

AIRPORT MANAGEMENT SYSTEM

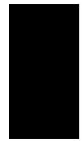


Table of Contents

1. Introduction	3
1.1 What the Problem is	3
1.2 Goals for the Project	3 - 4
1.3 Motivation for the Project	4
1.4 Process Flow Preview	4
2. Analysis and Design	5
2.1 Plan for Requirements Engineering	5
2.2 Functional Requirements	6 - 9
2.3 Non Functional Requirements	9 - 13
2.4 Use Cases	13 - 15
2.5 Models	16 - 19
3. Project Plan	19
3.1 Task Description	19 - 20
3.2 Task Assignment	20 - 21
3.3 Deliverables and Milestones	21
3.4 Project Schedule	22

4. Conclusion	23
4.1 The Problem and Solution	23
4.2 The Team and the SE Process	24
4.3 The Organization's Benefits	24
5. User Manual	25
5.1 Software Description	25
5.2 How to use the Software	25 - 27
5.3 Troubleshooting Common Problems	27 - 28

1. Introduction

1.1 What the Problem is

An airport management is wishing to have a user interface that will allow their authorized employees to organize the pilots, hosts, flights, flight companies and airports. The user can add or remove a pilot, host, flight company and airport also organize a new flight and set its crew. The Web App will be responsive, allowing for the user to view it only desktop computers. Also, the user will also be able to search the flight according to its wish and reserve it for any passengers.

1.2 Goals for the Project

This software, called 'Airport Management System', will allow for the company to access their database securely and safely in a user-friendly online environment. Also, the software allow for them to change the flight and employee informations with ease. The software will be in sync with Web App, allowing for reservation service for their customers.

Authorized users will be able to search flights, airports and flight company and see their details. The authorized user will also be able to add or delete pilot, host, passenger, flight, flight company and airport from the system. Also, the authorized user can organize the existing flights or create a new flight and reserve a flight for any passenger.

There should be a Web App version for the software to connect with for those who wish for a quick flight reservation service. The Web App will be responsive to only desktop computers. The software itself will be available on all computer platforms that are running any aspect of Linux, Windows 7, Windows 8, and Mac operating systems.

1.3 Motivation for the Project

With this airport management in need of a better system, we felt it was our obligation to help them in their time of need. To develop such a system that would not only ease the burden on the airport board members, but the airport management itself. Our team has an immense amount of knowledge when it comes to problem solving, programming. Not only would we strive to give the airport management service everything they desired, but we will continue to make sure the software is at its very best and beyond.

1.4 Process Flow Preview

For our process flow, we plan on how to design the software based on object oriented principles firstly. Our essential way is discussing about the software requirements and reaching the common approach. After this discussion process, all requirements as interfaces, classes etc. is meticulously created. Visualization is really important in such comprehensive projects. That's why, after the creation of based requirements, UML Diagrams is created as well. Then, We start to discuss which platform to use for Web App. After this discussion, we choose the Angular to develop Web App and use the JSON format to store data.

2. Analysis and Design

2.1 Plan for Requirements Engineering

Inception Task:

The goal for the beginning is to identify the software main purpose. Everything else will be built on top of this main step. That's why, perhaps this task is the most important task.

Therefore, this task should be done slowly, minutely and depending on a standard. To set this standard some questions must be asked. Questions that will be asked:

What are the basic functions? (What do you want the website and software to do? - What tasks/problems is the product supposed to accomplish?)

What sort of customers are you targeting with this new website? (Who is going to use it?)

Where this software will be implemented?

Will there be any issues or constraints that may affect the planning and construction? What will be done under this circumstance?

Elicitation Task:

Our goal at this stage is to identify the problem, propose solutions, and talk amongst each other on the many different approaches. The plan is to get a grounded idea of what the objectives for the system are, what should be accomplished, and how the overall system fits into the airport management system. The reason of this, this step will form the basis for the rest of the project.

Elaboration Task:

Information gathered from the inception and elicitation stage are grouped together and refined during this stage. Some models and diagrams are created during this stage to see more clearly the project details. Scenarios were created to describe and aid in understanding how the customer will interact with the website and how the employees will interact with the software. Any attributes are to be defined as well as how each function interacts with one another.

Specification Task:

During this task, we will work on adding new things on the basics created in previous tasks. Design of the user interface and some other extra attributes are created on this stage. How the data will be stored? Which framework and interface will be used with this software? All these questions are discussed and started to be implemented.

Validation Task:

During this stage, any requirements stated are to be ensured that they are clearly defined. No miss-interpretation should be present and any that exist should be resolved. All and any

Requirements Management:

Any changes that may occur throughout the project stages should be handled with clarity and care. Any potential changes would be looked over, discussed and determined if the time allotted for the construction of the project can allow for such a change - that is.

2.2 Functional Requirements

Hardware Requirements:

The software should be ran on any sort of desktop or laptop environment, regardless of the operating system. But the software has not the potential of running on tablets or mobile devices. Essential input/output devices are keyboards and mouses.

Website Interface - Primary Tasks:

- View all pilots
 - Data of pilots are read from JSON file to call all pilot objects and display them accordingly
- View all hosts
 - Data of hosts are read from JSON file to call all host objects and display them accordingly
- View all flights
 - Data of flights are read from JSON file to call all flight objects and display them accordingly
- View all flight companies
 - Data of companies are read from JSON file to call all company objects and display them accordingly

- View all airports
 - Data of pilots are read from JSON file to call all pilot objects and display them accordingly.

Website Interface - Secondary Tasks:

- Allow to add, remove and edit a pilot
 - There is a add button to add a new pilot at the top of the page of pilots. Also, there is be a delete button on the most-left side of the pilot's row. Finally, user can edit the pilot with double click on it.
- Allow to add, remove and edit a host
 - There is a add button to add a new pilot at the top of the page of pilots. Also, there is be a delete button on the most-left side of the pilot's row. Finally, user can edit the pilot with double click on it.
- Allow to add, remove and edit a flight company
 - There is a add button to add a new pilot at the top of the page of pilots. Also, there is be a delete button on the most-left side of the pilot's row. Finally, user can edit the pilot with double click on it.
- Allow to add, remove and edit an airport
 - There is a add button to add a new pilot at the top of the page of pilots. Also, there is be a delete button on the most-left side of the pilot's row. Finally, user can edit the pilot with double click on it.

- Allow to create or organize a flight
 - The user can create a new flight by entering some informations about flight such as source airport, departure time, pilot etc. Also, user can organize this flight.
- Allow to reserve a ticket for passengers
 - The user can reserve a ticket for passenger according to information of passenger.

2.3 Non Functional Requirements

Performance Requirements:

- Ability to maintain mass amount of users on the website at once without crashing
- Speedy performance / transmission of data
- Have a quick recovery time if anything were to go wrong
- Display accurately and efficiently on desktop computers

Security Requirements:

- Secure any transmissions of private information between the employees and passenger and the airport management
- Prevent any potential threats through the forms or search boxes
- Verify website security certificates (that lock in the address bar)

Quality Attributes:

- Maintain a user friendly environment that is visually appealing
- Easy to see and use navigation
- Maintain readable content for exceptions and errors
- Seperate pages for each different object type
- Ease the edit pages like using forms for multiple objects and dates

Screenshot Mockups:

Pilot

Host

Passenger

Flight

Flight Company

Search...

<div></div>	<div>Surname</div>	<div>Rank</div>	<div>Phone</div>	<div>Flight Company</div>	<div>Delete</div>
	Parlayan	first-pilot	123456	THY	<div></div>
	Özdemir	First Pilot	12312313	Fly Emirates	<div></div>

Name:	Surname:
<input type="text" value="Idrishan"/>	<input type="text" value="Parlayan"/>
Flight Company:	Rank:
<input type="text" value="THY"/>	<input type="text" value="first-pilot"/>
Phone:	Birthday:
<input type="text" value="123456"/>	<input type="text" value="8/31/1999"/>
Gender:	Weight:
<input type="text" value="male"/>	<input type="text" value="80.2"/>
Height:	Work Experience:
<input type="text" value="183"/>	<input type="text" value="23"/>
Salary:	
<input type="text" value="30000"/>	

Save

Cancel

Name:

Idrishan

Surname:

Parlayan

Flight Company:

THY

Name

Q

THY

Fly Emirates

Rank:

first-pilot

Birthday:

8/31/1999

Weight:

80.2

Work Experience:

23

Salary:

30000

Save

Cancel

Name:

Idrishan

Surname:

Parlayan

Flight Company:

THY

Phone:

123456

Gender:

male

Height:

183

Salary:

30000

Rank:

first-pilot

Birthday:

8/31/1999

<

AUGUST 1999

>

SUN	MON	TUE	WED	THU	FRI	SAT
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Pilot	Host	Passenger	Flight	Flight Company	
					<div>+</div> <div> <div>Q</div> <div>Rid</div> <div>×</div> </div>
Name	Surname	Rank	Phone	Flight Company	Delete
Ricvan	Özdemir	First Pilot	12312313	Fly Emirates	<div> <div></div> </div>

Pilot	Host	Passenger	Flight	Flight Company	
					<div>+</div> <div> <div>Q</div> <div>Search...</div> </div>
Source Airport	Destination Airport	Departure Time	Arrival Time	Plane code	Delete
Adnan Menderes Airport	Sabiha Gokcen Airport	2020-10-25	2020-10-25	TR-45	<div> <div></div> </div>
Sabiha Gokcen Airport	Adnan Menderes Airport	2020-6-1	2020-6-1	TR-16	<div> <div></div> </div>

					<div> <div>Q</div> <div>Search...</div> </div>
Availability	Seat Number	Passenger	Price		
<div> <div></div> </div>	1		101		
<div> <div></div> </div>	2		102		
<div> <div></div> </div>	3		103		
<div> <div></div> </div>	4		104		
<div> <div></div> </div>	5		105		
<div> <div></div> </div>	6		106		
<div> <div></div> </div>	7		107		
<div> <div></div> </div>	8		108		
<div> <div></div> </div>	9		109		
<div> <div></div> </div>	10		110		

10

25

50

1

2

3

2.4 Use Cases

Use Case #1: *Buying ticket*

Goal in Context: To buy a ticket of a flight chosen by a given departure date

Preconditions: The seat must be empty before the choice of passenger.

Trigger: A flight is chosen by the passenger.

Scenario:

1. Passenger: Logs into Airport Management System(Enters Username/ Password)
2. Passenger: Enters a date for wished departure date to the searching bar.
3. Passenger: Selects a displayed flight.
4. Passenger: If there is a green symbol on any seat, when the passenger clicks it, it enters necessary information.

Priority: Essential, must be implemented

When available: First increment

Frequency of use: Couple times per day/ week

Use Case #2: *Adding a new Pilot to an airport company on the system*

Goal in Context: To add a new pilot or host to company

Preconditions: Pilot or host must be not working in another company

Scenario:

1. Pilot/host page is opened
2. Add button is clicked.
3. Personal information of pilot/host is entered.
4. Information is saved and adding process is completed.

Exceptions:

1. Invalid name-surname entry: Page gives a warning when a name or surname entered is 2 letters or less, until a valid name and surname are entered.

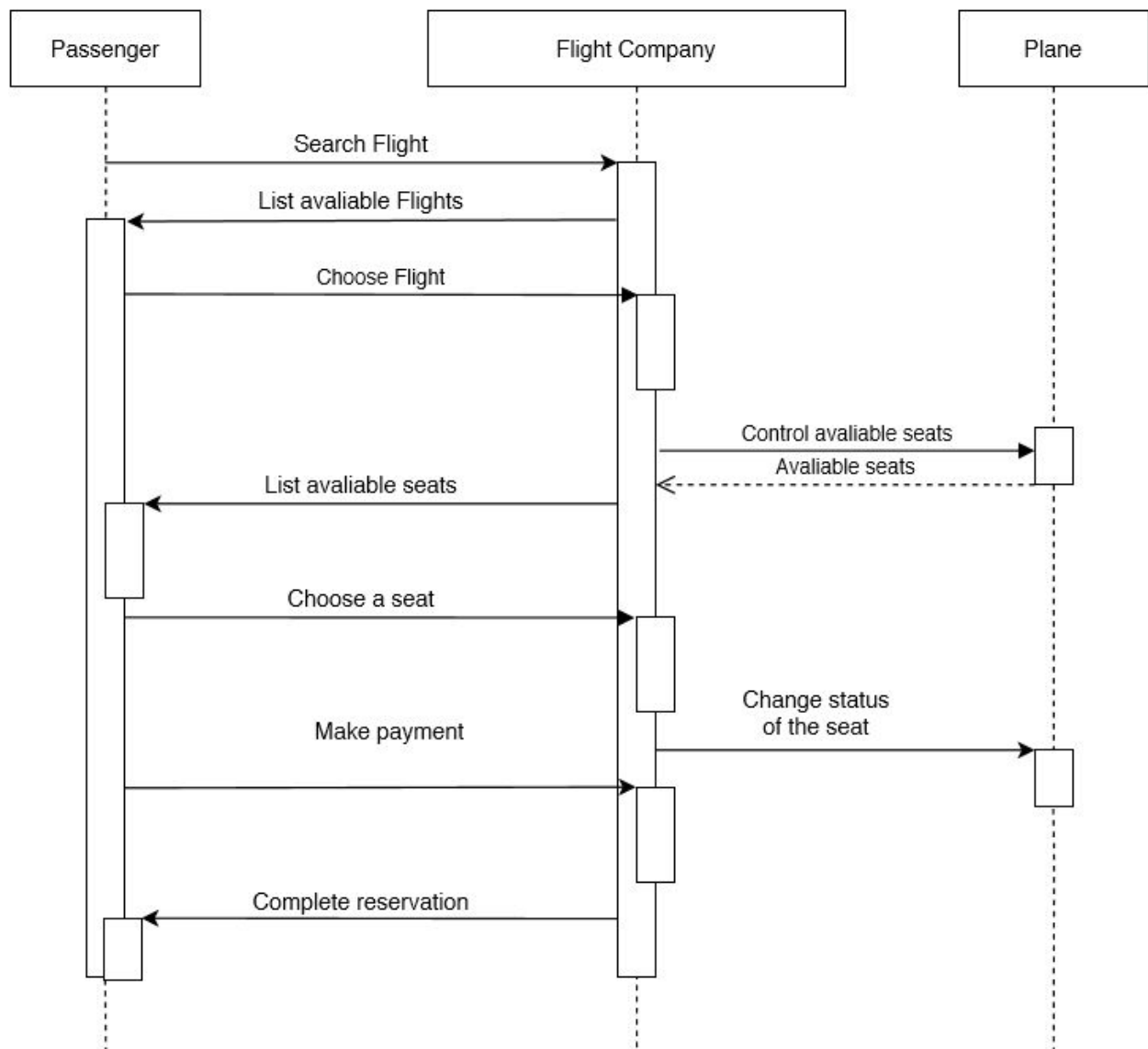
Priority: Essential, must be implemented

When available: First increment

Frequency of use: Once in every month

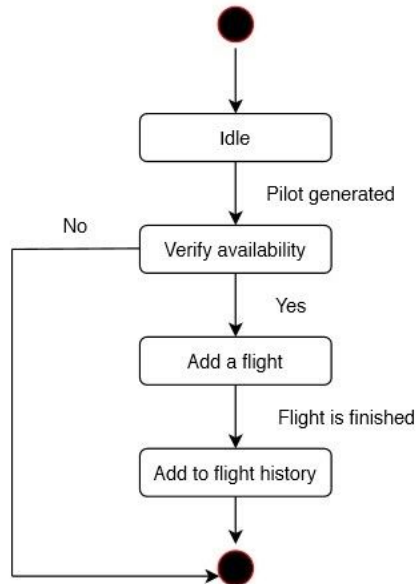
2.5 Models

SEQUENCE

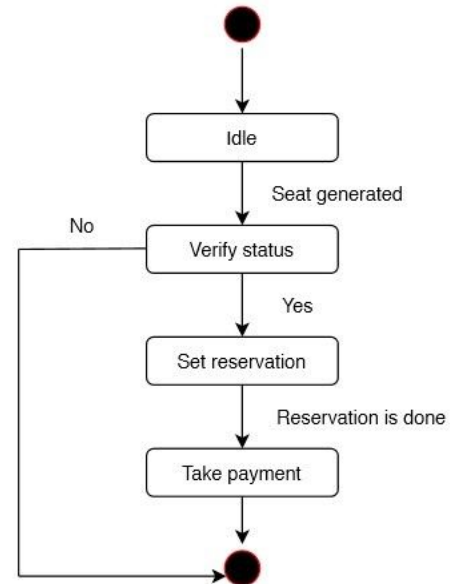


STATE

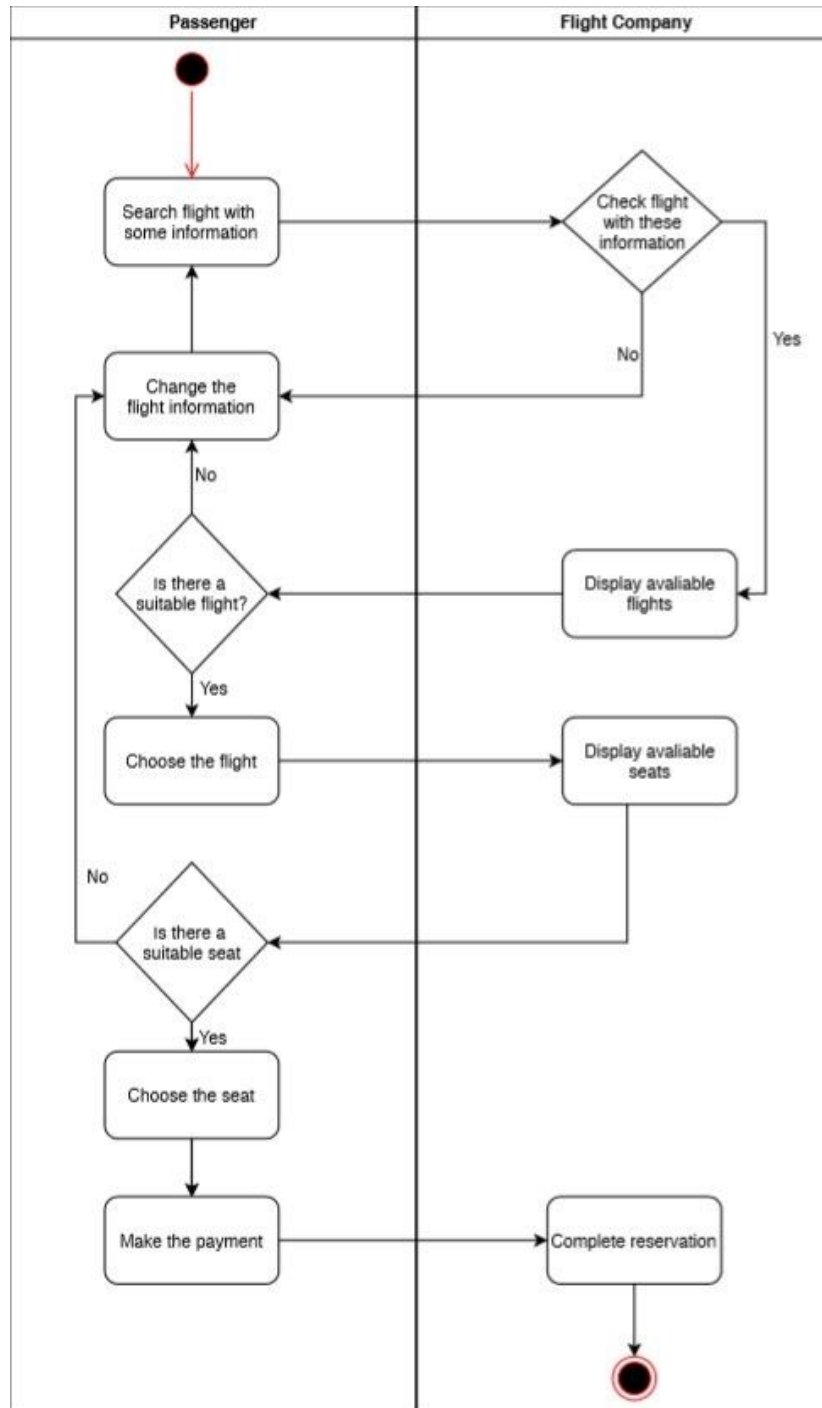
OBJECT:
PILOT



OBJECT:
SEAT



ACTIVITY



3. Project Plan

3.1 Task Descriptions

Design Models and Mockups

Designing the models and mockups help to ensure clarity in view of the project as well as how it works. Stakeholders are to sit through this process as drawings are created.

Database Creation

A database is created as JSON format. Pilot, host, passenger, flight, plane, airport objects are stored in this JSON file. The software uses data from this JSON file when necessary.

Website Creation

The web site will be designed using Angular programming language using the guide of the mockups, requirements, and models. The website will be implemented using high-quality design techniques. It should allow for users to immediately see flights and seats that are available. They can also have the ability to search for a certain pilots, hosts etc.

Testing

Testing will be implemented. Test cases may be used to guide and understand the basic actions of users .Any bugs or errors that occur will be identified and resolved.

Finalization and Reports

All testing and function processes are finalized at this stage. Reports will be created to ensure all information and functionality is clear in order to make the user manual and to help ensure employees can use the software with ease.

3.2 Task Assignment

Assignments were distributed evenly among the group assigned to the project. All three worked together in the project planning, sharing in the opportunity of any models and analyzing all specifications made by the proprietor.

Idrishan took a part in creation of all necessary classes, general design of the project and UML Diagrams for testing. He also took a part in general design and coding of angular project. He contributed the creation of the JSON database. He developed user interface with angular.

Rıdvan took a part in creation of all necessary classes, general design of the project and UML Diagrams for testing. He contributed the creation of the JSON database. He also helped Idrishan to create a web interface with Angular.

Alp took a part in general design of the program, required classes and methods and relations among them; also contributed to design of required UML diagrams and implementation of classes and methods in Java.

Reports were created throughout the process by all three group members and gathered to accurately and sufficiently create this final report.

3.3 Deliverables and Milestones

We had three major Milestones in this project:

1. Completion of Requirements Gathering.
2. Completion of Design and code.
3. Completion of Demonstration.

These milestones were all completed on schedule and yielded a Deliverable at the end of each.

Our four corresponding deliverables (respectively) in this project were as follows:

1. General outlines of software design.
2. Activity, sequence, state and use case diagrams.
3. A prototype of the software written in JAVA..
4. A finished and user friendly GUI (Graphical User Interface).

3.4 Project Schedule

The first month of the project start date (February) was used mainly for requirements setting and gathering. This took (over the course of two weeks) roughly 12 hours. Through the first two weeks of March, we were able to begin designing, which took 34 hours. After that, Web implementation process with "angular", started. It took approximately 50 hours. We then wrote out the user manual, and gave a final report of the software. The time this took was about 13 hours. We finished the project in late May.

The percentage breakdown was as follows:

Requirements: 12 hours - 11.02%

Design and code: 34 hours - 31.19%

Web Implementation: 50 hours - 45.87%

Manual and Final Report: 13 hours - 11.92%

Total: 109 hours - 100%

4. Conclusion

4.1 The Problem and Solution

The problem that undergo for Airport Management System was a lack of real time ticket reservation for their customers. They wanted a website that would allow for users to purchase a flight ticket with ease and admiration. Customers should have been able to reserve a ticket and view flights with logging into the system. Requests should be transferred directly into the database and displayed on the software within the company building to allow for a proper response and review to commence.

The solution was to provide Airport Management System with an user-friendly web application that would allow for customers to access and use on a wide range of devices: desktops, laptops. The website is designed to stay up to date by giving administrators the ability to change/add/remove any featured object like pilot, host, flight, flight company etc. on the site. Customers make a purchase ticket request through the Airport Management System. Airport Management admin can access all necessary information about customer and its wishing flight and organize them.

This software provides an easy-to-use interface to allow for simple access to flight ticket requests and employee informations such as pilot, host etc.

4.2 The Team and the SE Process

The Software Engineering process we used was the linear method. In this method, we start in the beginning of the design and problem, allowing all developer working on the software to be an active part of every aspect of the engineering.

4.3 The Organization's Benefits

Our Organization benefited greatly from the production of this software. Our benefits was not about the financial. The point we benefit from is the positive effect of the technical part of the software on our development. We've learned the requirements of the OOP and how to implemented it in this project. Also we've learned the UML diagrams and how to design the project by using these diagrams. Usage of github is also efficient part of this project. We've learned how to use git in our IDE's and how to use push, pull, merge operations on github. In the web application section, we learned how to works a framework, database. Also we learned how to use the JSON format as a database and usage of Angular.

5. User Manual

5.1 Software Description

The web application will allow to view, search, add, edit for currently pilots, hosts, passengers, flight and flight companies but flight can't be edited. The web application will use to buy ticket according to selected flight and available seats.

5.2 How to Use the Software

To run project user should go to webclient directory in project and run **ng serve** comment on terminal and user should go to Airport.json directory with another tab and run **json-server --watch Airport.json** comment on terminal. These to command must run same time to use user interface.

The web application designed for one user. User will see navigation bar on top of page when enter the main page. This navigation bar has 5 button to select page. The user should click on this button to go to page that want to go. After click a button system will navigate user to this page. First page is pilot page. If user want to go to pilot page, user should click pilot button on navigation bar. When user enter the pilot page , all pilots will be listed in table. If user click double on a row , edit popup will open and user can be edit the pilot in this row.

If user want to add new pilot user should click add button next to search box.

User can search a pilot from search box. User should click save button in edit or add popup to save changes. User click cancel button in edit or add popup to cancel changes. User should click delete button to delete row from table.

Second page is host page. If user want to go to host page, user should click host button on navigation bar. When user enter the host page , all hosts will be listed in table. If user click double on a row , edit popup will open and user can be edit the pilot in this row.

If user want to add new host user should click add button next to search box. User can search a host from search box. User should click save button in edit or add popup to save changes. User click cancel button in edit or add popup to cancel changes. User should click delete button to delete row from table.

Third page is passenger page. If user want to go to passenger page, user should click passenger button on navigation bar. When user enter the passenger page , all passengers will be listed in table. If user click double on a row , edit popup will open and user can be edit the passenger in this row. If user want to add new passenger user should click add button next to search box.

User can search a passenger from search box. User should click save button in edit or add popup to save changes. User click cancel button in edit or add popup to cancel changes. User should click delete button to delete row from table.

Forth page is flight page. If user want to go to flight page, user should click flight button on navigation bar. When user enter the flight page , all flights will be listed in table. If user click double on a row ,seats page will open. If user want to add new flight user should click add button next to search box.

User can search a flight from search box. User should click save button in edit or add popup to save changes. User click cancel button in edit or add popup to cancel changes. User should click delete button to delete row from table.

Fifth page is passenger flight company. If user want to go to flight company page, user should click flight company button on navigation bar.

When user enter the flight company page , all flight companies will be listed in table. If user click double on a row , edit popup will open and user can be edit the flight company in this row. If user want to add new flight company user should click add button next to search box. User can search a flight company from search box. User should click save button in edit or add popup to save changes. User click cancel button in edit or add popup to cancel changes. User should click delete button to delete row from table.

If user want to buy a ticket user should go to flight page and user should click double which flight user wants. All seats will list for this flight when user select a flight. User can select ticket that available button is green. User should click double on ticket and enter information about passenger and click save button to buy ticket.

5.3 Troubleshooting Common Problems

The Software

Problem: Loading incorrectly / Session has unreach

- Make sure your json server is working .

Problem: project can't run

- You should make sure Node.js, npm angular cli.
- You can check them with this terminal commend: `node -v` , `npm -v`
- You can run **`sudo apt install nodejs`** to install Node.js on linux.
- npm will come with Node.js installation
- **`npm install -g angular-cli`** commend install angular to your computer.

Problem: Json server can't run

- Make sure you have json server on your computer.
- **`npm install -g json-server`** this commend installs json server on your computer.
- **`json-server --watch Airport.json`** this commend run json server to watch Airport.json file.