Independent University **Bangladesh (IUB)**

IUB Academic Repository

Internship Reports Summer 2022

2022-08-14

OTA Management System of Tripey

Hossain, Arman

Independent University, Bangladesh

https://ar.iub.edu.bd/handle/11348/767

Downloaded from IUB Academic Repository



An Undergraduate Internship/Project on

OTA Management System of Tripey

Arman Hossain

ID- 1711118

Supervisor:

Raihan Bin Rafique

Lecturer

Department of Computer Science & Engineering

School of Engineering, Technology & Sciences

Independent University, Bangladesh (IUB)

Dissertation submitted in partial fulfilment for the degree of

Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

Thus, I hereby attest that I, Arman Hossain - 1711118, a student associated with Independent University Bangladesh, have completed the report, and submitted it in fulfillment of the condition for the Degree of Computer Science and Engineering from Independent University, Bangladesh (IUB). By giving credit where credit was due for the sources of the information used in my project and report, I adhered to the advice of my esteemed professor Raihan Bin Rafique.



Signature

Arman Hossain

Name

14.09.2022

Date

Acknowledgement

I am very grateful to Allah, the Almighty, for providing me with this opportunity and the ability to submit my internship report on time. Throughout my internship and the writing of this report, I would like to express my gratitude to my internal supervisor, Mr. Raihan Bin Rafique, Lecturer in the Department of Computer Science and Engineering at Independent University, Bangladesh (IUB), for all his invaluable advice, consistent guidance, support, and inspiration.

It has been an incredible honor to serve as an intern for "Tripey." I have received a lot of encouragement and support from the "Tripey" team. To my supervisor, I would want to convey my appreciation for giving of his time and knowledge, both of which were essential to the completion of this report. I want to thank everyone of my classmates. They have always been encouraging and on occasion have offered enlightening advice.

Finally, I want to thank my family. They have extended unconditional support without any restrictions. Their hopes for me and their faith in me kept me going, even on the toughest days.

Letter of Transmittal

Raihan Bin Rafique

Lecturer,

Department of Computer Science and Engineering,

School of Engineering and Computer Science

Independent University, Bangladesh

Subject: **Submission of Internship Report.**

Dear Sir,

I'm writing to let you know that I, Arman Hossain (ID: 1711118), would like to submit my internship report for the CSE 499 Internship Course for the summer 2022 semester. This report is based on the project I worked on at Tripey and my internship program. I tried to leverage my experience from my internship to the best of my ability to make my report as helpful as possible. I finished my internship program under Faayez Mohammad Neezamuddin's supervision.

I've tried my best to deliver a reliable report. However, it might not be faultless. I would greatly appreciate it if you could study this report and provide your wise judgment. I believe the report that follows will be adequate and win your approval.

Sincerely,

Arman Hossain

ID- 1711118

Department of Computer Science & Engineering

Independent University, Bangladesh

Evaluation Committee



.....

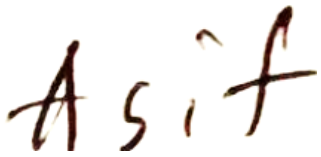
Name: Raihan Bin Rafique
Supervisor



.....

Name: Mohammad Noor Nabi

Internal Examiner-1 / Panel Member-1



.....

Name: Md. Asif Bin Khaled

Internal Examiner-2 / Panel Member-1



Abstract

This project includes a thorough analysis of the tour and travel management system. The main goals of this mobile app are to provide users with information about travel packages that have the best amenities and the most recent deals. It will be incredibly simple to search because you may get the information you need with only one tap.

A consumer illustrates how difficult and time-consuming it is to look for a variety of packages across important websites, in contact with travel agents, and through other means, all of which involve passive methods.

This mobile app will help travelers choose the best vacation package out of all the available options by providing them with useful details about the destinations they want to visit, including an image, a hotel's amenities, a Google map, a transportation option, and a description. This Online Travel Agency management system for tours and travel will be useful for the tourism industry.

Contents

Attestation ii
Acknowledgement iii
Letter of Transmittal iv
Evaluation Committee v
Abstract vi
Introduction 1
1.2 Objectives 2
1.3 Scopes 3
2.1 Relationship with Undergraduate Studies 4
3.1 Work Breakdown Structure 6
3.2 Process/Activity Wise Time Distribution 7
3.3 Gantt Chart 8
3.4 Estimated Costing 8
4.1 Methodology 9
5.1 Work Description 11
5.2.1 Rich Picture 12
5.3.1 Six Elements Analysis 13
5.3.2 ERD 20
5.3.3 Activity Diagram 21
5.4.1 Architecture 22
5.4.2 UML 24
5.5.1 Registering to the App 25
5.5.2 Login to the page 26
5.5.3 Landing Page: 27
5.5.4 Implementation of codes without front-end due to development: 28
5.6 Testing 35
6.1 Results & Analysis 37
7.1 Sustainability of the Project/Work 38
7.2 Social & Environmental Effects & Analysis 38
7.3 Addressing Ethics & Ethical Issues 39
8.1 Problems faced during this period 39
8.2 Solution of those Problems 40

9.1 Future Works 41
9.2 Conclusion 41
10 Bibliography 42
Consent Form 43

List of figures:

Figure 1- Rich Picture 12
Figure 2- ERD 20
Figure 3 - Login Activity Diagram 21
Figure 4- User Booking Activity Diagram 22
Figure 5 - Architecture of the system 23
Figure 6- UML diagram 24
Figure 7b - function which creates user when registration took place 25
Figure 7a - Register page 25
Figure 8b Login function 26
Figure 8a - Login Page 26
Figure 9b - Widget which displays the page 27
Figure 10a Landing page 27
Figure 11 Bus view 28
Figure 12 Email not verified 29
Figure 14 - SQL 31
Figure 15 - Exceptions 32
Figure 16 - Bus object 33
Figure 17 - homepage 34
Figure 18 - Test case 1 35
Figure 19 - Test case 2 35
Figure 20 - Test case 3 35
Figure 21 - Test case 4 36
Figure 22 - Test results 36

Chapter 1

Introduction

1.1 Overview/Background of the work

The primary goal of a tour and travel management system is to offer the best accommodations and travel options for a customer to reserve hotels, flights, and bus tickets for a trip. Our tour and travel management system were created to give users a search platform to locate the trip destinations of their choice.

Instead of giving customers and travel brokers the best travel services, this is done. To give a platform for discovery where a visitor may find their trip destinations based on their preferences, we have broadened our tours and travel management strategy. This strategy helps to encourage safe and interesting travel so that people can enjoy their vacations in their preferred locations.

Additionally, this method promotes the growth of tourism among other communities and cultures to enrich the traveler's adventure and sense of accomplishment. We develop this system to establish and broaden the framework of tourism, which offers beneficial possibilities for visitors and locals to engage and enhances knowledge of other cultures, traditional ways of life, traditional knowledge, and traditional morals. Additionally, this method offers a better approach to interact with different events.

1.2 Objectives

Bangladesh is a nation with many natural attractions, and tourism is its main industry. Young travellers are the key contributors to Bangladesh's rapidly growing domestic tourism market since they have a keen interest in traveling. We have around 7.5 million domestic tourists, mostly travels for business purposes and recreational purposes. While travelling these tourists often faces many hurdles from planning to booking their transportation and hotels. Trivial travel agencies mostly sell their pre-made trip plans, which often disappoints the tourists. To visit some places travel companion is a necessity but we do not have proper way of communication with local guides to help tourists to have safe and a beautiful experience.

OTA (Online Travel Agency) is a game changer as it gives users the freedom to book and plan for trips on the go. As all the required information are available in the platforms of the OTA users easily can find their packages and transportation tickets. But most of the OTA still are trivial travel agency tries to automate their current services. As we mentioned earlier travel agencies force tourists to buy packages and the tourists have not the freedom to extend and plan their trips as they want.

Tripey OTA management system is the solution of all the mentioned problems. Firstly, customers will have virtual no-cost travel assistant. Secondly, Tripey takes low commission from service providers. Thirdly, the booking system is created hassle -free to give customers the optimal experience. Gamified platform will allow users to get exciting offers and vouchers. Tripey also engages with the community to ensure community empowerment.

The four key services of Tripey:

- Transportation
- Accommodation
- Meal
- Guide

Users will login into the system and will have full access to all the mentioned services. Users do not have to take all the services together which makes the booking experience a lot easier. Its users wish to avail the service or services they want to use for their trip.

1.3 Scopes

An application called Tripey will assist in keeping track of tasks related to traveling and sight-seeing. Most individuals in this world enjoy traveling, whether it is over short distances or long ones. Some people enjoy traveling by plane, train, bus, or any other mode of transportation. The travel management tours

The system application is made for the travel agency, where customers can choose to book train or flight tickets to get to their destination.

- Machine learning can introduce so that the data collected from trips can be used to generate a recommendation system for the users
- With the help of AI, a new chatbot can be introduced for the users
- This kind of program can be expanded further to generate reviews. Pertaining to the needs of tourists.
- It was simple to locate the local landmarks, temples, and monuments.
- The developer may offer current locational information as well as Update the software to deliver better services.

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

My undergraduate degrees have given me the data and capacities that I want to foster this undertaking. On the off chance that these courses hadn't been finished prior to starting this task, it would have been more enthusiastically. Furthermore, the gathering, and individual undertakings I finished for my undergraduate courses were useful to me with this undertaking.

A few of the courses include:

- Data structures: which are a particular organization for coordinating, handling, recovering, and putting away information, are shrouded in CSC 203. Data structures come in both basic and complex structures, which are all made to coordinate Data for a specific use. Clients track down it easy to get to the Data they need and use it fittingly on account of Data structures. The getting sorted out of data is outlined by Data structures such that the two machines and individuals can more likely handle. As well as utilizing Data structures, it is vital to choose the best Data structure for every action. A terrible Data structure choice could prompt unfortunate run times or lethargic programming.

- Object-Oriented Programming: The underpinning of item situated writing computer programs is objects. Information designs or items with individual ascribes or properties are characterized in object-situated programming. Also, each article might have its own arrangement of rules or practices. Utilizing things that connect with each other, programming is made. OOP can be utilized to lessen how much work expected in assembling and plan applications. It tends to be applied, for example, while making flowcharts and diagrams. It supported the composition of the continuous framework configuration used to make this task.

- Database Management: A product program called a data set administration framework (DBMS) is made to get to, control, and oversee information put away in data sets. The actual information, the information design, field names, record structure, and _le structure are normally totally modified by a DBMS. It additionally determines rules for controlling and approving this information.

As data set organization methods advance, information base administration frameworks are based on unambiguous information taking care of standards. The principal information bases could oversee, exclusively pre-arranged pieces of information. The more modern frameworks of today can deal with many sorts of less organized information and associate them in additional mind-boggling ways. My most memorable course that showed me how to conceptualize and sort out a task was this one. I took in the basics of normal preparation and methodology strategies in the data set administration course, including the framework advancement life cycle, Six Element Analysis, Rich Picture, Requirement Analysis, Entity Relationship Diagram, Business Process Model, and some more.

- Web Applications and Internet: This course gives an intensive survey of web innovations and how they are utilized. We tended to principal subjects such OS and TCP/IP engineering, Internet Routing, IP tending to, and Domain Name System. My work was supported by conversations on normal programs, HTML and Cascading Style Sheets, HTTP, HTTPS, FTP, Client and Server-side prearranging, Scripting (Java Script, AJAX, XML) with Query libraries, and Web Servers (IIS, Apache). I gain information about making dynamic sites utilizing Django and MySQL and SQL server.

- System Analysis and Design: Planning, examination, plan, sending, and support are only a couple of the cycles that make up the orderly course of fostering a framework. We'll focus generally on System Analysis and System Design in this meeting. The techniques and devices utilized in the plan and examination of data frameworks are shrouded in this course. Frameworks and models, project the executives, apparatuses for sorting out framework necessities, information stream charts, choice tables, and choice trees, and frameworks examination: models for the frameworks advancement life cycle are a portion of the points covered. Brought together Modelling Language, use-case displaying, and object-arranged examination. Framework plan and execution incorporate practicality study, organized examination, framework model, application engineering, and client plan. information base plan, programming the board, front-end and back-end plan, as well as equipment determination. investigation of data frameworks case cases.

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

A work breakdown structure (WBS) organizes a project's tasks in a hierarchical manner (WBS). A project management technique called WBS helps to divide a challenging project into smaller, easier-to-manage activities or processes. The e-appointment system includes all the concept, design, development, maintenance, and closing procedures and operations. These processes are also broken down into smaller tasks and subtasks. In addition to a thorough sitemap, a project timeline, a risk analysis, and cost estimation, requirement analysis is a sub job. Two jobs are part of the design process. Design of systems and models that are development oriented. In the development-oriented model, we divide our duties into class diagrams, use case diagrams, and UML designs. For the system design, we have duties like rick images, ow charts, and system architecture. The two development processes for the project are frontend and backend. Four tasks for user acceptance Customer feedback, system evaluation, bug reporting, and bug remedies. Review the documentation and deployment protocols. Deliverable The deployment process, a closing activity, includes tasks linked to deploying the finished product and concluding adjustments. This WBS tries to manageably scale up a major project.

SL	Tasks
0	**Tripey OTA**
1	**Project Management**
1.1	Project Planning
1.2	Resource Allocation & Assign work
1.3	Project Kick off
2	**System Analysis**
2.1	Gather Requirment
2.2	Analysie Requirement
2.3	Prepare and Baseline SRS
3	**Desing**
3.1	Design HLD & LLD
3.2	Prepare and Baseline SDD
4	**Coding**
4.1	Module wise coding
4.2	Core review
4.3	Unit Testing
4.4	Draft release
5	**Testing**
5.1	Prepare test plan and test cases
5.2	Perform test and report bug
5.3	Bug fix
5.4	Re test
5.5	Release
6	**Implementation , Deployment & clouser**
6.1	Prepare the live enviroenment
6.2	Deploy the system
6.3	Perform UAT
6.4	Provide Traning
6.5	Prepare & Provide operational menual
6.6	Complete the system handover checklist
6.7	Take handover & sign off
6.8	Send project clouser note
6.9	Collect lesson learn

3.2 Process/Activity Wise Time Distribution

A software estimation method called Use-Case Points (UCP) is used to estimate the size of software using use cases. UCP and FPs share a similar idea.

Based on the following, a project's UCP count is determined:

- The quantity and degree of use cases in the system.
- The system's actor population's size and complexity.
  - Several non-functional criteria that are not expressed as use cases, such as portability, performance, and maintainability.
  - The setting in which the project will be produced (including the language used and the team's level of motivation, etc).

All use cases must be prepared with a purpose, at roughly the same level, and providing the same amount of detail to be used in estimation with UCPs. Therefore, the project team should make sure their use cases have been created with clear goals and at a thorough level before estimating. The use case is often finished in a single session, and after the objective is met, the user may go on to another task.

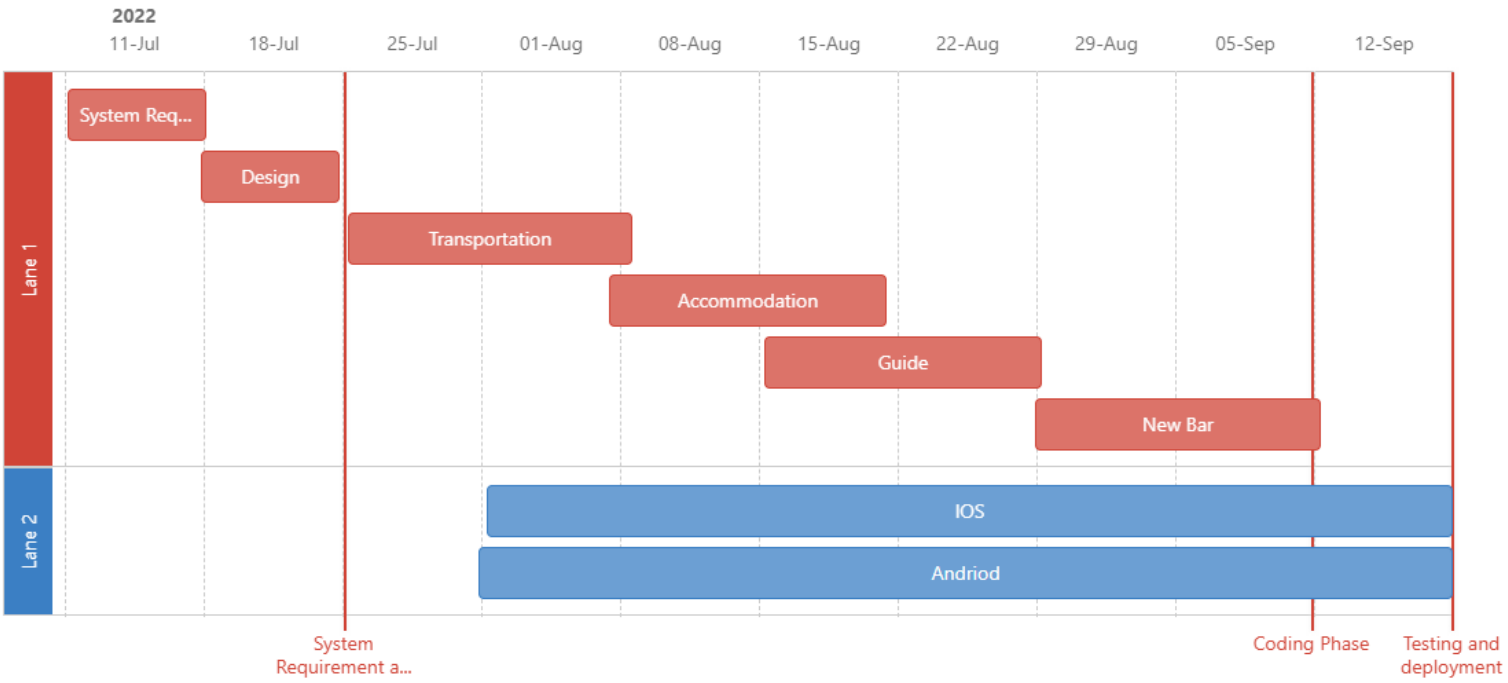
	Complexity Factors			
Module Name	High	Medium	Low	Funtion Points(FP)= No of module * Complexity Factor
login			3	3
Admin Pannel	10			10
User Registration		7		7
Transportaion Module	10			10
Accommodation Module	10			10
Add on functionalites	10			10
Dashboard	10			10
Total Funtion point				60
Total no of days (FP*1)				60
Total No of hours (FP*7)				420

1 Funtion Point(FP) =1 days
1 Work Day= 7 hours

Activity wise days of work distribution			
		Expected days	Considered Days
Project Management	10%	6	6
System Analysis	20%	12	12
Desing	15%	9	9
Coding	30%	18	18
Testign	15%	9	9
Implementation and Deployment	10%	6	6
	100%	60	60

3.3 Gantt Chart

The Gantt chart is one of the most popular and useful tools for showing activities (tasks or events) shown against time. It is commonly used in project management. On the left side of the chart is a list of the activities, and at the top is a suitable time scale. Each action is represented as a bar, with the beginning, middle, and end dates of the activity corresponding to the position and length of the bar. The progress of the project can be tracked using a Gantt chart.



3.4 Estimated Costing

My supervisor denied providing any information regarding costing.

Chapter 4

4.1 Methodology

Prototype Methodology was used for this project. A prototype is produced, tested, and revised until it is an acceptable prototype according to the prototyping paradigm of software development. It also lays the groundwork for the creation of the finished software or system. It functions well in circumstances where the project requirements are not fully understood. The process is iterative and based on trial and error between the client and developer. There are several sub-categories of prototype methodology and to be exact we used Incremental Prototype Methodology. When prototyping incrementally, the result is divided into numerous tiny iterations that are each produced separately. The various prototypes are eventually combined to create a single finished product. By using this technique, the user and the application development team may communicate more quickly.

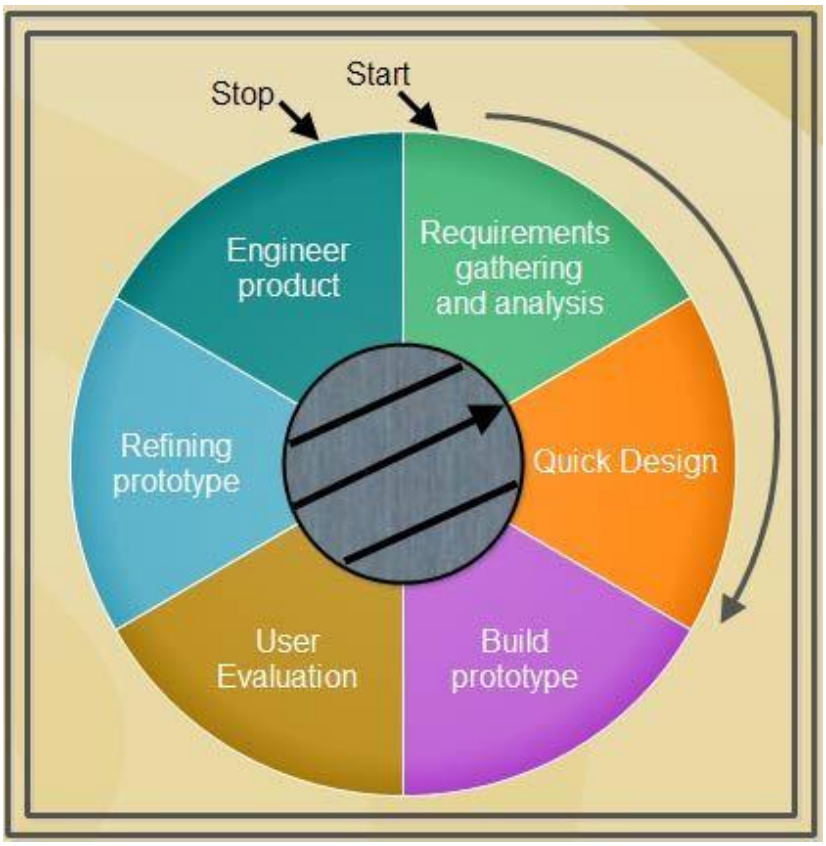


Figure: Prototype Methodology

Why Use this Methodology for this Project?

1.Gathering and analysing requirements: A prototype methodology starts with requirements analysis, during which the system requirements are stated in detail. To understand the needs of the system, the user is interviewed.

2. Quick design: A preliminary design or quick design for the system is made once the requirements are known. It is not a thorough design and simply comprises the crucial system components, giving the user a general overview of the system. An expedient design aids in the creation of the prototype.

3. Create a prototype: The initial prototype, which serves as a functioning model of the desired system, is created using the information acquired during quick design.

4. User evaluation: The suggested system is then given to the user for a complete assessment of the prototype to identify its advantages and disadvantages as well as what must be added or eliminated. Users' feedback and recommendations are gathered and given to the developer.

5. Changing the prototype: If the user is not satisfied with it after evaluating it, the present prototype is changed to better meet his needs. In other words, a new prototype is created using the user's additional information. The new prototype is assessed in the same way as the old prototype. The user's requirements will not be fully satisfied until this process is complete. A final system is produced based on the final prototype once the user is happy with the developed prototype.

6. Engineer the product: The user accepts the finished prototype if all requirements have been met. To preventing widespread failures and reducing downtime, the final system is carefully assessed before routine maintenance is performed on a regular basis.

Chapter 5

Body of the Project

5.1 Work Description

Tripey is trying to create a unique business model which will be focused on the customers to get best experience while travelling. To make this business model succeed, the most important factor is to create a mobile app for the travellers. Me as the new recruitment of the team Tripey, I was assigned with many tasks throughout my internship period.

I worked on:

- Authentication
- Creation of database using SQflite
- Worked with developers in testing
- I personally done the system analysis and UML

5.2 Requirement Analysis

5.2.1 Rich Picture

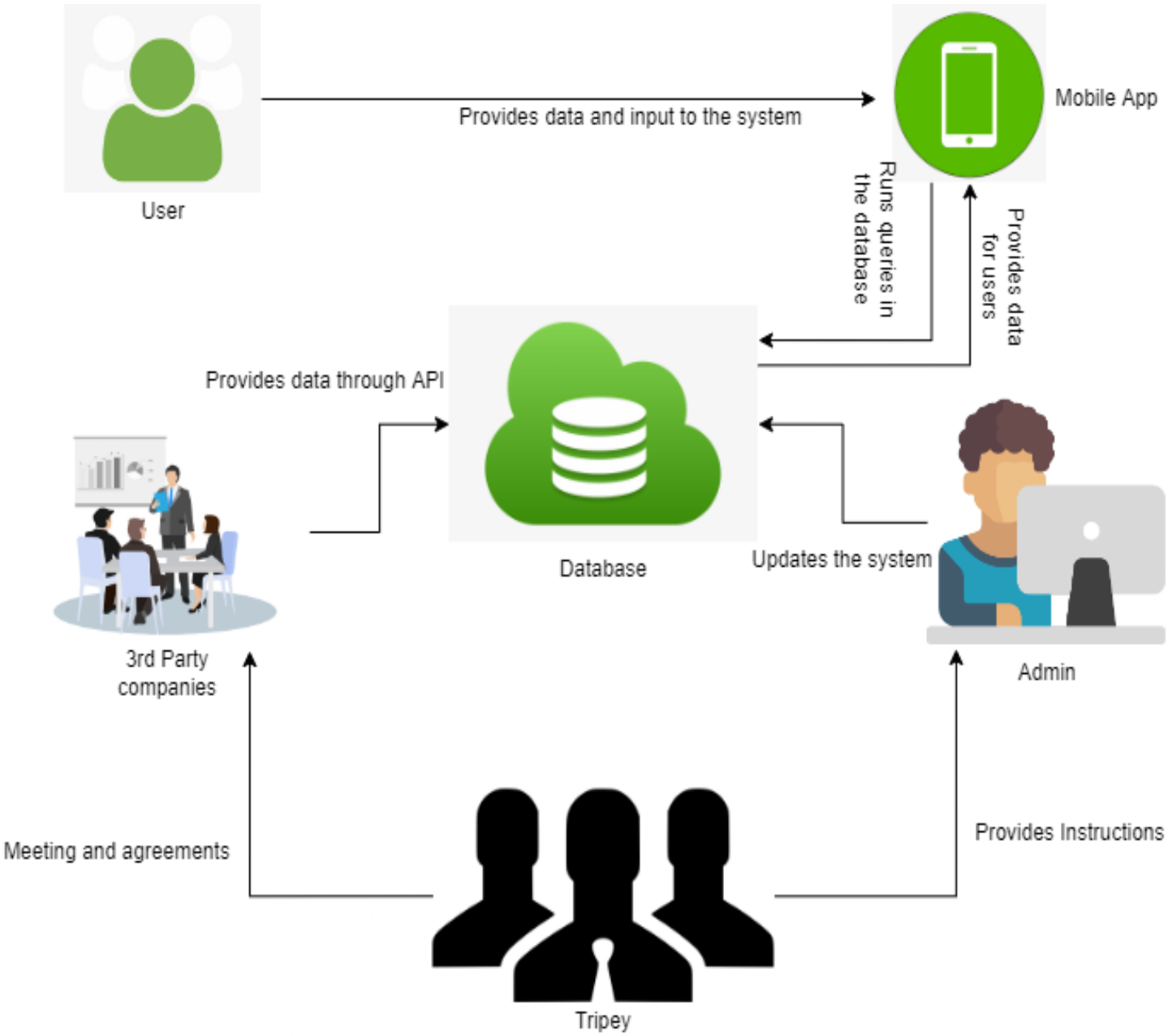


Figure 1- Rich Picture

5.3.1 Six Elements Analysis

0Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Provides data and input to the system	**User:** 1. User logs in to the system. 2. Select service which he/she wants to avail. 3. Fills information which will submitted to the system.	**N/A**	**Mobile:** 1. User will use their mobile devices to input data into the system.	**Tripey:** 1. User will use Tripey for this process	**SQLite:** 1. Flutter uses a plugin called SQflite which stores data.	**Internet:** Users will need internet to access to their account to avail services

Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Runs queries in the database	**N/A**	**N/A**	**Mobile:** 1. Tripey app will require a mobile to function	**Tripey:** 1. The queries will run internally to filter or fetch data from database	**SQLite:** 1. Flutter uses a plugin called SQflite which stores data. 2. Data from user will either be stored in database or will be used to fetch required data according to the user inputs	**Internet:** Users will need internet to access to their account to avail services

Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Provides data for users	**N/A**	**N/A**	**Mobile:** 1. Tripey app will require a mobile to function	**Tripey:** 1.The result will be generated internally and displayed into the screen	**SQLite:** 1. Flutter uses a plugin called SQflite which stores data. 2. Data from user will either be stored in database or will be used to fetch required data according to the user inputs	**Internet:** Users will need internet to access to their account to avail services

Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Updates the System	**Admin:** 1. Admin will login to the system. 2. Will make changes to the system according to the instructions provided by Tripey team	**N/A**	**Computer:** **1.** Admin will use computer to run applications which will allow him create changes in the system and deploy the system to the live server	**VSCode:** 1. VSCode is an editor which helps programmers to write code for flutter.	**SQLite:** 1. Flutter uses a plugin called SQflite which stores data.	**Internet:** Admin will need internet to access to their account to avail services

Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Provides Instructions	**Tripey Team:** 1. Tripey team provides instructions to the admin.	**N/A**	**Computer:** **1.** For communication or creating models for the admin **Mobile:** **1.** For communication	**N/A**	**N/A**	**Internet:** Tripey team will need internet to access to communicate or transfer files.

Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Meeting and Agreements	**Tripey Team:** 1. Tripey team arranges meeting with 3rd party companies. 2. Discusses business agendas and finalize the agreements between two organizations	**Documents:** 1. Many documents are provided to the 3rd party companies.	**Computer:** **1.** For communication and generating reports for the meeting **Mobile:** **1.** For communication	**N/A**	**N/A**	**Internet:** Tripey team will need internet to access to communicate or transfer files.

Process	Human	Non-Computing Hardware	Computer Hardware	Software	Database	Comm & Networking
Provides Data through API	**Tripey Team:** 1. Tripey team arranges meeting with 3rd party companies. 2. Discusses business agendas and finalize the agreements between two organizations	**Documents:** 1. Many documents are provided to the 3rd party companies.	**Computer:** **1.** For communication and generating reports for the meeting **Mobile:** **1.** For communication	**N/A**	**N/A**	**Internet:** Tripey team will need internet to access to communicate or transfer files.

5.3.2 ERD

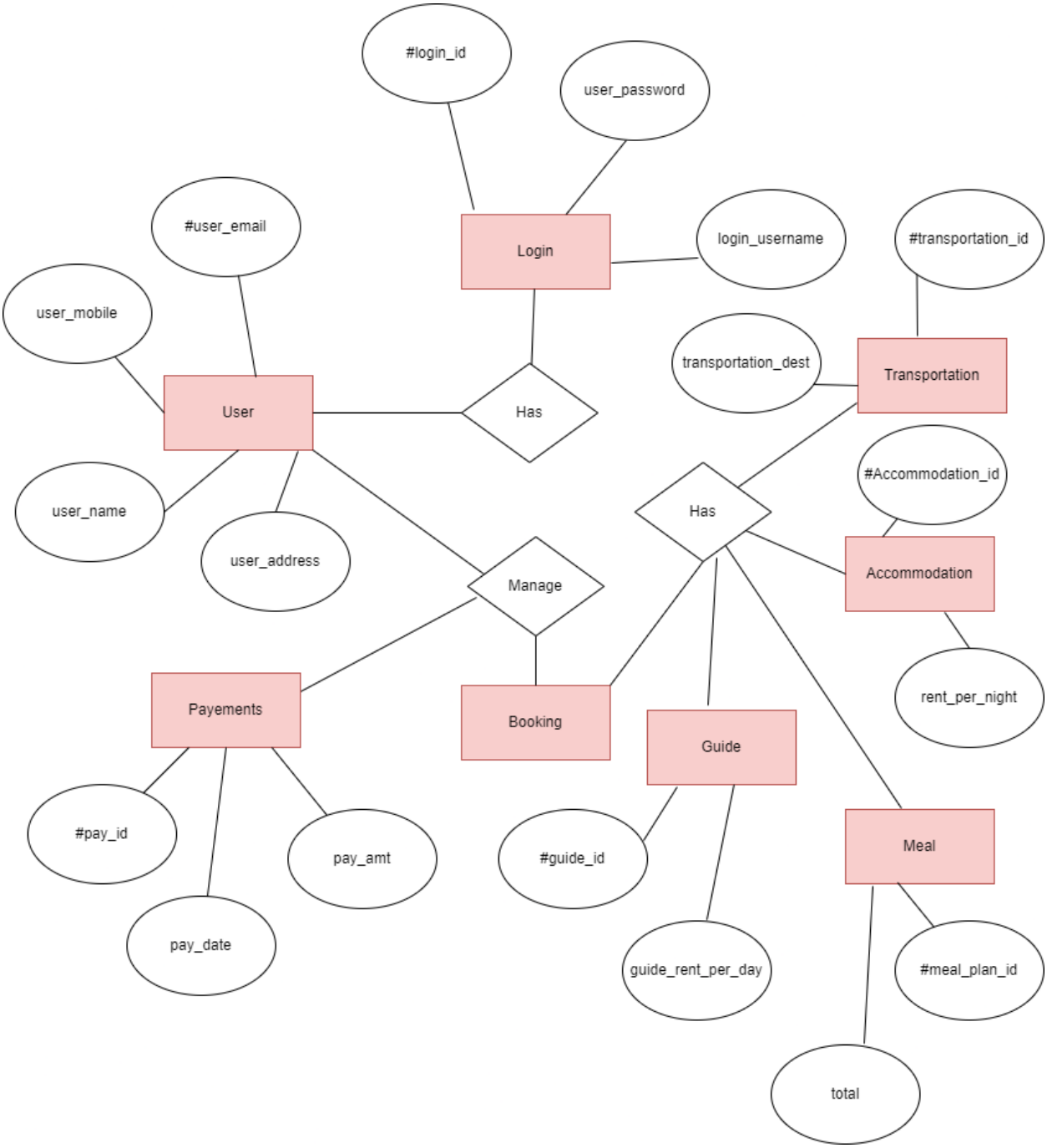


Figure 2- ERD

5.3.3 Activity Diagram

Activity Diagram - Login

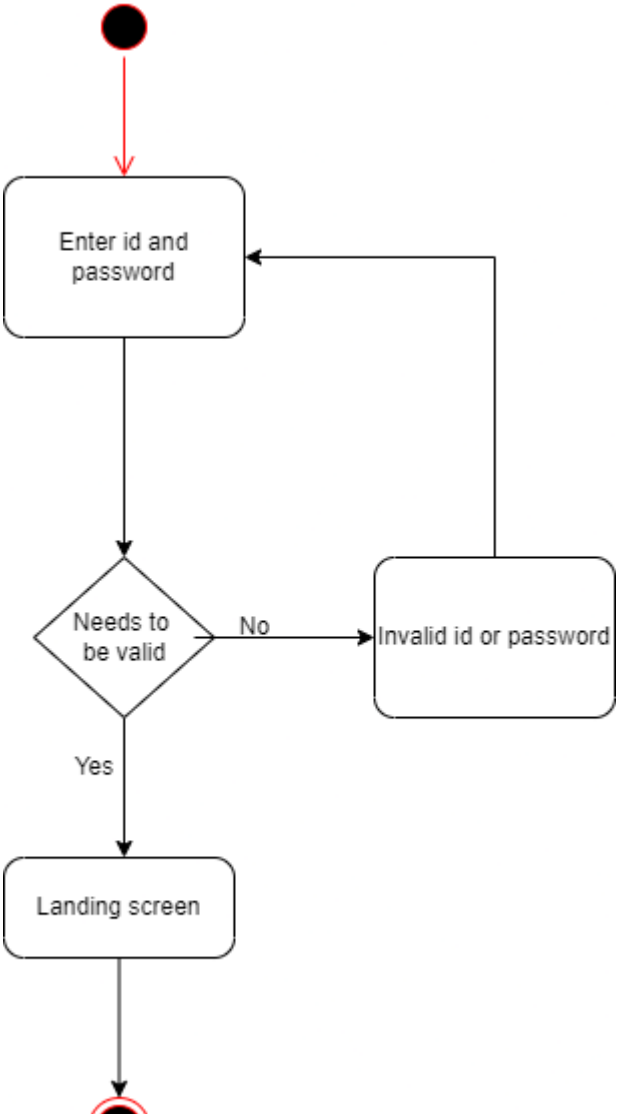
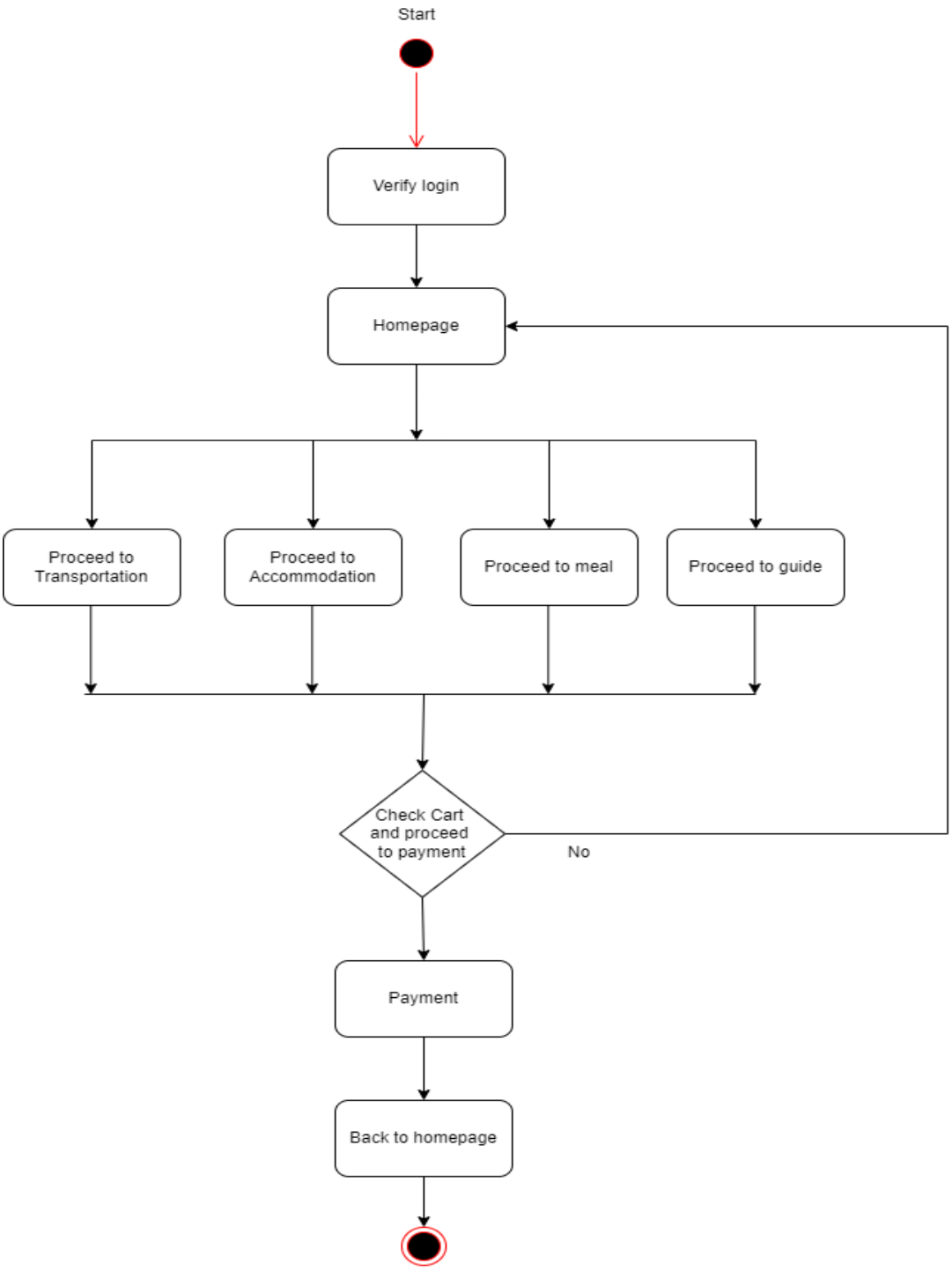


Figure 3 - Login Activity Diagram

Activity diagram- user booking



5.4.1

Figure 4- User Booking Activity Diagram

Architecture

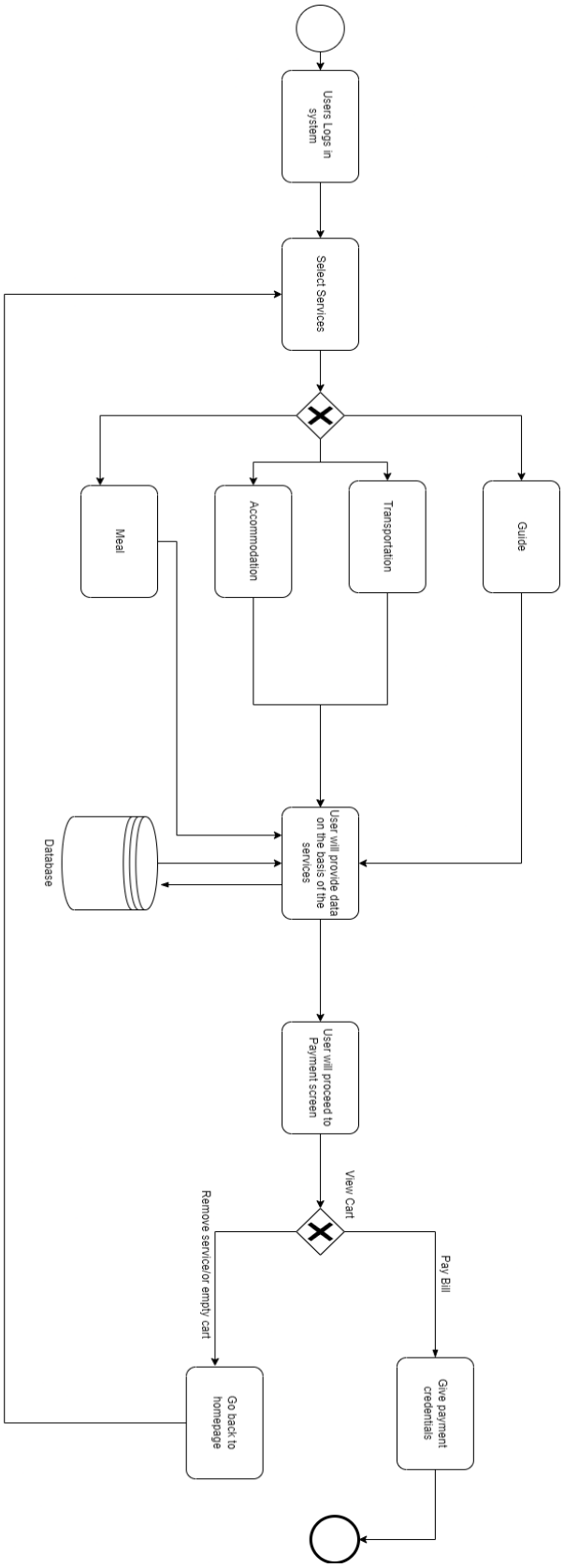


Figure 5 - Architecture of the system

5.4.2 UML

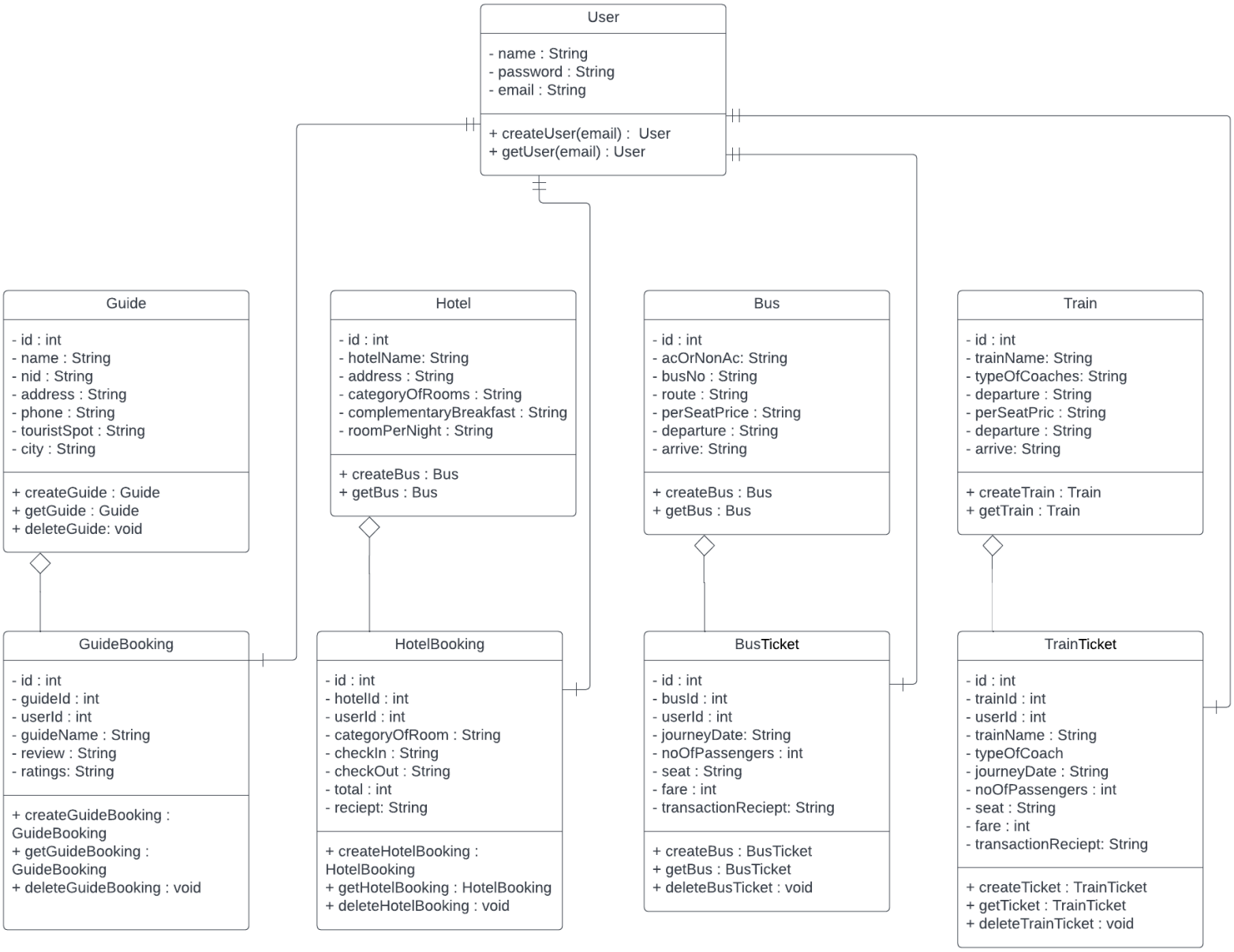


Figure 6- UML diagram

5.5 Implementation

5.5.1 Registering to the App

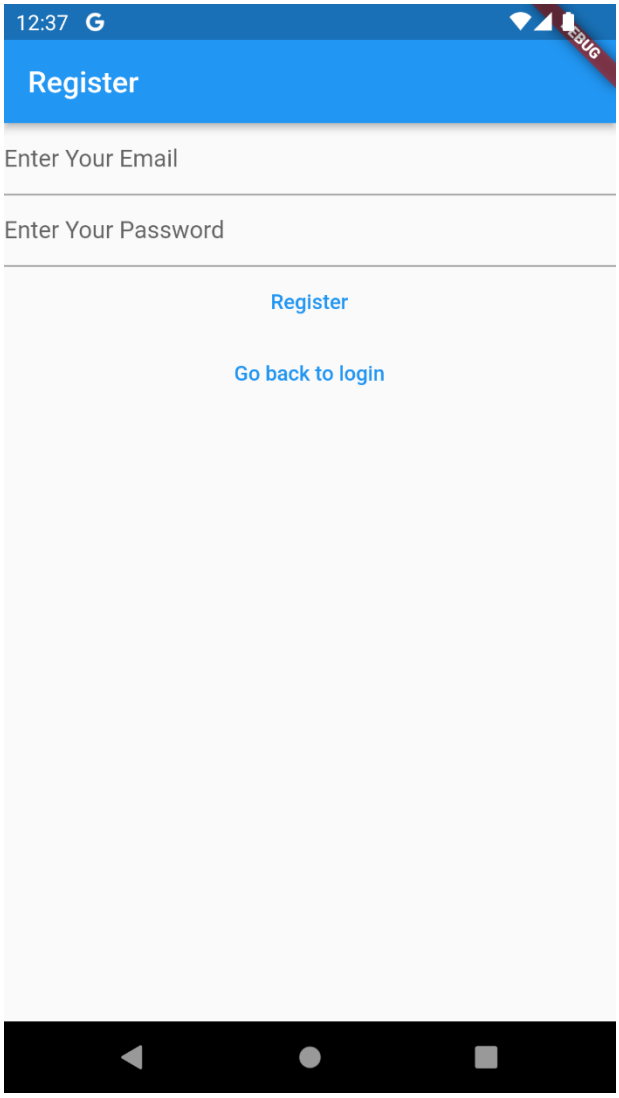


Figure 7a - Register page

```dart
class FirebaseAuthProvider implements AuthProvider {
  @override
  Future<AuthUser> createUser({
    required String email,
    required String password,
  }) async {
    try {
      await FirebaseAuth.instance.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );
      final user = currntUser;
      if (user != null) {
        return user;
      } else {
        throw UserNotFoundAuthException();
      }
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        throw WeakPasswordAuthException();
      } else if (e.code == 'email-already-in-use') {
        throw EmailAlreadyInUseAuthException();
      } else if (e.code == 'invalid-email') {
        throw InvalidEmailAuthException();
      } else {
        throw GenericAuthException();
      }
    } catch (_) {
      throw GenericAuthException();
    }
  }
```

Figure 7b - function which creates user when registration took place

5.5.2 Login to the page

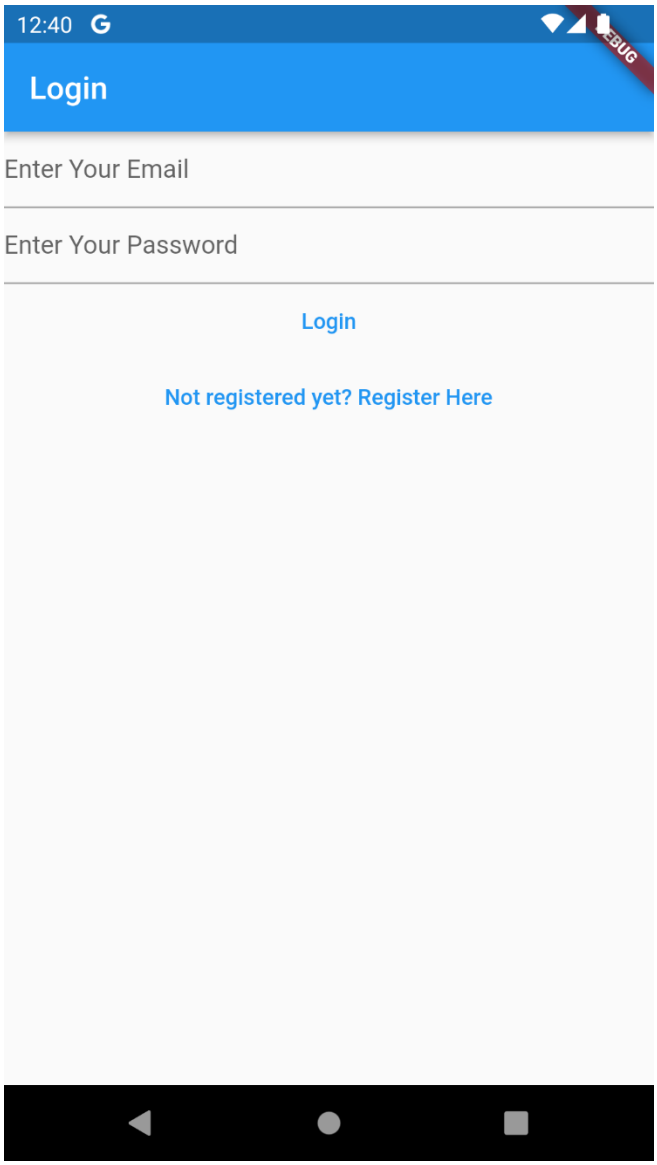


Figure 8a - Login Page

```dart
@override
Future<AuthUser> logIn({
  required String email,
  required String password,
}) async {
  try {
    FirebaseAuth.instance.signInWithEmailAndPassword(
      email: email,
      password: password,
    );
    final user = currntUser;
    if (user != null) {
      return user;
    } else {
      throw UserNotFoundAuthException();
    }
  } on FirebaseAuthException catch (e) {
    if (e.code == 'user-not-found') {
      throw UserNotFoundAuthException();
    } else if (e.code == 'wrong-password') {
      throw WrongPasswordAuthException();
    } else {
      throw GenericAuthException();
    }
  } catch (_) {
    throw GenericAuthException();
  }
}
```

Figure 8b Login function

5.5.3 Landing Page:

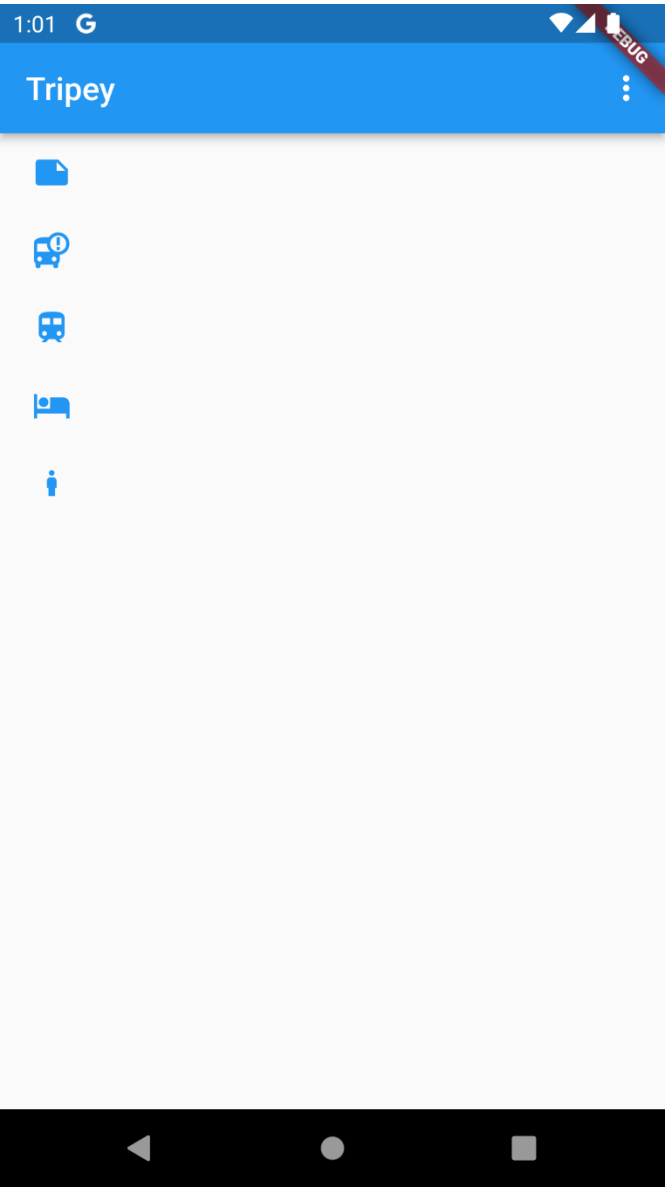


Figure 10a Landing page

```
        TextButton(
          onPressed: () {
            Navigator.of(context).pushNamed(busViewRoute);
          },
          child: const Icon(Icons.bus_alert),
        ), // TextButton
        TextButton(
          onPressed: () {
            Navigator.of(context).pushNamed(newTrainRoute);
          },
          child: const Icon(Icons.train),
        ), // TextButton
        TextButton(
          onPressed: () {
            Navigator.of(context).pushNamed(newHotelRoute);
          },
          child: const Icon(Icons.hotel),
        ), // TextButton
        TextButton(
          onPressed: () {
            Navigator.of(context).pushNamed(newGuideRoute);
          },
          child: const Icon(Icons.boy),
        ), // TextButton
      ],
    ); // Column
  default:
    return const CircularProgressIndicator();
 }
},
 // FutureBuilder
// Scaffold
```

Figure 9b - Widget which displays the page

5.5.4 Implementation of codes without front-end due to development:

Bus View:

```dart
import 'package:flutter/material.dart';

class NewBusView extends StatefulWidget {
  const NewBusView({Key? key}) : super(key: key);

  @override
  State<NewBusView> createState() => _NewBusViewState();
}

class _NewBusViewState extends State<NewBusView> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("New Bus"),
      ), // AppBar
      body: const Text("Write your new Bus"),
    ); // Scaffold
  }
}
```

Figure 11 Bus view

Email not verified view:

```dart
class VerifyEmailViewState extends State<VerifyEmailView> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Verify Email"),
      ), // AppBar
      body: Column(
        children: [
          const Text(
              "We have sent you an email verification. Please open it to your account."), // Text
          const Text(
              "If you have not recieve a verification email yet, press the button below"), // Text
          TextButton(
            onPressed: () async {
              await AuthService.firbase().sendEmailVerification();
            },
            child: const Text("Send Email Verification"),
          ), // TextButton
          TextButton(
            onPressed: () async {
              await AuthService.firbase().logOut();
              Navigator.of(context).pushNamedAndRemoveUntil(
                registerRoute,
                (route) => false,
              );
            },
            child: const Text("Restart"),
          ) // TextButton
        ],
```

Figure 12 Email not verified

Database creation using SQflite :

Creating a Bus class:

```
class DatabaseBus {
  final int id;
  final String acOrNonAc;
  final String busNo;
  final String route;
  final String perSeatPrice;
  final String departure;
  final String arrive;
  final bool isSyncedWithCloud;
  DatabaseBus({
    required this.id,
    required this.acOrNonAc,
    required this.busNo,
    required this.route,
    required this.perSeatPrice,
    required this.departure,
    required this.arrive,
    required this.isSyncedWithCloud,
  });

  DatabaseBus.fromRow(Map<String, Object?> map)
      : id = map[idColumn] as int,
        acOrNonAc = map[acOrNonAcColumn] as String,
        busNo = map[busNoColumn] as String,
        route = map[routeColumn] as String,
        perSeatPrice = map[perSeatPriceColumn] as String,
        departure = map[departureColumn] as String,
        arrive = map[arriveColumn] as String,
        isSyncedWithCloud =
            (map[isSyncedWithCloudColumn] as int) == 1 ? true : false;
```

Figure 13 - Creating Bus class

Some of the SQL:

```
const createHotelBookingTable = '''
      CREATE TABLE IF NOT EXISTS "hotel_booking" (
        "id"  INTEGER NOT NULL UNIQUE,
        "hotel_id"  INTEGER NOT NULL,
        "user_id" INTEGER NOT NULL,
        "category_of_room"  TEXT,
        "check_in"  TEXT,
        "check_out" TEXT,
        "total" INTEGER,
        "reciept" TEXT,
        "is_synced_with_cloud"  INTEGER NOT NULL,
        FOREIGN KEY("hotel_id") REFERENCES "hotel"("id"),
        FOREIGN KEY("user_id") REFERENCES "user"("id"),
        PRIMARY KEY("id" AUTOINCREMENT)
      );
      ''';

const createTrainTable = '''
      CREATE TABLE IF NOT EXISTS "train" (
        "id"  INTEGER NOT NULL,
        "train_name"  TEXT NOT NULL UNIQUE,
        "route" TEXT,
        "type_of_coaches" TEXT,
        "departure" TEXT,
        "arrive"  TEXT,
        "per_seat_price"  TEXT,
        "is_synced_with_cloud"  INTEGER NOT NULL DEFAULT 0,
        PRIMARY KEY("id" AUTOINCREMENT)
      );
      ''';
```

Figure 14 - SQL

Some of the exceptions:

```
class UnableToGetDocumentsDirectory implements Exception {}

class CouldNotDeleteUser implements Exception {}

class CouldNotFindUser implements Exception {}

class UserAlreadyExists implements Exception {}

class CouldNotDeleteNote implements Exception {}

class CouldNotFindNote implements Exception {}

class CouldNotUpdateNote implements Exception {}

class BusAlreadyExists implements Exception {}

class GuideAlreadyExists implements Exception {}

class HotelAlreadyExists implements Exception {}

class TrainAlreadyExists implements Exception {}

class CouldNotFindBus implements Exception {}

class CouldNotFindTrain implements Exception {}

class CouldNotFindGuide implements Exception {}

class CouldNotFindHotel implements Exception {}
```

Figure 15 - Exceptions

Creating a Bus object:

```
Future<DatabaseBus> createBus({required String busNo}) async {
  await _ensureDbIsOpen();
  final db = _getDatabaseOrThrow();
  final results = await db.query(
    busTable,
    limit: 1,
    where: 'bus_no = ?',
    whereArgs: [busNo],
  );
  if (results.isNotEmpty) {
    BusAlreadyExists();
  }
  const acOrNonAc = '';
  const route = '';
  const perSeatPrice = '';
  const departure = '';
  const arrive = '';
  final busId = await db.insert(busTable, {
    acOrNonAcColumn: acOrNonAc,
    busNoColumn: busNo,
    routeColumn: route,
    perSeatPriceColumn: perSeatPrice,
    departureColumn: departure,
    arriveColumn: arrive,
    isSyncedWithCloudColumn: 1,
  });
  final bus = DatabaseBus(
    id: busId,
    acOrNonAc: acOrNonAc,
    busNo: busNo,
    route: route,
    perSeatPrice: perSeatPrice,
    departure: departure,
    arrive: arrive,
    isSyncedWithCloud: true,
  );
  _bus.add(bus);
  _busStreamController.add(_bus);
  return bus;
}
```

Figure 16 - Bus object

Homepage:

```
class HomePage extends StatelessWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
      future: AuthService.firbase().initialize(),
      builder: (context, snapshot) {
        switch (snapshot.connectionState) {
          case ConnectionState.done:
            final user = AuthService.firbase().currntUser;
            if (user != null) {
              if (user.isEmailVerified) {
                return const LandingView();
              } else {
                return const VerifyEmailView();
              }
            } else {
              return const LoginView();
            }
          default:
            return const CircularProgressIndicator();
        }
      },
    ); // FutureBuilder
  }
}
```

Figure 17 - homepage

5.6 Testing

Test case 1:

```
Run | Debug
test('User should be null after initialization', () {
  expect(provider.currntUser, null);
});
```

Figure 18 - Test case 1

Test case 2:

```
Run | Debug
test('Should be able to log out and log in again', () async {
  await provider.logOut();
  await provider.logIn(
    email: 'email',
    password: 'password',
  );
  final user = provider.currntUser;
  expect(user, isNotNull);
});
```

Figure 19 - Test case 2

Test case 3:

```
Run | Debug
test(
  'Should be able to initialize in less than 2 seconds',
  () async {
    await provider.initialize();
    expect(provider.isInitialized, true);
  },
  timeout: const Timeout(Duration(seconds: 2)),
);
```

Figure 20 - Test case 3

Test case 4 :

```
Run | Debug
test('Cannot log out if not initialized', () {
  expect(
    provider.logOut(),
    throwsA(const TypeMatcher<NotInitializedException>()),
  );
});
```

Figure 21 - Test case 4

Test Result:

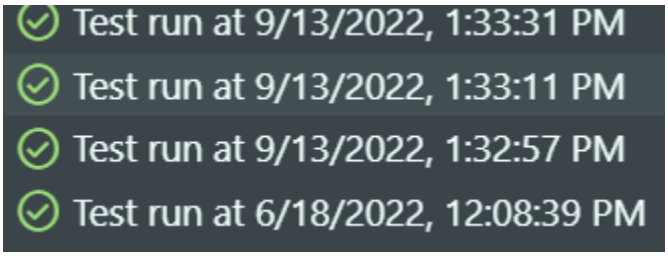


Figure 22 - Test results

All the above test cases were focused to find error in authentications. All the test cases were successful, and which confirms the chances of getting a bug in this portion is reduced.

Chapter 6

6.1 Results & Analysis

The research paper's results section should strive to condense the findings without attempting to analyse or interpret them. The analysis is described, and the findings are shown. Several problems were discovered when the application was tested. This little problem was solved by us. When these issues were resolved, test cases were documented.

As in the previous chapters we discussed about the testing, I am going to conclude the result based on the fact from those findings.

Firstly, we thought about few test cases, like scenarios where programs might crash, or it will be a very unpleasant experience for the users.

Secondly, we approached those issue and tried to design test cases.

Finally, we provide the test case with some inputs and give some clauses to check if the inputs are equivalent to the expected outcome than the test will pass.

We tested four scenarios, and all are related to authentication related bugs:

- User should be null after initialization
- User should be able to log out and log in again
- User will not be able to log out, if that user is not initialized
- User should initialize within two seconds

According to the test results we were successful at all the test cases which ensures us from getting any kind of error while logging in the system and logging out.

More test cases are being analysed now, in coming weeks those tests will be designed, and the system will be tested to make it deployable.

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

Project/Work Sustainability Engineering issues frequently have multiple solutions. The engineer's goal is to find the best answer he or she can use the tools at hand. Engineers are held formally accountable for the reliability and effectiveness of their creations. The goal is to handle a given problem as simply, safely, and economically as feasible. A product can fall into one of two categories to determine its sustainability:

Community Sustainability: It is anticipated that a strong farming foundation will be established following the creation and public release of the GIS-based Smart Farming System, and that this basis will give rise to a community of users who share common interests.

Financial Stability: The government is the system's primary target audience. The government will therefore be the system's primary user. Therefore, there won't be a problem with the project's financial expense.

7.2 Social & Environmental Effects & Analysis

Technology is developing very quickly. It affects people's life and changes the way they communicate, learn, and think. It has a big impact on society, and it's hard to imagine life without it now. Technology and society are interwoven, dependent on one another, and influence on one another.

Social Effect: All service providers that seek to reopen once the restrictions are relaxed in the previous epidemic situation have safety as their priority. The University can better organize the findings, making it easier for students and teachers to live their lives.

Effects on the environment: As a result of this project's successful completion through the system, the institution will need to produce less paper documentation because the records are already in the system. Less ink will be used if less paper is used.

7.3 Addressing Ethics & Ethical Issues

There are a few implicit laws and moral standards that must be watched out for when developing mobile apps in the world of smartphones, where there is so much information collection, hacking, and other criminality. The designers took care to ensure that all goals were taken into serious consideration and that there were no violations of behaviour. Some of them are as follows: The website, so to speak, gathers basic user information such a user's name and email address. The following administratively necessary data is also gathered: No additional data are gathered.

Philosophical inquiry into morality has its roots in classical Greece. It refers to a system of principles that have the capacity to fundamentally modify earlier choices and actions. The dynamics of moral judgment are said to be the focus of the philosophy subfield of ethics. Scientific research is limited by societal, community, and personal standards, just like all human endeavours. Rules for daily activities, protecting people's dignity, and sharing study findings are all part of research ethics.

Chapter 8

Lesson Learned

8.1 Problems faced during this period

Being an intern at "Tripey" has been a very eye-opening experience for me. I ran into a lot of problems; therefore, I was up for the challenge of working with a start-up team. Even though it was a difficult road, I was able to go beyond every challenge.

In addition to all of this, I have faced other challenges while working on this project. Here is a short list of some of these:

Workplace: I also ran into some difficulties there. Every day I had to attend meetings and get there on time. I had to make sure I abided by all the rules and laws that had to be strictly followed. I had to learn about their work environment in a very short period. I had to learn the dart programming language on top of my lack of knowledge of flutter and mobile app development. Getting acclimated to flutter widgets was the hardest part of the learning phase.

Adapting to New Technologies: Since this was my first time working with a team that uses flutter and mobile app development, I had to learn and adapt to new technologies quickly or risk losing the company. Even though mastering the skill set was doable, applying it to real-world situations became difficult.

Finding and Repairing Bugs: Finding bugs was usually challenging, and even when they were discovered, fixing them was sometimes quite tough. Some bugs were so difficult to fix that it would take a week to get rid of them.

8.2 Solution of those Problems

As an undergraduate student over the past four years, I've learned a lot. It helped me find most of the answers to the problem. The following is a list of those problems and their fixes:

- **Work Environment**: Through my academic coursework, I learned a lot about time management. I was able to plan my time well therefore and meet the project's strict deadlines while simultaneously attending other classes and studying for them.
- **Getting Used to New Technology**: I initially found it difficult to adapt to new technologies. But after a few days, I've grown accustomed to the entire procedure with the help of my employer and the web development team.
- **Locating and Resolving Bugs**: I frequently utilize the online forum "Stack overflow," and a senior developer helped me resolve several issues. I encountered a problem that I couldn't remedy, therefore I had to rewrite my code.

Chapter 9

Future Work & Conclusion

9.1 Future Works

The business intends to add numerous new capabilities to the current system as the project continues to develop to increase its functionality and user appeal. Before they can be developed, several Features still need to be refined. Gamification is one of the new elements that will be included in the system. so that consumers who use their reward points can take advantage of enticing deals.

Some of them are:

- A web application for the users
- Some key features like gamification
- Implementation of ML and Al to create chat bots, and recommendation systems.

9.2 Conclusion

During this time, I had a fantastic internship. I've made a lot of new friends, learnt a lot, and developed new abilities. I was able to work and gain professional experience in my field. An internship is a fantastic approach to gain this knowledge. During the internship, it was helpful to describe my capabilities and restrictions. This helped me to decide what knowledge and skills I should work on the next time. My internship taught me how to use flutter, which gave me access to the world of mobile development.

Thanks to Tripey's team for hiring me. Project manager of Tripey helped me a lot and my supervisor guidance thorough out the internship period helped me successfully acquire skills.

10 Bibliography

- What is entity relationship diagram (ERD)? (n.d.). Retrieved September 13, 2022, from https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/#:~:text=When%20present%20in%20an%20ERD,and%20many%2Dto%2Dmany.
- UML class diagram tutorial. (n.d.). Retrieved September 13, 2022, from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/
- বাংলাদেশ ট্যুরিজম বোর্ড. (n.d.). Retrieved September 13, 2022, from http://www.tourismboard.gov.bd/
- Feliu, C. (n.d.). *10 examples of Big Data, machine learning and AI in Travel Industry*. Business and Operational Analytics News. Retrieved September 13, 2022, from https://blog.datumize.com/10-examples-of-big-data-machine-learning-and-ai-in-travel-industry
- Nelson, O. (2022, July 21). *How AI and ML can solve business problems in tourism - chatbots, recommendation systems, and sentiment analysis*. neptune.ai. Retrieved September 13, 2022, from https://neptune.ai/blog/how-ai-and-ml-can-solve-business-problems-in-tourism-organization-chatbots-recommendation-systems-and-sentiment-analysis

An Undergraduate Internship/Project on

OTA Management System of Tripey

By

Arman Hossain

Student ID: **1711118**

Summer, 2022

Consent Form

The student modified the internship final report as per the recommendations made by his/her academic supervisor and/or panel members during final viva, and the department can use this version for archiving.



(Signature of the Supervisor)

Raihan Bin Rafique
Department of Computer Science & Engineering
Independent University, Bangladesh