Independent University **Bangladesh (IUB)**

IUB Academic Repository

Internship Reports Autumn 2022

2023-02-02

An Undergraduate Internship/Project on Mobile Application development of Tripey

Imam, Muzayed

Independent University, Bangladesh

https://ar.iub.edu.bd/handle/11348/706

Downloaded from IUB Academic Repository



An Undergraduate Internship/Project on Mobile Application development of Tripey

By

Muzayed Imam

Student ID: **1820223**

Autumn, 2022

Supervisor:
Sanzar Adnan Alam

Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

February 2, 2023

Dissertation submitted in partial fulfillment for the degree of
Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I hereby attest that I, **Muzayed Imam (1820223),** 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 supervisor, Sanzar Adnan Alam.



Signature

Muzayed Imam

Name

02/02/2023

Date

Acknowledgement

Firstly, all praise to the Almighty Allah, for Whom my internship has been completed without any major interruption.

Secondly, I express my immense gratitude to my supervisor Sanzar Adnan Alam sir, Lecturer in the Department of Computer Science and Engineering at Independent University, Bangladesh (IUB), for all his invaluable advice, consistent guidance, support, and inspiration. He guided us through every step throughout. And we are also thankful to Subrata Kumar Dey sir, senior lecturer in the Department of Computer Science and Engineering at Independent University, Bangladesh (IUB), who helped clear out any confusions we had.

It has been an incredible honor to work as an intern for ‘Tripey’. The Tripey team has been extremely encouraging and supportive. I would like to express my gratitude to my supervisor, Faayez Muzahammed Neezamuddin, for his time and knowledge, both of which were crucial to the completion of this report.

And, finally, we thank our parents without whose support it may not have been possible for us to pull through. Because of their constant support and prayers, we are now on the verge of graduating.

Letter of Transmittal

Sanzar Adnan Alam

Lecturer,

Department of Computer Science and Engineering, School of Engineering and Computer Science Independent University, Bangladesh

Subject: Submission of Internship Report for the completion of graduation.

Dear Sir,

I'm writing to let you know that I, Muzayed Imam (ID: 1820223), would like to submit my internship report for the CSE 499 Internship Course for the Autumn 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 informative as possible. I completed my internship program under the supervision of Faayez Mohammad Neezamuddin.

I would be grateful if you could review this report and offer your wise judgment. I believe the report that follows will be sufficient and will receive your approval.

Sincerely,

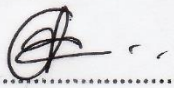
Muzayed Imam

ID- 1820223

Department of Computer Science & Engineering

Independent University, Bangladesh.

Evaluation Committee



.....

Name: Mr. Sanzar Adnan Alam
Supervisor



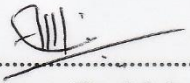
.....

Name: Ms. Ajmiri Sabrina Khan
Internal Examiner-1 / Panel Member-1



.....

Name: Mr. Sarwar Shahidi
Internal Examiner-2 / Panel Member-1



.....

Name: Dr. Mahady Hasan
Head, Department of Computer Science and Engineering

Abstract

In this report, I have included my information and experiences I have gained as an Intern at a startup OTA, Tripey. Here, I worked on a mobile application project which includes a thorough analysis of the tour and travel management system. The primary goals of this mobile application are to provide users with four essential travel criteria: transportation, accommodation, food, and tour guide services. The users of this application can avail any of the mentioned categories according to their needs.

A consumer demonstrates how difficult and time-consuming it is to search for a variety of packages across major websites, contact travel agents, and other means, all of which involve passive methods. This application provides users with the best deals and packages, as well as individual functions, from the aforementioned categories.

This mobile application is created using Flutter framework and Dart language in Visual Studio Code platform. Google Firebase and SQflite have been incorporated and implemented for authentication purposes and database.

Table of Contents

Attestation i
Acknowledgement ii
Letter of Transmittal iii
Evaluation Committee iv
Abstract v
Chapter 1: Introduction 1
1.1 Overview/Background of the work 1
1.2 Objectives 1
1.3 Scopes 2
Chapter 2: Literature review 3
2.1 Relationship with Undergraduate Studies 3
2.2 Related Works 3
Chapter 3: Project Management and Financing 5
3.1 Work Breakdown Structure 5
3.2 Process/Activity wise time distribution 6
3.3 Gantt Chart 8
3.4 Estimated Costing 8
Chapter 4: Methodology 9
Chapter 5: Body of the project 10
5.1 Work description 10
5.2 Requirement Analysis 11
5.2.1 Rich Picture 11
5.2.2 Functional Requirements 12
5.2.3 Non-Functional Requirements 14
5.3 System Analysis 15
5.3.1 Six Elements Analysis 15
5.3.2 Feasibility Analysis 16
5.3.3 Problem Solution Analysis 16
5.3.4 Effects and constraints 16
5.4 System Design 17
5.4.1 UML 17
5.4.2 Architecture 18
5.5 Implementation 19
5.5.1 Registration page 19
5.5.2 Login page 20

5.5.3 Implementation of codes without front-end: 21
5.5.4 Some error dialog boxes: 23
5.5.5 Database creation using SQflite : 24
5.5.6 Some of the SQL: 25
5.5.7 Some of the exceptions: 26
5.5.8 Homepage: 28
5.5.9 Routes: 29
5.6 Testing 29
Chapter 6: Results and Analysis 31
Chapter 7: Project as Engineering Problem Analysis 32
7.1 Sustainability of the Project/Work 32
7.2 Social & Environmental Effects & Analysis 32
7.3 Addressing Ethics & Ethical Issues 32
Chapter 8: Lesson Learned 34
8.1 Problems faced during this period 34
8.2 Solution of those Problems 34
Chapter 9: Future Work & Conclusion 35
9.1 Future Works 35
9.2 Conclusion 35
References 36

List of Figures

Figure 3.1: Work Breakdown Structure 5
Figure 3.2: Critical Path Method for Tripey app 7
Figure 3.3: Gantt Chart for Tripey app 8
Figure 4.1: Prototype Model 9
Figure 5.1: Rich Picture..... 11
Figure 5.2: UML Diagram..... 17
Figure 5.3: Architecture of the system..... 18
Figure 5.4: Register View..... 19
Figure 5.5: Login View 20
Figure 5.6: Bus view 21
Figure 5.7: Email not verified 22
Figure 5.8: Error Dialog Boxes 23
Figure 5.9: Database class for Bus 24
Figure 5.10: SQL..... 25
Figure 5.11: Exceptions..... 26
Figure 5.12: Bus Object 27
Figure 5.13: Homepage 28
Figure 5.14: Routes..... 29
Figure 5.15: Test case 1..... 29
Figure 5.16: Test case 2..... 29
Figure 5.17: Test case 3..... 30
Figure 5.18: Test case 4..... 30

List of tables

Table 3.1: Process-wise Time Distribution..... 6
Table 5.1: Functional requirement- User Registration 12
Table 5.2: Functional requirement- User Login 12
Table 5.3: Functional requirement- Transportation Booking 12
Table 5.4: Functional requirement- Hotel reservation.....13
Table 5.5: Functional requirement- Guide booking 13
Table 5.6: Functional requirement- Restaurant reservation.....13
Table 5.7: Functional requirement- View booking 14
Table 5.8: Six elements Analysis.....16

Chapter 1: Introduction

1.1 Overview/Background of the work

A tour and travel management system's primary goal is to provide customers with the best accommodations and transportation facilities when booking hotels and travel tickets for a trip. Tripey's tour and travel management system was created to provide users with a platform to make all of their vacation arrangements stress-free.

They expanded their tours and travel management strategy to provide a platform where visitors can enjoy the trip based on their preferences. This strategy promotes safe and enjoyable travel, as the basic requirements will be met.

Furthermore, this system encourages the growth of tourism in our country, which is beneficial for our economy and also enhances the knowledge of different cultures amongst people.

1.2 Objectives

Bangladesh is a country with many natural attractions and tourism is one of its main industries. Young travelers from all over the country play a big role in the rapidly growing domestic tourism market. Millions of people travel to different places around the country every day. Some might go for recreational purposes, some for business, some for work, some to their families, etc. While traveling people often undergo many difficulties such as transportation, accommodation, food, etc. and for this the Online Travel Agency (OTA), Tripey, has come up with a solution. While other travel agencies mostly sell their pre-planned trip facilities/ packages, which often leave the tourists in disappointment, Tripey is a game changer as it gives users the freedom to book and plan for trips on the go. Tripey gives its users full freedom on the 4 key services required or a tour. The 4 key services include:

- Transportation
- Accommodation
- Meals
- Tour Guides

While other agencies provide a pre-planned package system for the tour, you can plan your trip according to yourself using the Tripey mobile app. People can choose their own transportation, hotels, restaurants, or tour guides. They can even book any one of the four services mentioned. Tripey takes low commission from service providers and the booking system is created hassle - free to give customers of all ages an optimal experience. Gamified platform will allow users to get exciting offers, vouchers, and deals. 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. It is up to the users to choose the services they want to avail.

1.3 Scopes

Tripey, an app, will help users keep track of tasks related to their tours and travels. The system application is made for the travel agency, where customers can choose to book bus or train or flight tickets, hotels, book restaurants and tour guides for now. In future the company will broaden the functionalities o the application.

- Chat bots will be implemented using AI.
- Users can provide and view reviews about different locations, restaurants, hotels, and tour guides.
- Information about different local landmarks, temples, monuments, etc. will be provided.
- Machine learning can be used to generate a recommendation system for users based on data collected from trips.

Chapter 2: Literature review

2.1 Relationship with Undergraduate Studies

Some courses I took during my undergraduate studies provided me with the knowledge and skills I need to pursue this endeavor. The knowledge I gained from those courses helped to strengthen my understanding of application development and proved to be extremely useful.
A few of the courses include:

• **Data structures:** Data structures are useful for coordinating, handling, recovering, and storing information. They are classified as simple or complex, and they are all designed to coordinate data for a specific purpose. Data structures define how data is organized so that both machines and humans can more easily handle it. The 'Tripey app' involves many complex data structures which were implemented from the knowledge gained from this course.

• **Object-Oriented Programming**: In object-oriented programming, information designs or items with unique attributes or properties are defined. The course teaches about classes and its objects of programming. Furthermore, each article may have its own set of rules or practices. Programming is created by connecting various elements. OOP can be used to reduce the amount of work required in the assembly and planning of applications. Dart is an OOP language and this course helped me with the initial knowledge required in making the application.

• **Database Management:** A database management system (DBMS) is a software program designed to retrieve, control, and monitor information stored in data sets. This course also teaches how to design, plan, gather information and requirements, etc. The system designs (such as Rich picture, Relational Schema , Requirement analysis), database, CRUD operations and SQLs for 'Tripey App' were made possible because of this course.

• System Analysis and Design: Only a few of the cycles that make up the orderly course of facilitating a framework are planning, examination, planning, sending, and support. This course covers the techniques and tools used in the design. Flow charts, choice tables, and choice trees, and frameworks examination: models for the frameworks advancement life cycle are a few of the topics covered in this course. The designs for 'Tripey App', such as UML diagrams, activity diagrams, system architecture, etc. were made possible because of this course.

2.2 Related Works

There are applications similar to my project that are already available in the market. Some of them are listed below:

XpenseCo: XpenseCo is India's fastest growing travel aggregator, enabling their customers travel more efficiently by providing accurate pricing and inventory. They provide a self-service booking tool for business users as well as a booking platform for travel agents. XpenseCo's defined approval structure will

make the entire cycle and each transaction auditable. They are based in Gurgaon, NCR, India, and provide services in India and around the world. They constantly monitor the deals offered by various platforms and make them available to customers. The Flight & Hotel Booking feature allows the mobile app to book flights and hotels for both business and personal travel. Booking Deals feature provides the benefits of corporate fares for flights and hotels to help their customers to save money on personnel travel.

TripAdvisor: Every month, Tripadvisor, the world's largest travel guidance platform, assists hundreds of millions of people in becoming better travelers, from planning to booking to traveling. Travelers from all over the world use the Tripadvisor website and app to find out where to stay, what to do, and where to eat based on recommendations from other travelers. Travelers use Tripadvisor to find cheap hotels, book experiences, reserve tables at delicious restaurants, and discover new places to visit. Tripadvisor, a travel guidance company with 43 markets and 22 languages, makes trip planning simple regardless of the type of trip.

Roadtrippers: Roadtrippers transforms the way people discover the world around them by combining discovery, planning, booking, and navigation into an engaging and intuitive process. Customers can search and sort for nearby places to stay along their route, read reviews and photos, and book the best option. Using their general categories, sub-categories, and unique search, customers can find and save places they want to visit. Their saved locations sync with their phone, allowing them to use their maps app or Roadtrippers' in-app navigation.

GoZayaan: GoZayaan, Bangladesh's first online travel aggregator (OTA), is leading and revolutionizing this industry. GoZayaan is a travel company on a never-ending quest to improve the travel experiences through tech-driven innovations. GoZayaan has hotels, flights, buses, tours, and more to help plan trips. Their app also allows their users to personalize their trips on request.

Chapter 3: Project Management and Financing

3.1 Work Breakdown Structure

The Work Breakdown Structure (WBS) is a hierarchical structure that showcases how a project is divided into smaller segments. We created a work breakdown structure (WBS) for our project to ensure that our work is coordinated. WBS provides a visual representation of all scopes, risks, points of communication, responsibilities, costs, and ensures that essential outcomes are not overlooked. It is the ideal team tool for brainstorming and collaboration. We used a top-down approach in our work breakdown structure.

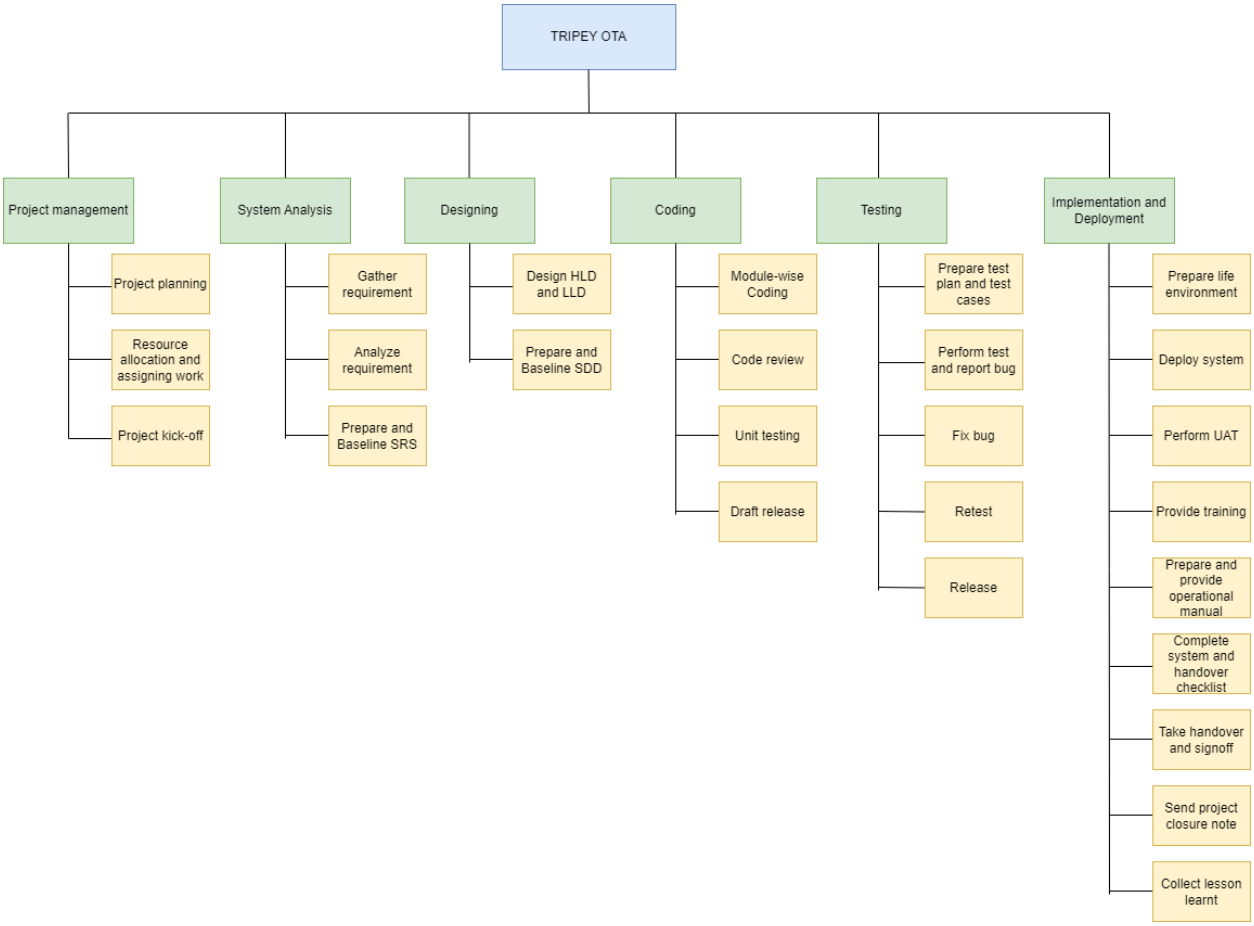


Figure 3.1: Work Breakdown Structure

3.2 Process/Activity wise time distribution

The estimated time required to successfully complete a project defines process-wise time distribution. This assists developers in creating a mind map of how efficiently they must work to meet deadlines. Time and time management are the most difficult challenges in correctly designing an application. As a result, the content must be fixed first, and development must be based on this context. Time management is the process of planning and regulating how much time to devote to various tasks. Good time management enables people to accomplish more in less time, reduces stress, and leads to professional success. To complete any project, time management is essential.

Index	Activity	Dependencies	Duration (Days)
A	System Requirements	None	9
B	Cost Analysis	A	2
C	Risk Analysis	A	2
D	Design	A	10
E	Coding	A,D	30
F	Testing	E	5
G	Deployment	E,F	5

Table 3.1: Process-wise Time Distribution

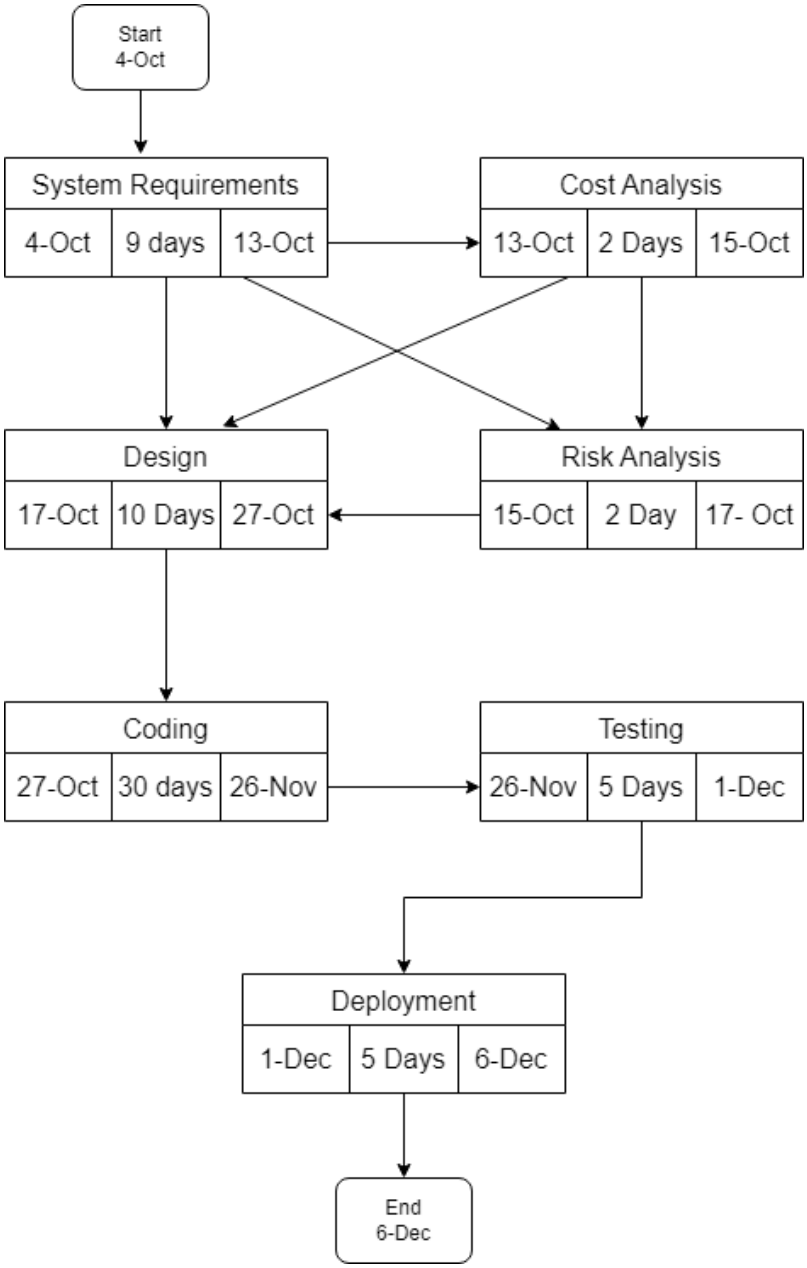


Figure 3.2: Critical Path Method for Tripey app

Here, we need 9 days for system requirements, 2 days for Risk analysis and 2 days for Cost and scheduling which add up to 13 days. After these parts, we need 10 days for designing which is considered 15.9% of the total works and 30 days for development/coding which is considered 47.6% of the whole project. After development, we need 5 days (7.9% of the works) for testing the project and 5 days (7.9% of the works) for Deployment of the project. Depending on these data, the critical path would be 63 days taking the longest path.

3.3 Gantt Chart

A Gantt chart is a project management tool that helps with the planning and scheduling of projects of all sizes, but they are especially useful for simplifying complex projects. Project management timelines and tasks are converted into a horizontal bar chart, which shows start and end dates, dependencies, scheduling, and deadlines, as well as how much of the task is completed per stage and who is the task owner. When the scope changes, this is useful for keeping tasks on track when there is a large team and multiple stakeholders.

Gantt charts are used for the following tasks:

• Create an initial project schedule, including who will do what, when, and how long it will take.

• Distribute resources - make sure everyone knows who is in charge of what.

• Make project adjustments – the initial plan will require numerous changes.

Monitoring and reporting progress keeps you on track.

• Manage and communicate the schedule - create clear visuals for stakeholders and participants.

• Show key events by displaying milestones.

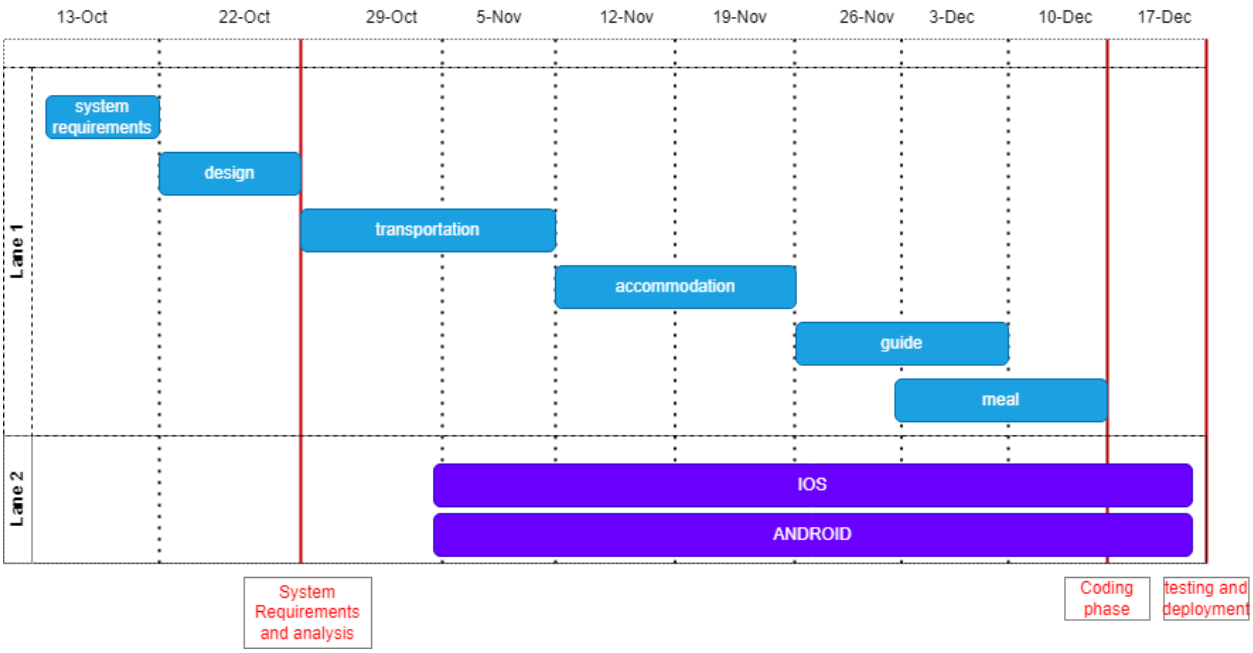


Figure 3.3: Gantt Chart for Tripey app

3.4 Estimated Costing

My supervisor denied providing any information regarding costing.

Chapter 4: Methodology

Prototype Methodology is being used for this project.

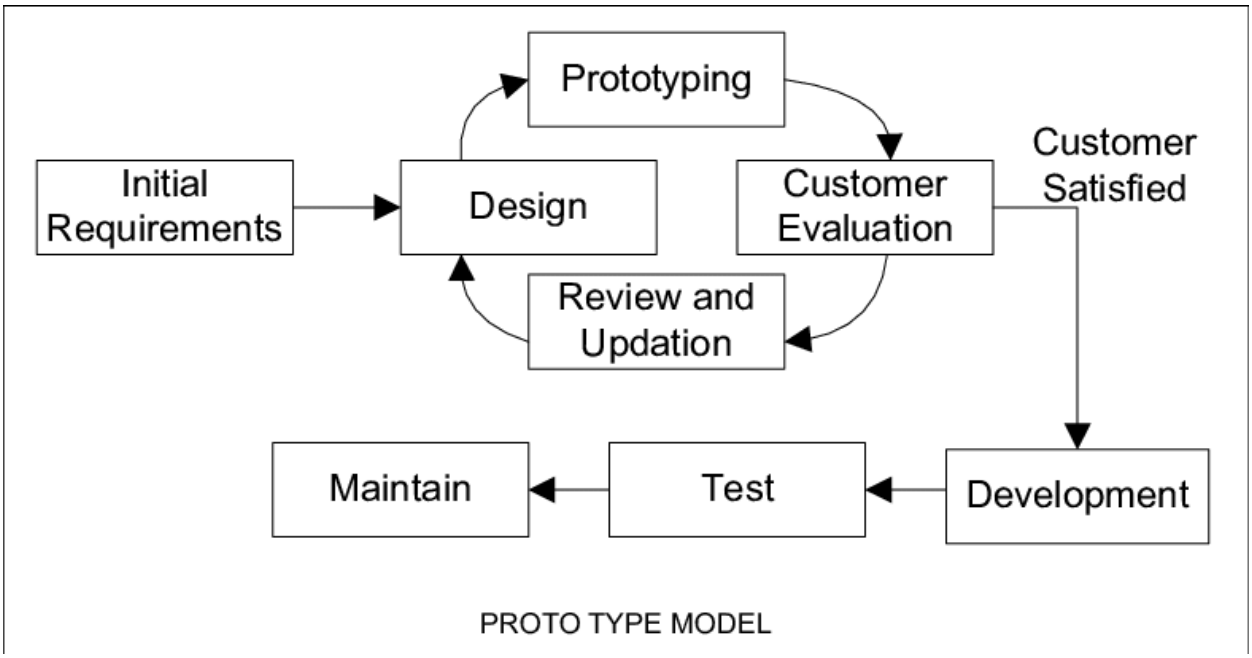


Figure 4.1: Prototype Model

Why Use this Methodology for this Project?

1. Initial requirements: A prototype methodology starts with requirements analysis, during which the system requirements are stated in detail.
2. Quick design: A preliminary design or quick design for the system is made, once the requirements are known, giving the client a general overview of the system.
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 client for a complete. Their feedback and recommendations are gathered.
5. Changing the prototype: With the gathered information and requirements the prototype is then changed again.
6. Engineer the product: If all requirements are met, 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

The application starts by the users tapping on the ‘Tripey App’ while being connected to the internet. The user logs/registers into the application. If the user is registering for the first time, he/she needs to verify their email. Then they will be landing on the home screen where they will meet the four functionalities of the application as stated in the previous chapters. Later they can proceed to the payment gateway and also check the selected items.

Over the course of my internship, I received some tasks as the team Tripey's new intern.
I worked on:

- Authentication using Google Firebase.
- Creation of database using Flutter SQlite (SQFlite).
- Unit testing for authentication.
- System Design.

5.2 Requirement Analysis

5.2.1 Rich Picture

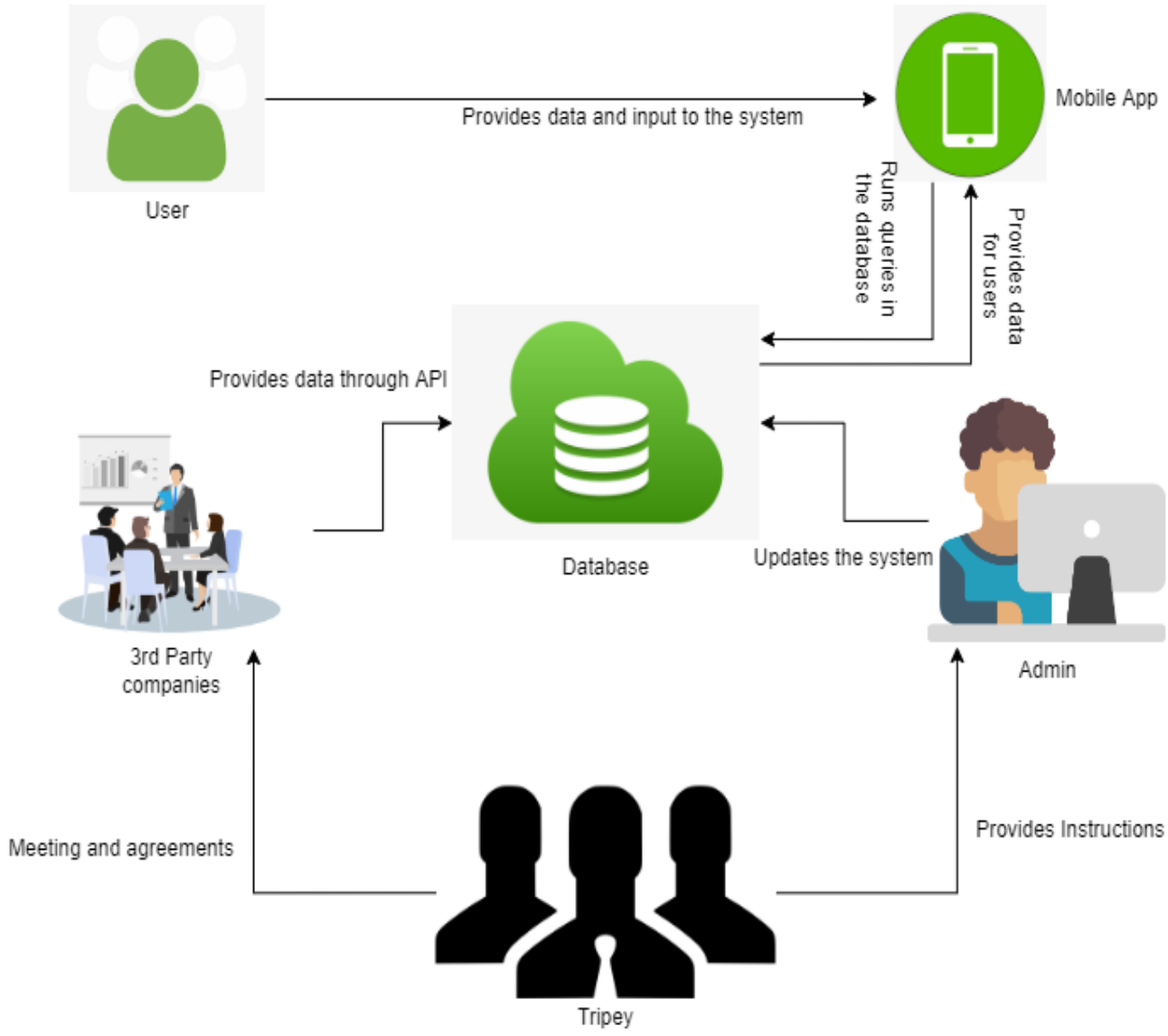


Figure 5.1: Rich Picture

5.2.2 Functional Requirements

User Registration		
Input: email, password	**Process:** create new user and save the sign-up details in the database	**Output:** new user created and saved to database
Pre-condition: User must go to registration page		
Post condition: User must go to their email for email verification process		

Table 5.1: Functional requirement- User Registration

User Login		
Input: email, password	**Process:** given credentials are matched with database	**Output:** User logs into app can now use its functionalities
Pre- condition: needs access of internet		
Post condition: user is redirected to main page		

Table 5.2: Functional requirement- User Login

Transportation Booking		
Input: destination, mode of transport, journey date, number of seats	**Process:** user is given choice of the selected transport with seats and timings. The selected option is forwarded and saved to database.	**Output:** user has now booked their transportation for their holiday
Pre- condition: user must have internet and must go to the transportation page		
Post condition: user will have the option to go to any other booking pages or straight to payment gateway		

Table 5.3: Functional requirement- Transportation Booking

Hotel Reservation		
Input: location, check-in and check-out dates, number of guests, number of rooms	**Process:** user will be given different options of hotels to choose from. The selected option is forwarded and saved to database.	**Output:** user has now reserved their hotel for their holiday
Pre- condition: user must have internet and must go to the hotel reservation page		
Post condition: user will have the option to go to any other booking pages or straight to payment gateway		

Table 5.4: Functional requirement- Hotel reservation

Guide Booking		
Input: location, date and time	**Process:** user will be provided with different guides, their reviews and their personal information. The selected option is forwarded and saved to database.	**Output:** user has now booked a guide for their tours
Pre- condition: user must have internet and must go to the guide booking page		
Post condition: user will have the option to go to any other booking pages or straight to payment gateway		

Table 5.5: Functional requirement- Guide booking

Restaurant Reservation		
Input: location, date, time, number of guests	**Process:** user will be provided with different restaurants and their reviews. The selected option is forwarded and saved to database.	**Output:** user has now reserved a restaurant for meal at any time for their holiday
Pre- condition: user must have internet and must go to the restaurant reservation page		
Post condition: user will have the option to go to any other booking pages or straight to payment gateway		

Table 5.6: Functional requirement- Restaurant reservation

View Bookings		
Input: no input is required	**Process:** all the bookings and reservations are called from database and shown in the form of table or list	**Output:** user can see their reservations or bookings
Pre-condition: user needs to book at least one feature		
Post condition: user will have the option to book again or redirect to main page		

Table 5.7: Functional requirement- View booking

5.2.3 Non-Functional Requirements

Performance: Response time of the application is very fast. The application starts within 2 seconds. The login, logout and initialization are very quick and smooth. Other functionalities such as bookings are also very quick and responsive.
Usability: The application is very direct and easy to use. People of all ages can easily adapt to the UI and functionalities of the application. It won't take time to learn to operate. Users can accomplish the goals without any help due to the simplicity of the interfaces.
Security: User's data and login credentials are stored in the database. All the authorization has been completed by google firebase and the passwords are also encrypted and saved securely. It is protected from unauthorized access to the app itself and its data.
The users will also have to verify their emails after registration to ensure authenticity of recipients.
Reliability and availability: The system has gone through performance tests and based on test results it can be said that system will run smoothly for a long time.

Portability: The application does not require a high specification phone but it does need internet access in order to run. The application is tested on various phones and all of them could run it very smoothly. Although earlier versions of android are no longer available, the lowest we could check it on was android version 9 and there were no problems found.

5.3 System Analysis

5.3.1 Six Elements Analysis

Process	System Roles					
	Human	**Non-computing Hardware**	**Computing Hardware**	**Software**	**database**	**Communication and network**
Login	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite), Firebase	Internet
Register	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite), Firebase	Internet
Email verification	User	N/A	Smartphone/ Computer	Tripey App, Chrome, firefox, Gmail app, etc	Flutter SQlite (SQFlite), Firebase	Internet
Home Screen	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite)	Internet
Transportation	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite)	Internet
Hotel booking	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite)	Internet
Guide booking	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite)	Internet
Restaurant Reservation	User	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite)	Internet
Payment	User	N/A	Smartphone	Tripey App, Bkash,etc	Flutter SQlite (SQFlite)	Internet
Provides data for users	N/A	N/A	Smartphone	Tripey App	Flutter SQlite (SQFlite)	Internet

Updates the System	Admin/ developers	N/A	Smartphone, Computer	VS Code, Flutter	Flutter SQlite (SQFlite)	Internet

Table 5.8: Six elements Analysis

5.3.2 Feasibility Analysis

A Feasibility analysis looks at the viability of an idea with an emphasis on identifying potential problems and if it is feasible to proceed with the project. Some of the feasibility studies are as follows:

Technical feasibility: The Tripey app is being developed using Flutter. The language used is Dart. Google Firebase is used for authentication and SQflite is used for the database. The development is done in Microsoft Visual Studio. All of the technologies used are very popular in the modern industry and are being used thoroughly with a growing community.
Operational feasibility: Tripey is developing the application with such a plan that it can be used with ease and people of all age can easily use it. Too much prior technical knowledge isn't required to use the application.

5.3.3 Problem Solution Analysis

Various problems were faced during this project. These were solved and implemented using some basic steps which are as follows:

- Identification: identifying the root of the problem
- Solution: coming up with accurate and precise solutions
- Implementation: implementing them and fixing the problems
- Testing: testing if the problems are solved. If they are not, then the steps are done again.

5.3.4 Effects and constraints

The application is still under development. The APIs for the transportation systems and hotels are yet to be incorporated. The application's payment process is dependent on online/mobile banking services which are yet to be implemented. Some time is needed for the application to be fully running and launched into the stores.

5.4 System Design

5.4.1 UML

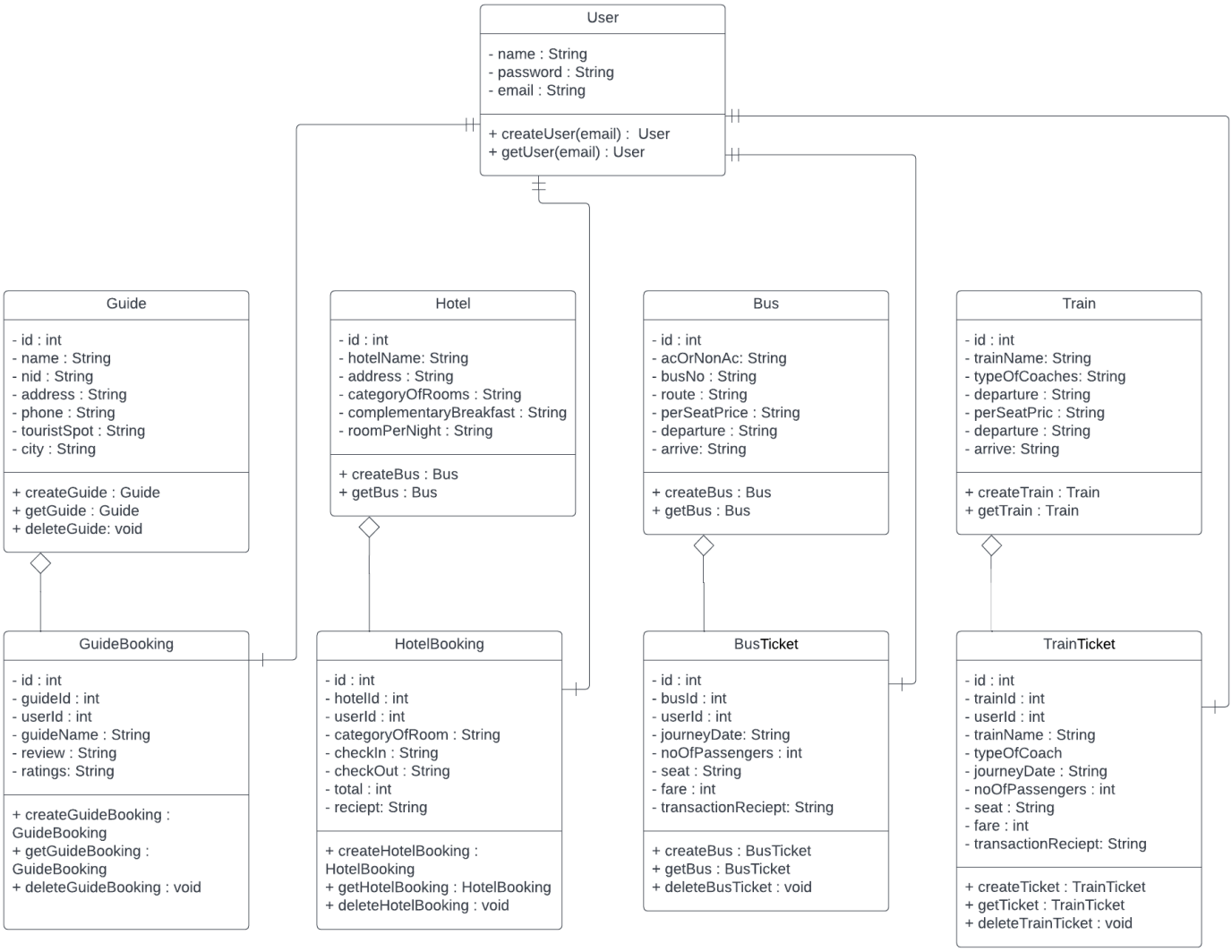


Figure 5.2: UML Diagram

5.4.2 Architecture

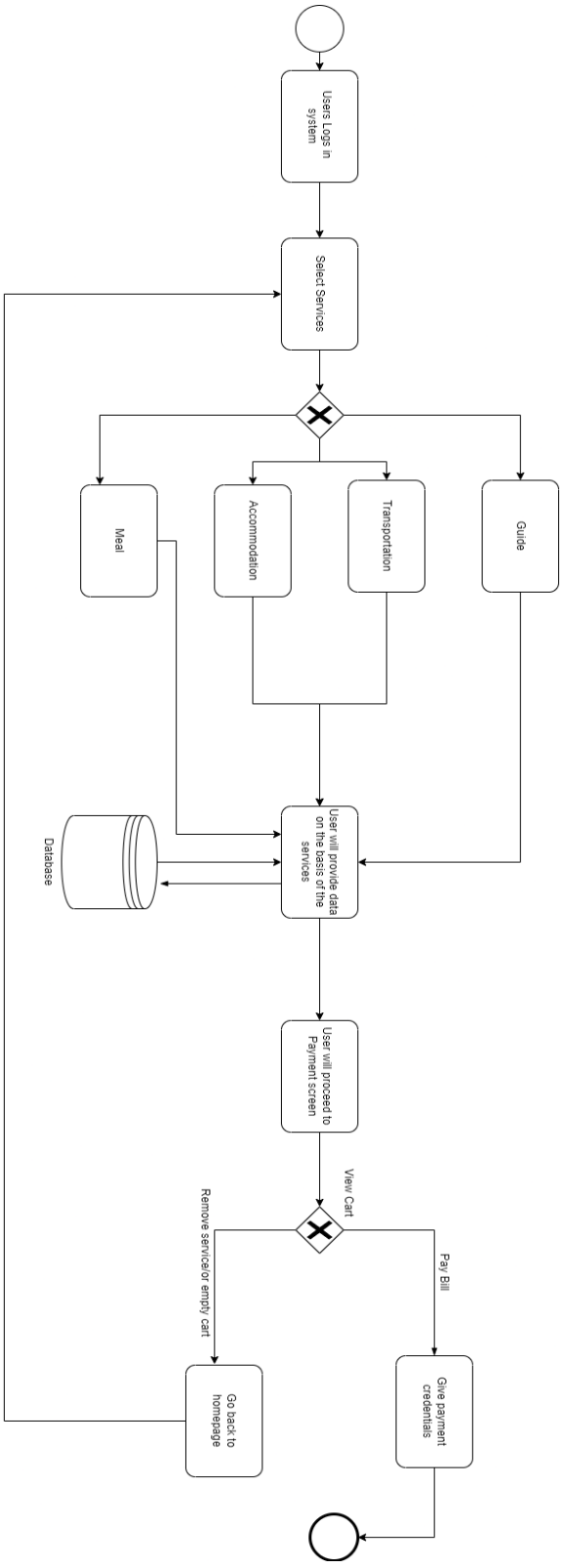


Figure 5.3: Architecture of the system

5.5 Implementation

5.5.1 Registration page

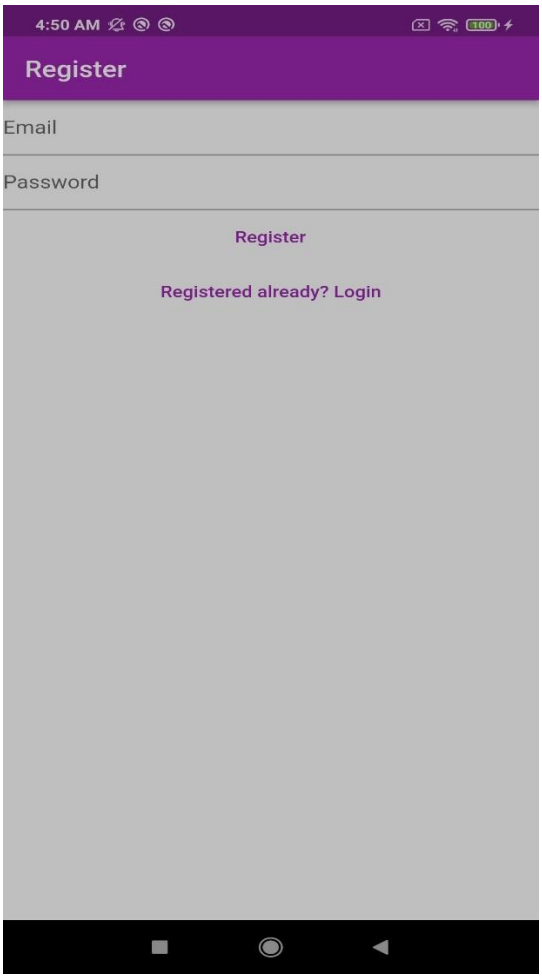
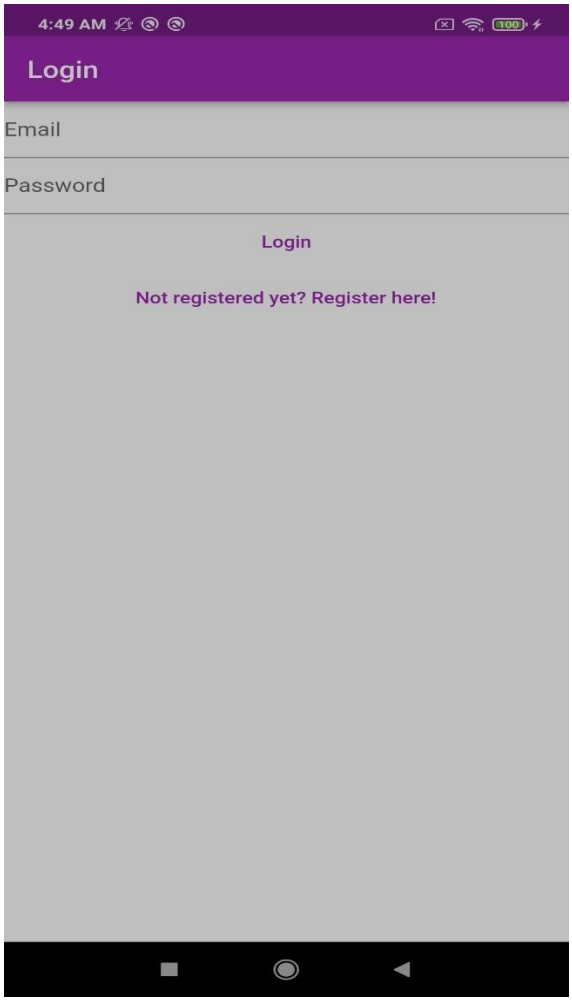


Figure 5.4: Register View

```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();
    }
  }
```

5.5.2 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 5.5: Login View

5.5.3 Implementation of codes without front-end:

Transportation View (Bus):

```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 5.6: Bus view

Email not verified view:

```
class _VerifyEmailViewState extends State<VerifyEmailView> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Verify Email'),),
      body: Column(children: [
            const Text('Check your mail for verification'),
            const Text('If you did not recieve any verification, please click below.'),
            TextButton(onPressed: () async {
              await AuthService.firebase().sendEmailVerification();
            },
             child: const Text('Send email verification again')
             ), // TextButton
             TextButton(onPressed:  () async{
               await AuthService.firebase().logOut();
               Navigator.of(context).pushNamedAndRemoveUntil(registerRoute, (route) => false);
             },
             child: const Text('Restart'),) // TextButton
          ],), // Column
    ); // Scaffold
  }
}
```

Figure 5.7: Email not verified

5.5.4 Some error dialog boxes:

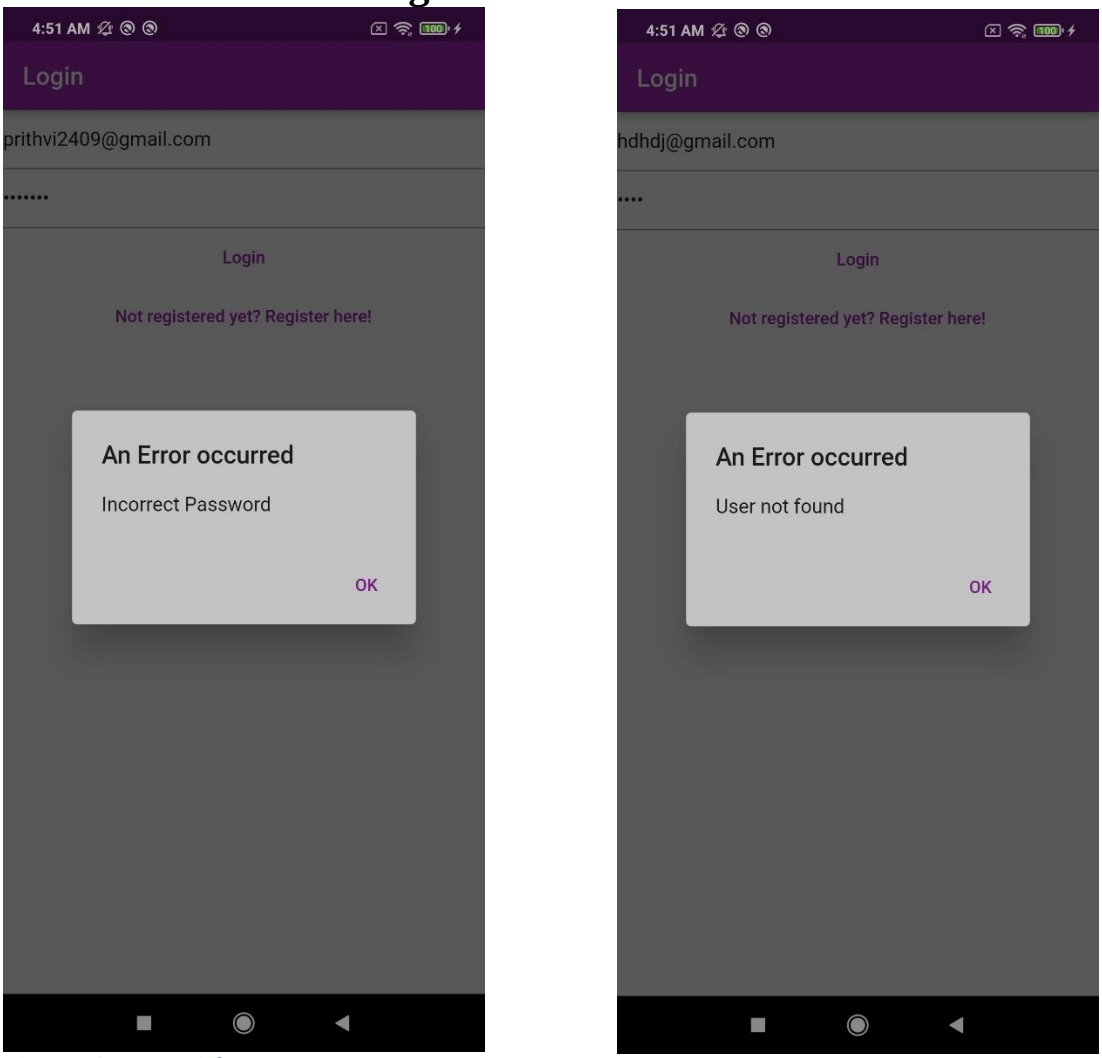


Figure 5.8: Error Dialog Boxes

5.5.5 Database creation using SQflite :

Creating a Bus class for transportation:

```dart
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 5.9: Database class for Bus

5.5.6 Some of the SQL:

```
const createUserTable= ''' CREATE TABLE IF NOT EXISTS "user" (
        "id"  INTEGER NOT NULL,
        "email" TEXT NOT NULL UNIQUE,
        PRIMARY KEY("id" AUTOINCREMENT)
      );
''';
const createNoteTable= ''' CREATE TABLE IF NOT EXISTS  "note" (
              "id"  INTEGER NOT NULL,
              "user_id" INTEGER NOT NULL,
              "text"  TEXT,
              "is_synced_with_cloud"  INTEGER NOT NULL DEFAULT 0,
              PRIMARY KEY("id" AUTOINCREMENT)
            );
''';
const createBusTable = '''
      CREATE TABLE IF NOT EXISTS "bus" (
        "id"    INTEGER NOT NULL,
        "ac_or_non_ac"    TEXT,
        "bus_no"    TEXT NOT NULL UNIQUE,
        "route"    TEXT,
        "per_seat_price"    TEXT,
        "departure"    TEXT,
        "arrive"    TEXT,

        "is_synced_with_cloud"    INTEGER NOT NULL DEFAULT 0,
        PRIMARY KEY("id" AUTOINCREMENT)
      );
      ''';
```

Figure 5.10: SQL

5.5.7 Some of the exceptions:

```
class DatabaseAlreadyOpenException implements Exception{}
class UnableToGetDocumentDirectory implements Exception{}
class DatabaseIsNotOpenException implements Exception{}
class CouldNotDeleteUser implements Exception{}
class UserAlreadyExists implements Exception{}
class CouldNotFindUser implements Exception{}
class CouldNotDeleteNote implements Exception{}
class CouldNotFindNote implements Exception{}
class CouldNotUpdateNote implements Exception{}
class CouldNotFindBus implements Exception{}
```

```
class UserNotFoundAuthException implements Exception{}
class WrongPasswordAuthException implements Exception{}
class WeakPasswordAuthException implements Exception{}
class InvalidEmailAuthException implements Exception{}
class EmailAlreadyInUseAuthException implements Exception{}
class GenericAuthException implements Exception{}
class UserNotLoggedInAuthException implements Exception{}
```

Figure 5.11: 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 5.12: Bus Object

5.5.8 Homepage:

```
@override
 Widget build(BuildContext context) {
   return FutureBuilder(
       future: AuthService.firebase().initialize(),
       builder: (context, snapshot) {
         switch (snapshot.connectionState){

           case ConnectionState.done:
           final user= AuthService.firebase().currentUser;
           if (user!= null){

           return const MyappView();
           } else {
              return const LoginView();

           }

             default:
             return const CircularProgressIndicator();
         }

       },

     ); // FutureBuilder

 }

}
```

Figure 5.13: Homepage

5.5.9 Routes:

```
home: const HomePage(),
routes: {
loginRoute:(context) => const LoginView(),
registerRoute:(context) => const RegisterView(),
myAppRoute:(context) => const MyappView(),
verifyMailRoute:(context) => const VerifyEmailView(),
phoneRoute:(context) => const RegisterPhoneView(),
otpRoute:(context) => const OTPScreen(),
noteViewRoute:(context) => const NotesView(),
},
```

Figure 5.14: Routes

5.6 Testing

Test case 1:

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

Figure 5.15: 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 5.16: 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 5.17: Test case 3

Test case 4 :

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

Figure 5.18: Test case 4

All the above test cases were based on authentications and all of them were successful.

Chapter 6: Results and Analysis

Several problems were discovered while carrying out tests. The authentication related bugs were solved by me. The analysis is described, and the findings are shown. When these issues were resolved, test cases were documented.

The results are concluded based on the test runs as stated in the previous chapter.

Firstly, we thought about few test cases involving the authentication or initialization processes, in scenarios where programs might crash.

Secondly, we approached those issues by designing our unit test cases by mocking one of the methods.

Four scenarios were tested so that users would be:

- Null after initialization.
- Able to log out and log in.
- Not able to log out if not initialized first.
- Initialized within two seconds.

All of the tests were successful, ensuring no errors related to authentication or initialization.

More test cases will be designed to make it ready for deployment.

Chapter 7: Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

The long-term use of a software system is ensured by its sustainability. The most important aspect of software engineering is determining how to make the software easily adapt to future changes. During the initial software development process, it is critical to develop software with a low environmental impact and a sufficient economic balance, and continuous quality monitoring ensures the software system's sustainability during its maintenance period.

Tripey OTA management system is being developed to make traveling more comfortable, and because tourism is important to our country's economy, financial sustainability can be achieved. It is a very promising application, and with more internal training and workshops, it will be possible to improve and make it more appealing to users. Monitoring continuous user feedbacks will also help to improve the software's quality and performance.

7.2 Social & Environmental Effects & Analysis

Social sustainability focuses on how to create software that increases the value of social capital. As a result, it is essential to focus on the software's social values. In terms of social sustainability, we must research on how this software is affecting the society. During pandemic, the entire country was placed under lockdown, and people were unable to travel safely for an extended period. People can finally travel freely now as the situation has been brought under control, and the Tripey OTA management system is being developed to make their vacations more relaxing and enjoyable.

Environmental sustainability focuses on how to create, use, protect, and discard software while minimizing its environmental impact. Efficient use of energy, runtime, and algorithm, processor density, memory usage can all be used to control the software's energy consumption. We need to be concerned about how the software affects the environment during development and maintenance. During the development of this application, it was ensured that only a small amount of electrical energy was used and that no harm was done to the environment.

7.3 Addressing Ethics & Ethical Issues

The “Software Engineering Code of Ethics and Professional Practice” specifies main principles that apply to software engineers' behavior and decisions, as well as the ethical guidelines to which they must comply. The principles ensure prioritizing the public's health, security, and wellbeing above everything. In developing the application, each decision is made upon available budget, clients’ needs, available software, reliability requirements and societal effects.

This mobile application collects basic user information such as a user's name, mobile number and email address, and the security of these information was prioritized because hackers can cause long-term damage to real people and businesses. Data is collected and managed in a way so that it is not shared anywhere. Security is addressed during development and after code release, rather than focusing just on getting the product to market.

Chapter 8: Lesson Learned

8.1 Problems faced during this period

Being hired as an intern at a start-up company has advantages and disadvantages. Start-up businesses typically lack both resources and staff and are unable to handle the enormous workload. I had no prior experience working in an office under a supervisor, so everything seemed very overwhelming at first. Getting used to the corporate environment while also learning about new languages seemed to be a lot of pressure at first.

I was not required to come into the office every day as I had the opportunity of a hybrid work environment. I also attended weekly check-in meetings and monthly presentations to discuss and evaluate the task's progress ensured a heavy workload. Missing deadlines was at times a close call, and I had to work overtime to learn and compete my assigned tasks on time.

It seemed overwhelming at first because it was my first time working with a new mobile application development framework like Flutter and learning a new programming language, Dart. Learning new syntaxes and their functionalities appeared to be confusing and tiresome initially. And at last, applying everything learned to real-world situations proved to be even more complicated.

In longer codes, bugs are almost unavoidable. Because of these bugs, the application used to crash on its own, and finding them was tough. Even when they were discovered, it was more difficult to remove them. Some bugs were so hard to eliminate that sometimes it took a few days due to thorough reviewing the entire code several times.

8.2 Solution of those Problems

As time passed, I gradually became accustomed to the corporate environment and could complete tasks on time. Working at a small company enabled me to gain hands on experience and allowed me to learn everything thoroughly at my own pace and comfort. Four years of undergraduate life taught me a great deal about hard work and patience.

To deal with immense workload and time crunch, I had to make a schedule beforehand and strictly maintain it to meet all the deadlines. Not piling up work and finishing them at the last moment also helped me to remain calm and be content with my tasks.

I spent my free time on the internet learning about Flutter and Dart from various documents and videos, which helped me cope with tasks assigned to me at work.

I frequently searched for different documentations from ‘Stack Overflow’, ‘Geeks for Geeks’, etc. in resolving several issues. Most of the time, I was able to resolve the problems on my own. My supervisor was always willing to assist me whenever I approached him with a problem.

Chapter 9: Future Work & Conclusion

9.1 Future Works

The company plans to add more functionalities to the current system as it grows. Before that, the current stage of the application is still under development.

Some of the future aspects are:

- A web application which will be under development soon after the application gets launched into the market.
- Gamification: the users can earn various points which they can use to win reward such as discounts on tickets, meals, a free night at a hotel, etc.
- Implementation of ML for creating recommendation systems.
- Implementation of AI to create live chat bots.

9.2 Conclusion

I learned a lot about developing mobile applications and corporate culture throughout my internship. Working with Flutter and Dart taught me a lot about software development as I faced new obstacles every day. The most crucial lesson from the whole program was to use multiple methods to find solutions to all the challenges. Giving up was not an option, and it took a lot of patience to find and fix bugs. My capacity to conduct research and find materials that supported my work was also improved. I was able to work and develop professional experience in my area of interest. I'm sure that this internship program will be useful to me in my future professional endeavors.

I am grateful to Tripey for providing me with this opportunity to learn about mobile application development and preparing me for the corporate sector. The guidance of my supervisor helped me learn everything from scratch with patience. My experience would not have been as smooth without him.

References

1. “XpenceCo.” XpenceCo, xpenseco.com. Accessed 13 Oct. 2022.
2. “Tripadvisor: Over a Billion Reviews and Contributions for Hotels, Attractions, Restaurants, and More.” Tripadvisor, www.tripadvisor.com. Accessed 13 Oct. 2022.
3. “Welcome to Roadtrippers.” Roadtrippers, roadtrippers.com. Accessed 13 Oct. 2022.
4. “GoZayaan.” GoZayaan, www.gozayaan.com. Accessed 13 Oct. 2022.
5. Cook, B. (2022, December 19). Everything You Need to Know About Software Engineering Ethics. Fellow.app. https://fellow.app/blog/engineering/engineering-everything-you-need-to-know-about-software-engineering-ethics/
6. Eischen, K. (2002). The Social Impact of Informational Production: Software Development as an Informational Practice. UC Santa Cruz: Center for Global, International and Regional Studies. Retrieved from https://escholarship.org/uc/item/9rp0c3w4
7. Cirak, K., & Bolat, H. B. S. (2022). Analysis of the Relationship Between Sustainability and Software Performance. International Journal of Business Analytics, 9(5), 1–13. https://doi.org/10.4018/ijban.298019
8. Gotterbarn, D. (2002). Software Engineering Ethics. Encyclopedia of Software Engineering. https://doi.org/10.1002/0471028959.sof314
9. Penzenstadler, Birgit. (2013). What does Sustainability mean in and for Software Engineering?.



An Undergraduate Internship/Project on yourTopic

By

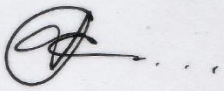
Muzayed Imam

Student ID: **1820223**

Autumn, 2022

Consent From

The student modified the internship final report as per the recommendations made by his/her academic supervisor and/or panel members during and/or before final viva, and the department can use this version for archiving as well as the OBE course material for CSE499.



Mr. Sanzar Adnan Alam

Department of Computer Science & Engineering

Independent University, Bangladesh