

#### German Jordanian University

#### School of Electrical Engineering and Information Technology

**Very Blue Sky**

*A project submitted*

*in partial fulfillment of the requirements for the degree of Bachelor in Computer Science*

#### by

Saleh Ramzi Saleh Anouz (20191501016)

Hasan Haitham Hasan Al-Omari (20191501013)

#### Supervised by

Professor Ismail Hababeh

#### 22 / Aug / 2023

**ACKNOWLEDGEMENT**

We would like to thank our friends and family for their constant support and firsthand user interface testing, their notes and user experience points that were taken into consideration when developing the project to make it an extremely user friendly based, and simple, and extremely reliable.

special thanks to our instructor and supervisor Professor **Ismail Hababeh**, for his continuous flow of ideas and questions that helped build the base of this project.

Special thanks to Professor **Ahmad Barghash**, for his ideas and projects in Kotlin and XML, that were heavily used in this project.

Special thanks to Professor **Samer Nofal** for building our knowledge in Neural networks that was used in this project.

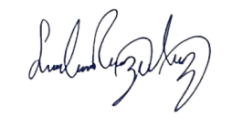
Special thanks to our User experience testers and recommendations givers, Dima Tarabaih (for direct continuous user experience interactions), Maya Bohairy, Mohammad Abu Zaid, Yousif Al-Ayaydeh, Saif Anouz, Yahya Al-Khatib, Baraa’ Alah-Rashi, Abdullah Abu Ali, Khaled Al-Omari, Rakan Abu Joudeh, Tareq Salah, and Özlem Köse (PhD in fine arts).

And a thanks to everyone who got be part of the survey’s leading up to deciding the model, the UI, the bettering of the user experience and the color palette used.

**UNDERTAKING**

This is to declare that the project entitled “Very Blue Sky” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Bachelor in Computer Science” at Computer Science Department, School of Electrical Engineering and Information Technology, German Jordanian University.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student Name: Saleh Ramzi Anouz 

Student Name: Hasan Haitham Al-Omari 

Note: sign across your name

**ABSTRACT**

*Very Blue Sky, is an Android mobile app, that can help the user book flights in his desired filtered enhanced way, track the flight by date, then move the flight to the previous flight section, i.e.: the user has the full control of booking and tracking of his flights, also the app can sort the upcoming flights by price, and track the upcoming flights by date.*

*Also the app helps push the research of implementing Neural Networks and AI into booking flights, by predicting certainty of a specific flight to be delayed and recommending it or not based on that.*

*Also the admin, gets to add and remove flights from DB from the app directly.*

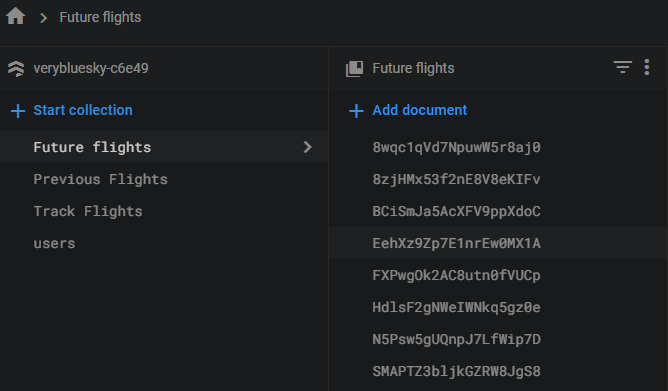
**Table of Contents**

(Fill where applicable)

1. [LIST OF TABLES vii](#_bookmark0)
2. [LIST OF FIGURES viii](#_bookmark1)
3. [CHAPTER 1: Introduction 1](#_bookmark5)
   1. [Overview 1](#_bookmark6)
   2. [Project Motivation 1](#_bookmark7)
   3. [Problem Statement 1](#_bookmark8)
   4. [Project Aim and Objectives 1](#_bookmark9)
   5. [Project Scope 1](#_bookmark10)
   6. [Project Software and Hardware Requirements 2](#_bookmark11)
   7. [Project Limitations 2](#_bookmark12)
   8. [Project Expected Output 2](#_bookmark13)
   9. [Project Schedule 2](#_bookmark14)
   10. [Project, product, and schedule risks 2](#_bookmark15)
4. [CHAPTER 2: Related Works 3](#_bookmark16)
   1. [Introduction 3](#_bookmark17)
   2. [Existing Systems 3](#_bookmark18)
   3. [Overall Problems of Existing Systems 3](#_bookmark19)
   4. [Overall Solution Approaches 3](#_bookmark20)
5. [CHAPTER 3: Methodology 5](#_bookmark27)
   * 1. [Overview 5](#_bookmark28)
     2. [Software architecture 5](#_bookmark29)
        1. [Logical view 5](#_bookmark30)
        2. [Process view 5](#_bookmark31)
        3. [Physical view 5](#_bookmark32)
        4. [Details of each component in a separate section 5](#_bookmark33)
     3. [Software design 5](#_bookmark34)
        1. [UML sequence/communication diagram 5](#_bookmark35)
        2. [Class diagram 5](#_bookmark36)
        3. [ER diagram (if any) 5](#_bookmark37)
        4. [State transition diagram 5](#_bookmark38)
     4. [User interface design (prototype) 5](#_bookmark39)
6. [CHAPTER 4: Experimental Results and Performance Evaluation 6](#_bookmark40)
   * 1. [Description of Implementation 6](#_bookmark41)
     2. [Programming language and technology 6](#_bookmark42)
     3. [part of implementation if possible 6](#_bookmark43)
7. [CHAPTER 5: Conclusion and Future Works 8](#_bookmark48)
8. [CHAPTER 8: References 9](#_bookmark49)

# LIST OF TABLES

Our data bases:

 A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

TABLES

A screenshot of a computer

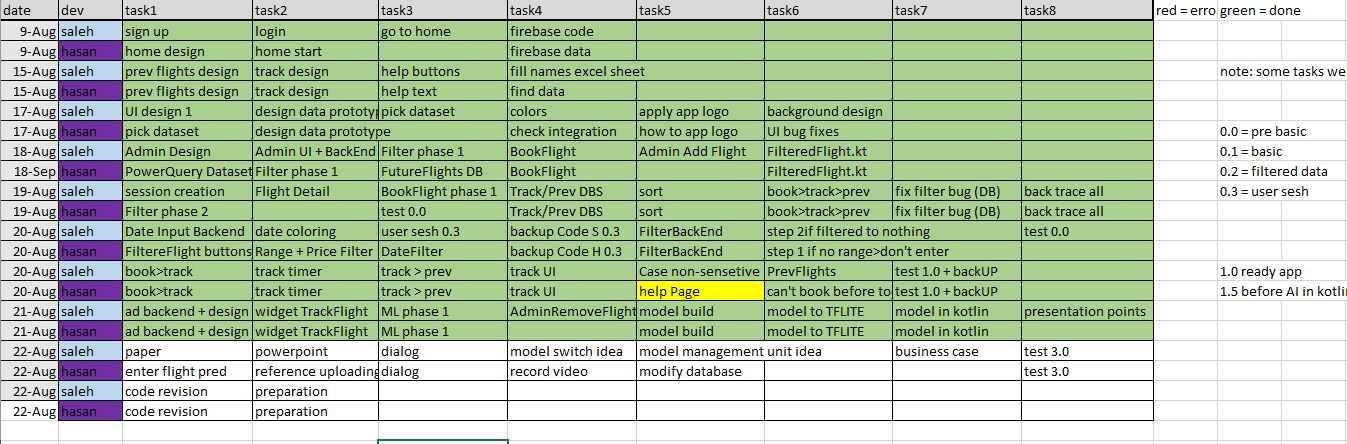
Description automatically generated

DATASET for the ML tenserflow model

A table of numbers and letters

Description automatically generated

Post project checks and modifications task list that shows how work was split and performed in the later days of the project i.e.: the implementation



# LIST OF FIGURES

**CHAPTER 1: Introduction**

The app is based on providing the user with the best experience while trying to book a flight, where it has an impressive layout and design format, recommended by two students with PhD’s in both fine and visual arts.

The problem in apps like this one and the general problem is: modern apps are complex and user reliant, and they don’t provide any idea or will to integrate AI with the flight booking apps.

Our solutions: provide a simple well design, user attracting app that can scale on an extremely beneficial base, where more companies can buy into the application, and have they’re flights posted on the application, tracked, updated, and completed

As well as implement a base for further research into integrating AI in mobile applications, where our AI side (AVA BOT), can predict chance of delay of any flight based on the date of the flight, the destination airport, and the origin airport, and it provides a base for future enhancements for the bot, where if you add more aviation companies, the bot can learn that specific companies tend to delay flights more and so on, which shows the base for a perfectly and largely expandable app and AI BOT.

Also it provides the methodology to control and manage the AI BOT, to teach it how to predict better and faster.

## Overview

A user friendly, flight booking app, that builds the base to integrate AI into such mobile applications.

## Project Motivation

Q1: What are the reasons behind your choice to develop this project?

The complexity of other flight booking apps, and to implement a base of integrating the use of AI in flight booking.

Q2. Why your project is important?

It’s built on the bases of abstract art; the user should fall in love with the simple design, but at the same time benefit from complex backend and AI in the project.  
as well as moving a step forward in the Science of AI, where more companies can focus on brining machine learning into app development, as well as providing an implementation of AI in Amman Jordan.

Q3. What is the new idea that have been proposed by this project?

Implementing a simpler form of flight booking, and most importantly, the inspiration on implementing an AI neural network in the application, and further insist on brining AI into our daily life, as well as help expanding the world of AI.

## Problem Statement

The problem we faced was the limitations of AI use in app development, as brining a large and scalable model of a machine learning network, isn’t currently possible through libraries, but only can be used if the model is uploaded on a server, as well as the lack of the use for the idea of bringing AI into flight booking applications.

## Project Aim and Objectives

Q1. What is the goal that this project wants to achieve?

Provide a simple user friendly UI, with a secure user environment. And a base AI model that could inspire more integration between AI and mobile apps.

Q2. How this project can achieve this goal?

By studying the demo’s most important parts:

The UI and the art behind the designs

The complex Kotlin core

Making the backend complicated with functions and algorithms to make the frontend simple and user friendly

By presenting the results of predicting a data set of flights from the AI model.

## Project Scope

The overview first then snippets of code.

Feature 1: the main page, where you can you sign up and login, based on a database for user authentication back end.

Feature 2: when user logs in the app, it goes into the Home page, that contains a widget for the upcoming flight that the user has booked, an ad for the cheapest up coming flight, both of these use an algorithm, one was developed by Saleh Anouz and the other was developed by Hasan Al-Omari, and 4 buttons, each goes to a page:

Feature 3: button1, Book Flight, where the user goes to filter flights by what he wants, and can sort it using sort by price, or sort with ai, which in theory shows chance of delay of each flight, but hardware and software limitations can’t provide for the button to work currently on an android app, but the idea is the revolutionary base for the future integration of such systems.

Feature 4: track button: shows currently tracked flights, which have been booked by user.

Feature 5: previous flights: shows passed flights that the user has already booked and finished.

Feature 6: the admin can upload a new flight to the Future flights data base, and remove it from it.

Feature 7:

The data bases: Future flights DB, for upcoming flights

Track flights DB, for currently tracked flights for each user,

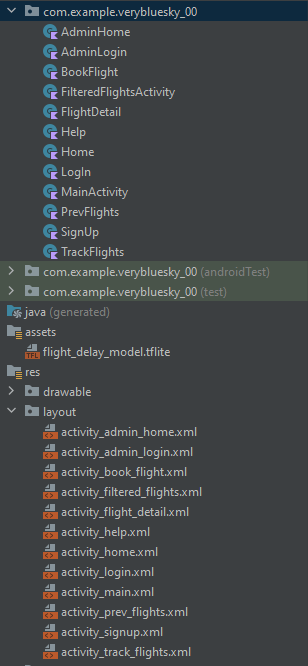
**Where each user has an email and password, that book the flight by that email, hence making it a user specific flight.**

Previous flights, also filtered by the user session, that end up showing the flights completed by said user.

Feature 8: the data set, and AI model, and algorithm that train a TensorFlow neural network, that learns and is able to predict when a flight might be delayed and how much of a delay chance it is.

Feature 9: the synchronization of the backend algorithms to keep track of the user session and user inputs and database integration, with the live databases.

Feature 10: the complexity vs simplicity of integration of the AI part into the application.  
\*here’s some snippets of code related to the features mentioned. And please find the schedule of who worked on what in section **“LIST OF TABLES**” as the last table



A screen shot of a computer program

Description automatically generated A computer screen shot of a program code

Description automatically generated A computer screen shot of a program code

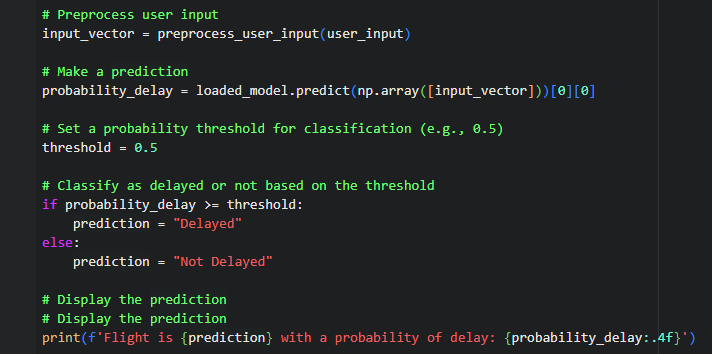
Description automatically generated A screen shot of a computer code

Description automatically generated A computer screen shot of a program

Description automatically generated A screen shot of a computer program

Description automatically generated





Result example: 

If we change the flight to another one:



## Project Software and Hardware Requirements

The app can work on android 11 minimum, and needs a Wi-Fi connection, hardware need is minimum, and software use is simple.

## Project Limitations

The integration process between the AI model from google collab aka the python code, can’t be transformed with the amount of data that it has, the model scale is too huge to run on android studio, or android software yet, and it needs to be uploaded to a server to be linked with the application, so when the app sends request to predict certain flight, it sends it to server, and server sends back the result of prediction then the app sorts the flights based on them. The server’s access was not issued for us as student, nor could our app transform the model without completely crashing the function or breaking the use of the model., so what is to be excluded is the model running in real time with the app, but the functions and the frame for that is in the code still, it’s just better not to press the button that activates this code

## Project Expected Output

A friendly user experience app, that can book, track and show previous flights of a certain user, and enhance the user experience with the algorithms and model idea that is projected in the app core.

## Project Schedule

## 

## Please note that this schedule only shows the final steps of implementing the project, the milestones of research and data gathering and user interaction survey’s were not mentioned in the table. Also the actual start of the project was on 5th of may 2023, that’s when we established the scope of the application, the goals , and started gathering info, and data for the project, as well as doing research and learning courses on how to write a Kotin code for the app, and a python code for the AI model

## Project, product, and schedule risks

the risk that the project takes longer than scheduled is the loss of motivation, and the loss track of data and making the already coded parts harder to read and re modify.in other words, work and finish when the inspiration hits, and don’t postpone.

**CHAPTER 2: Related Works**

## Introduction

## Our app has a solution for the users to check if their flight is going to be delayed or not which no other Flight app has, also it provides a user friendly interface that everybody can understand and use effectively, we also provided the users with a help page to make the use of the app more simple.

## Existing Systems

## Overall Problems of Existing Systems

## Some of the flight booking apps have a complex interface which may be overwhelming for the user and they might not want to use it again due to the complexity of its interface, also most of the existing apps have a little to none AI integration for the user to use, the existing AI models in any current flight booking app works as recommendation systems which recommends the upcoming flights for the user according to their search or their past flights.

## Overall Solution Approaches

## Our approaches to solve these problems are that we made a simple user interface that anyone can use and we also provided a help page to help the users with any confusion they have, and we provided it with a filter for the users to search for their desired flights, then they can sort the flights by its prices, and we provided the users with an AI model that will give them the delay probability of the flights.

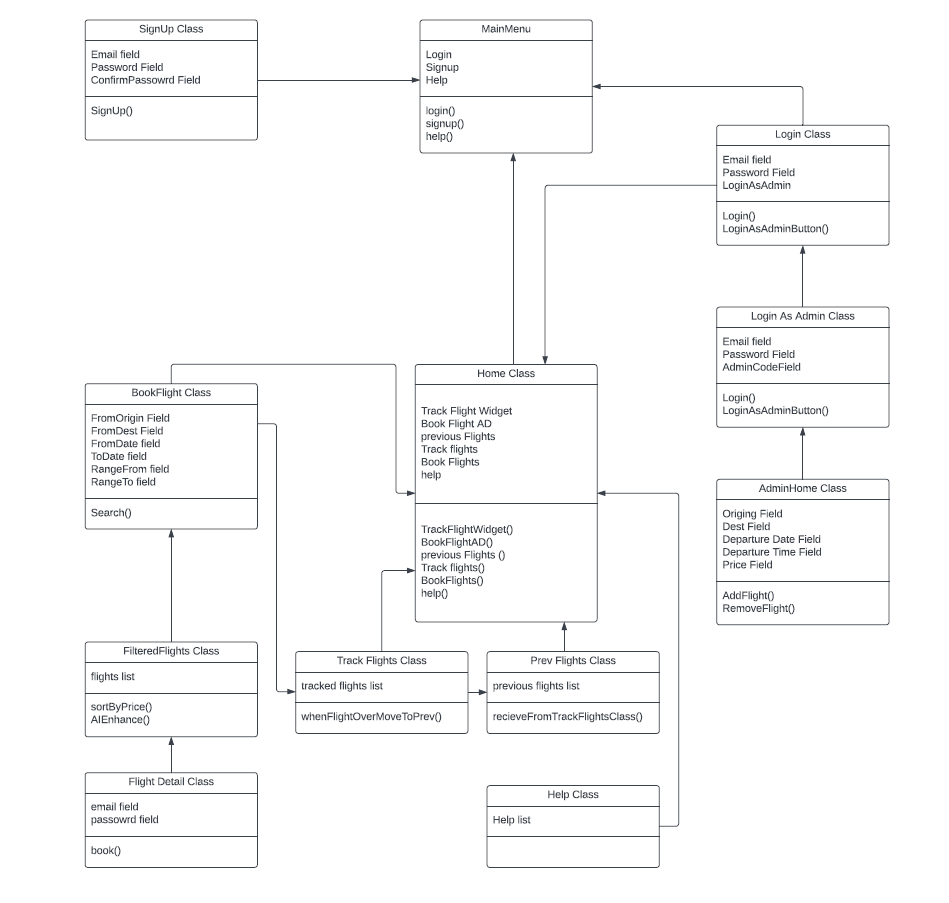
# CHAPTER 3: Methodology

## Overview

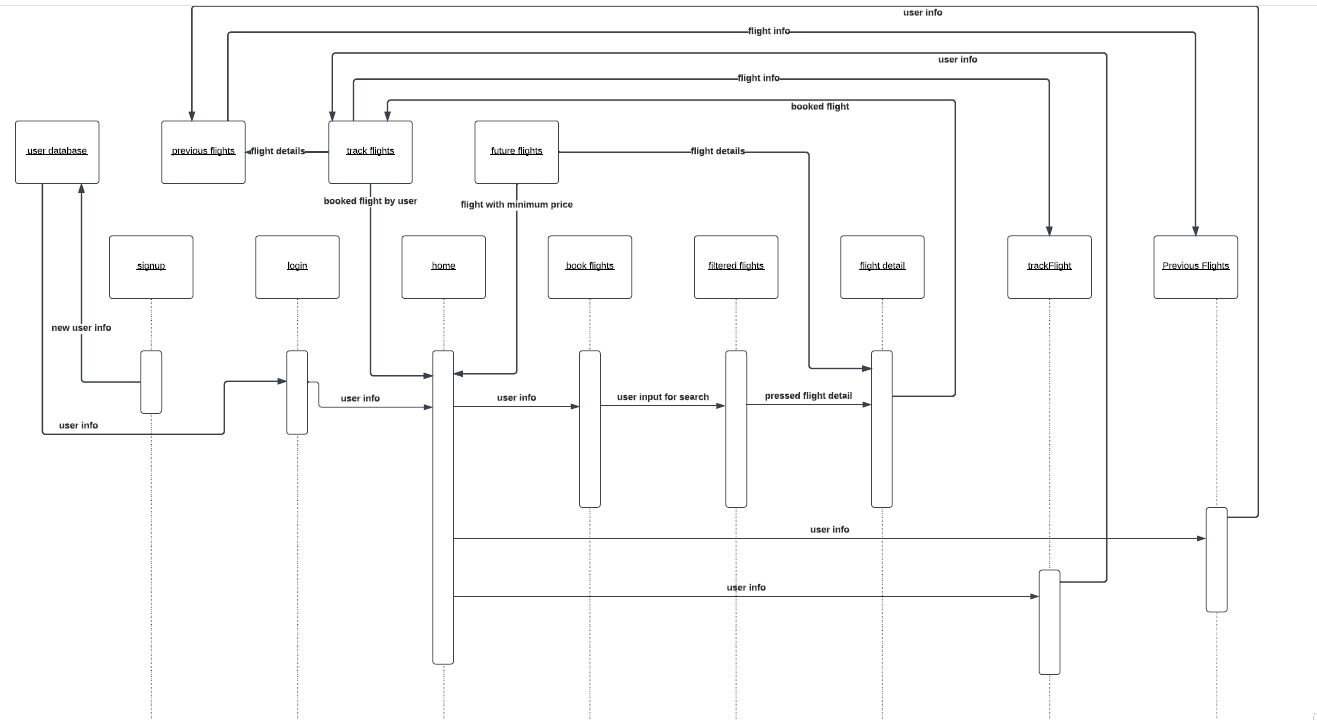
## Software architecture

### Logical view

* 1. Provide the software-architecture logical view for the major components as UML component diagram (or class diagram).



* 1. Provide the software-architecture process view for the major components as UML sequence diagram (or communication diagram).

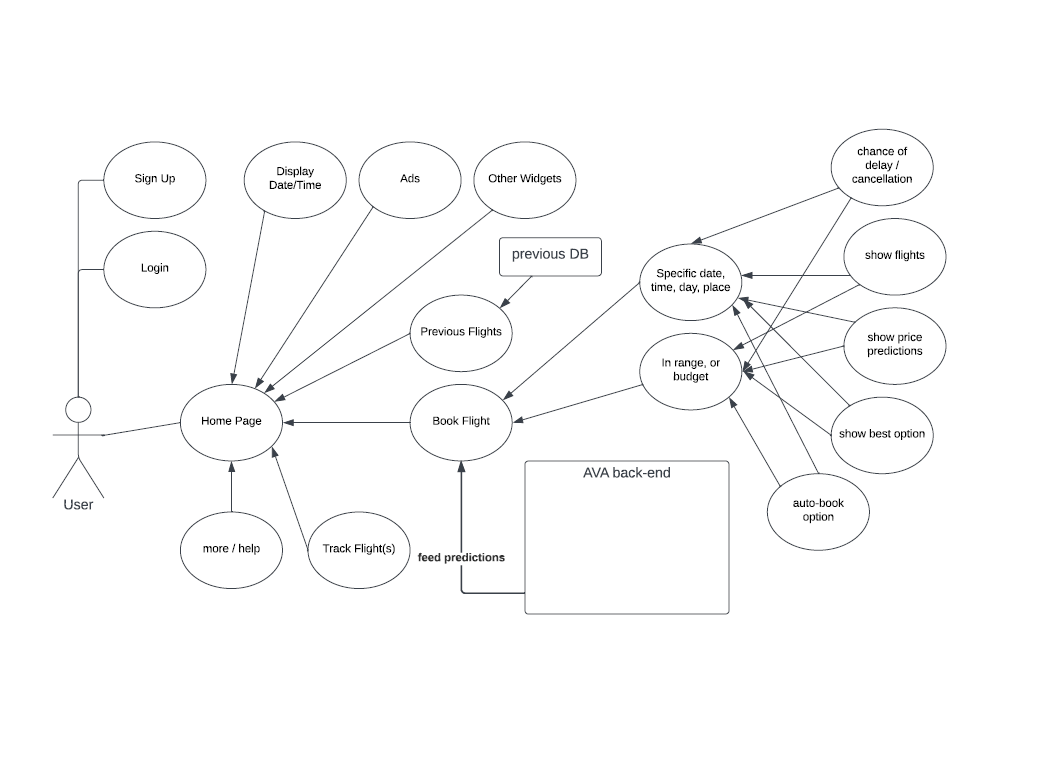


* 1. Provide the software-architecture physical view as UML deployment diagram.

***THE SCREENTSHOTS OF THE APP IN SECTION 3.7***

* + 1. Details of each component in a **separate section**.

## Software design

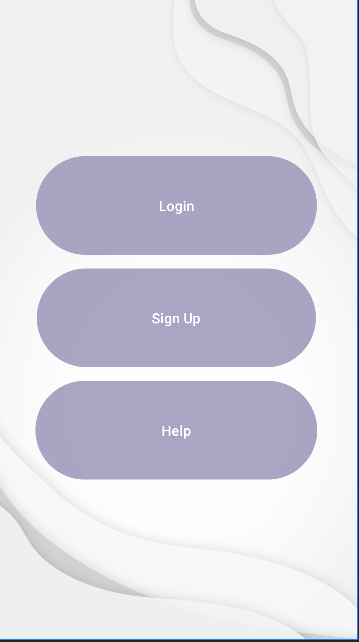
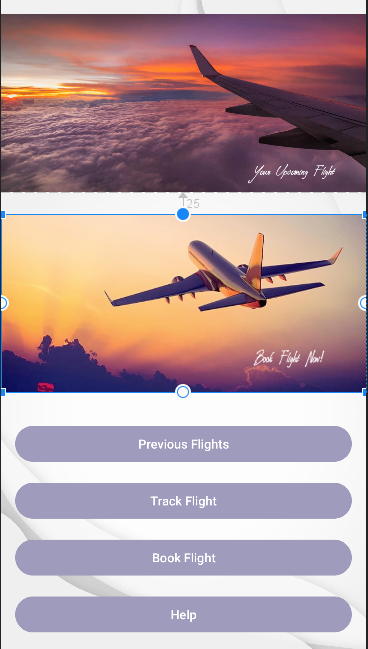
. 

A diagram of a software system

Description automatically generated with medium confidence

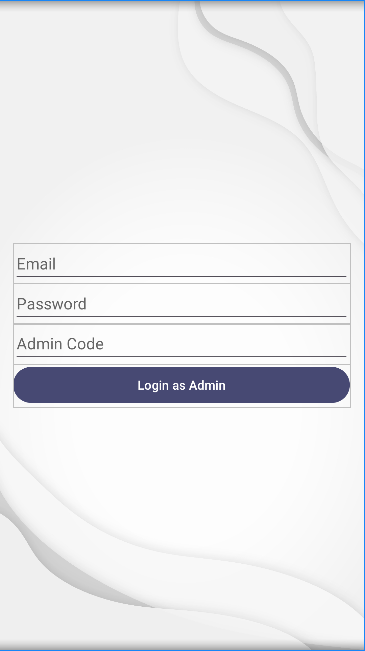
## User interface design (prototype)

* 1. Provide snapshots for the graphical user interface screens of the system.

  A screenshot of a login form

Description automatically generated A screenshot of a login form

Description automatically generated

 A screenshot of a flight

Description automatically generated

A screenshot of a computer

Description automatically generated

# CHAPTER 4: Experimental Results and Performance Evaluation

## Description of Implementation

The application was developed on android studio with Kotin and xml mainly, and the AI model is developed using python on google collab, the databases are on Google Firebase.

In android studio, we can connect an android phone and install the program as an app directly on the phone, and simply running the code or clicking the app will open the app and actually physically run it. As for the python part, it needs to be implemented directly on google collab, there it can run the model, install it and save the model for future use as well. The integration part which isn’t possible between the model and the app directly, can be resolved by owning and hosting a server to upload the model on (couldn’t be implemented right now for us as students).

## Programming language and technology

kotlin,android studio, firebase for database, tensorflow neural networks

here provided the libraries used:in import format

python:

import pandas as pd

import tensorflow as tf

from sklearn.model\_selection import train\_test\_split

from sklearn.impute import SimpleImputer

from google.colab import files

import matplotlib.pyplot as plt # Import matplotlib

kotlin:

import android.content.Intent

import android.os.Bundle

import android.view.View

import android.widget.Button

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import com.google.firebase.firestore.FirebaseFirestore

import java.text.SimpleDateFormat

import java.util.\*

import com.google.firebase.firestore.FirebaseFirestore

import com.google.firebase.auth.FirebaseAuth

import android.app.AlertDialog

import android.app.DatePickerDialog

import android.content.Intent

import android.os.Bundle

import android.view.View

import android.widget.DatePicker

import java.text.SimpleDateFormat

## part of implementation if possible

## *ALREADY PROVIDED THE KEY CODES IN Project Scope*

.

# CHAPTER 5: Conclusion and Future Works

**The conclusion** is that the app is perfectly running, a user can fully go into the app, and simply find how to book a flight of their desire, they can filter through flights, book and trace the flight when it’s upcoming in the dates, then when the flight date is passed, it gets transferred into the previous flights section which the user can also access.

The admin section can upload and remove flights from the database, so full admin functionality is needed.

The AI model implementation is literally the future of the project, since the next paragraph will explain the future of the project and the business model and scalability of the project.

**The Future** of the project is that if there’s a server to host the trained Neural Network TensorFlow AI model and we can directly connect the application to the database, improve the algorithms of faster fetching of predictions and gathering can also be implemented in the server based model.

Then after that we can gather more private information with authorization from local Aviation companies like Royal Jordanian, Emirates, Qatar Airways, and Jordan Aviation, and grow the model to host multiple companies, where it can have a network of subnetworks of neural networks, each neural network can learn each companies dataset, and provide predictions for each specific case.

Also the flights that are half company, half another company, like flying with RJ to berlin, then with S7 to Moscow, that needs to have a special kind of data set and a larger more scalable model, but our base model provides a perfect base for our goal, which again is:

Providing a pioneering step into the world of connecting ML and mobile apps, and in specifically transportation applications, more specifically aviation apps.

Where a future insight based on this model could go as high as saying this app will be the only app users need to book their flight anywhere they want, with the best algorithms to provide the best flight they can get.

**CHAPTER 8: References**

1. professor Ahmad Barghash’s slides for kotlin and xml development.
2. <https://www.kaggle.com/datasets/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018?select=2018.csv> and we used the csv file from 2018
3. https://github.com/tensorflow/tensorflow
4. <https://github.com/keras-team/keras>
5. <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html>
6. <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Future/>
7. <https://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html>
8. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www2.econ.iastate.edu/tesfatsi/DeepLearningInNeuralNetworksOverview.JSchmidhuber2015.pdf | https://www2.econ.iastate.edu/tesfatsi/DeepLearningInNeuralNetworksOverview.JSchmidhuber2015.pdf (Book)
9. <https://realpython.com/python-ai-neural-network/>
10. <https://www.activestate.com/resources/quick-reads/how-to-create-a-neural-network-in-python-with-and-without-keras/>
11. <https://www.analyticsvidhya.com/blog/2021/10/implementing-artificial-neural-networkclassification-in-python-from-scratch/>
12. Professor Samer Nofal’s Slides on AI, and professor’s Mahmoud al-sarayrah’s machine learning slides and assignments
13. <https://pythonprogramming.net/neural-networks-machine-learning-tutorial/>
14. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://aiindex.stanford.edu/wp-content/uploads/2022/03/2022-AI-Index-Report\_Master.pdf (standford book on tenserflow)
15. <https://developer.android.com/courses/android-basics-kotlin/course>
16. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron
17. Introduction to Machine Learning with Python" by Andreas C. Müller & Sarah Guido
18. Pattern Recognition and Machine Learning" by Christopher M. Bishop
19. Python for Data Analysis" by Wes McKinney
20. Neural Networks and Deep Learning: A Textbook" by Charu C. Aggarwal