Avionics Systems Conference.

APPENDICES

Appendix A: Airport List

The list of airports used in the optimisation model is shown in figure A.1.



IATA Code	ICAO Code	City Name
ABJ	DIAP	Abidjan
ABV	DNAA	Abuja
ACC	DGAA	Accra
ADD	HAAB	Addis Ababa
AMS	EHAM	Amsterdam
APL	FQNP	Nampula
AUH	OMAA	Abu Dhabi
BGF	FEFF	Bangui
BJM	HBBA	Bujumbura
BKK	VTBS	Bangkok
BKO	GABS	Bamako
BLZ	FWCL	Blantyre
BOM	VABB	Mumbai
BZV	FCBB	Brazzaville
CAI	HECA	Cairo
CAN	ZGGG	Guangzhou
CDG	LFPG	Paris
CMB	VCBI	Colombo
COO	DBBB	Cotonou
CPT	FACT	Cape Town
DAR	HTDA	Dar es Salaam
DEL	VIDP	Delhi
DKR	GOOY	Dakar
DLA	FKKD	Douala
DXB	OMDB	Dubai
DZA	FMCZ	Dzaoudzi
EBB	HUEN	Entebbe
EDL	HKEL	Eldoret
FBM	FZQA	Lubumbashi
FIH	FZAA	Kinshasa
FKI	FZIC	Kisangani
FNA	GFLL	Lungi
GBE	FBSK	Gaborone
HAH	FMCH	Moroni
HAN	VVNB	Hanoi
HKG	VHHH	Hong Kong
HRE	FVHA	Harare
JED	OEJN	Jeddah
JIB	HDAM	Djibouti
JNB	FAOR	Johannesburg
JRO	HTKJ	Kilimanjaro
JUB	HSSJ	Juba
KGL	HRYR	Kigali
KIS	HKKI	Kisumu
KRT	HSSS	Khartoum
LAD	FNLU	Luanda
LBV	FOOL	Libreville
LHR	EGLL	London
LLW	FWKI	Lilongwe
LOS	DNMM	Lagos
LUN	FLKK	Lusaka
LVI	FLHN	Livingstone
MBA	HKMO	Mombasa
MPM	FQMA	Maputo
MRU	FIMP	Mauritius
MWZ	HTMW	Mwanza
MYD	HKML	Malindi
NBO	HKJK	Nairobi
NLA	FLSK	Ndola
NSI	FKYS	Yaounde
OUA	DFFD	Ouagadougou
PEK	ZBAA	Beijing
POL	FQPB	Pemba
ROB	GLRB	Harbel
SEZ	FSIA	Mahe Island
SSG	FGSL	Malabo
TNR	FMMI	Antananarivo
WDH	FYWH	Windhoek
ZNZ	HTZA	Zanzibar

Figure A. 1 List of All Airports

Appendix B: Administrator Login Page

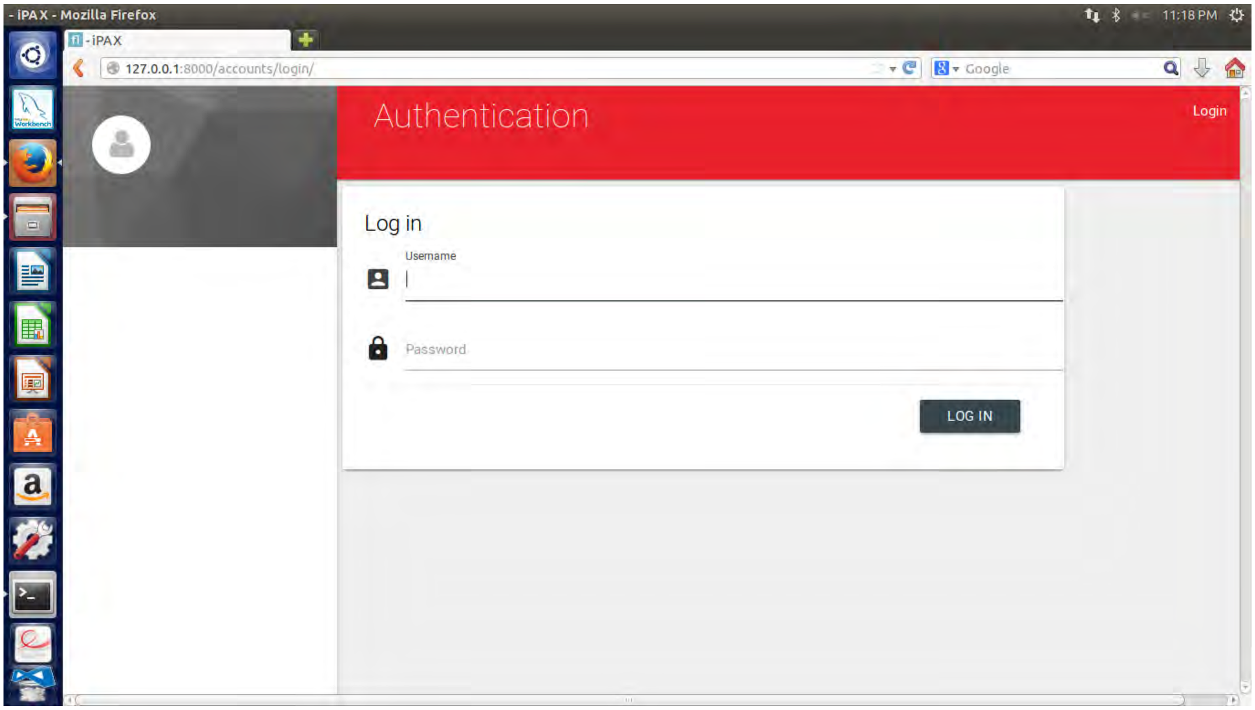
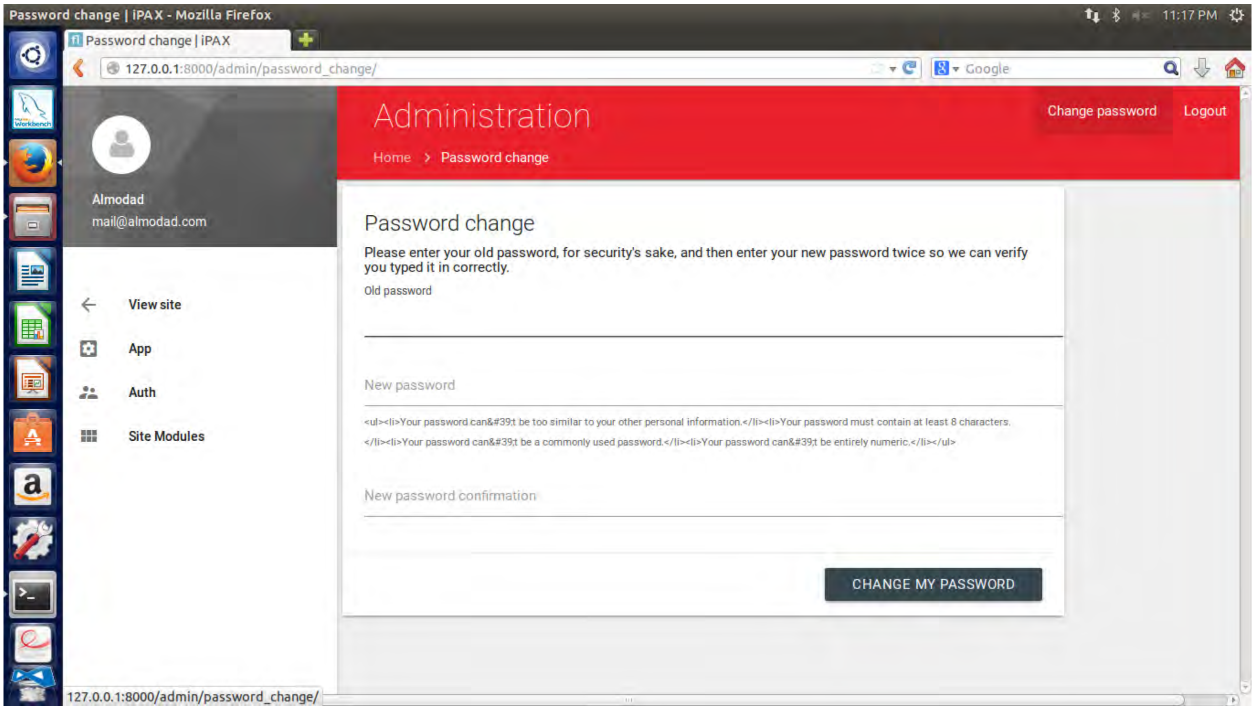


Figure B. 1 Login Page for System Administrators



Password change

Please enter your old password, for security's sake, and then enter your new password twice so we can verify you typed it in correctly.

Old password

New password

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

New password confirmation

Figure B. 2 Change Password Page

Appendix C: UAT Interview Guide

1. How would you rate the efficiency of the model in terms of computation time and accuracy?
    - Very high
    - High
    - Medium
    - Low
    - Very low
2. How do you find the ease of us of the system in terms of navigation?
    - Easy to use
    - Average
    - Hard
3. How is the user friendliness of the system look and feel?
    - Very user friendly
    - Good
    - Not user friendly
    - I don’t know
4. Do you think the system meets all the requirements?
    - Yes
    - No
    - Not sure
5. Would you use this system for your day-to-day operations?
    - Yes
    - No
    - Not sure
    - Neutral
6. What do you think should be done in order to make the system more useful?

    ……………………………………………………………………………………..

    ……………………………………………………………………………………..

Appendix D: System Analysis Interview Guide

1. Please tick the common causes of disruption in the airline?
    - Technical issues
    - Natural Calamities
    - Congestion at check-in
    - Power outage
    - Political unrest
    - Crew/Staff
    - Systems failure
    - Congestion at air traffic control

    Any other? Please specify

    ………………………………………………………………………………………………
2. Which among the selected causes is the most prevalent?

    ………………………………………………………………………………………………
3. What is the frequency of occurrence of the number 2 above?
    - Every few hours
    - Daily
    - Every 3 days
    - Weekly
    - Monthly
4. What costs do you incur as a result of a disruption?
    - Change of reservation cost
    - Accommodation cost
    - Meals cost
    - Landing fees
    - Transport cost
    - Visa cost
    - Without prejudice cost
    - Medical cost

Are there any other costs? Please specify

……………………………………………………………………………………………………

5. How do you recover disrupted passengers?
    - Hotel accommodation
    - Meals (lunch, snacks)
    - Rebooking
    - Ticket refund

Please specify any other recovery options

……………………………………………………………………………………………………

6. What is the criteria of selecting a recovery option?
    - Cost
    - Time

Others? Please specify

……………………………………………………………………………………………………

7. How do you notify disrupted passengers?
    - Email
    - SMS
    - Face to face communication

Any other? Please specify

……………………………………………………………………………………………………

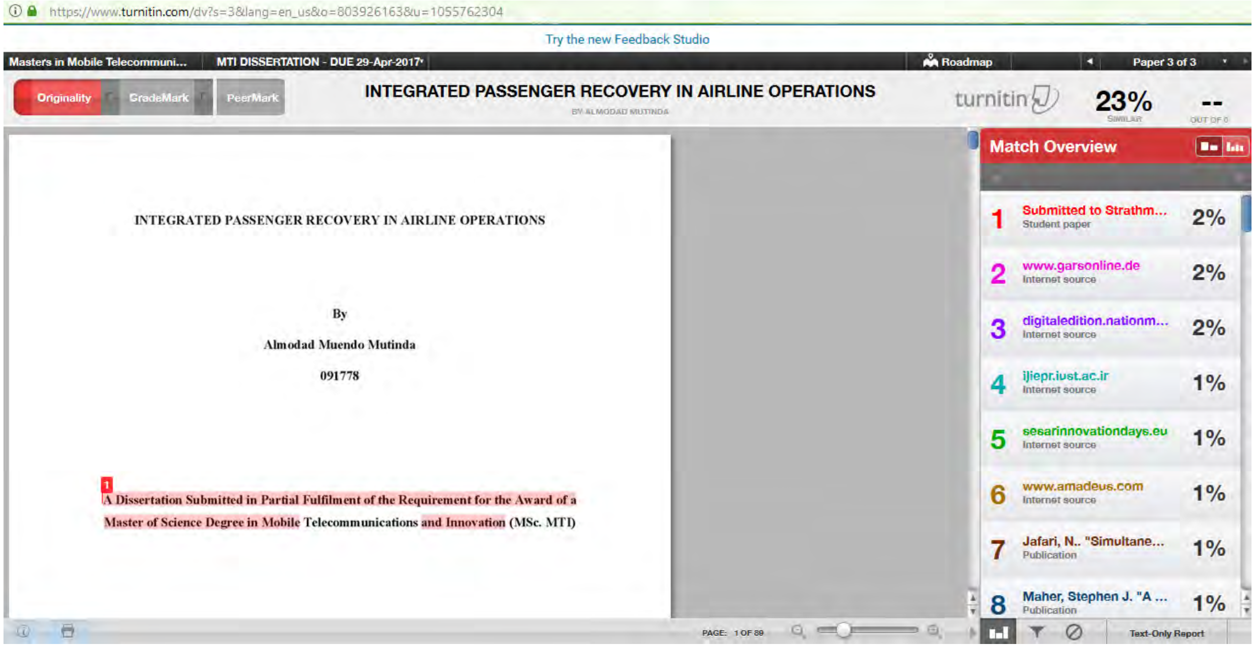
8. What methods do you use to handle disruptions?
    - Manually
    - Using AMADEUS
    - Using BMS

Others? Please specify

……………………………………………………………………………………………………

Appendix E: Turnitin Report

The turnitin report is shown in figure E.1



INTEGRATED PASSENGER RECOVERY IN AIRLINE OPERATIONS

By

Almodad Muendo Mutinda

091778

1 A Dissertation Submitted in Partial Fulfilment of the Requirement for the Award of a Master of Science Degree in Mobile Telecommunications and Innovation (MSc. MTI)

1	Submitted to Strathm... Student paper	2%
2	www.garsonline.de Internet source	2%
3	digitaledition.nationm... Internet source	2%
4	ijiepr.iust.ac.ir Internet source	1%
5	sesarinnovationdays.eu Internet source	1%
6	www.amadeus.com Internet source	1%
7	Jafari, N.. "Simultane... Publication	1%
8	Maher, Stephen J. "A ... Publication	1%

Figure E.1 Turnitin Report