



WOLKITE UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF SOFTWARE ENGINEERING

PROJECT TITLE: -

TRAVEL GUIDE APPLICATION IN ANDROID

PREPARED BY

<u>NO</u>	<u>NAME OF STUDENTS</u>	<u>ID NO</u>
1	SEWMEHON ASCHENEK	CIR/ 304/10
2	AMBESAW SIMACHEW	CIR /246/10
3	MERTIANOS ALEHEGN	CIR/ 250/10
4	ADUGNAW TESHALE	CIR/ 244/10

PROJECT ADVISOR: YAREGAL NIGATU (MSC.)

August 22, 2021 Wilkeite, Ethiopia

WOLKITE UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF SOFTWARE ENGINEERING

PROJECT TITLE: -

TRAVEL GUIDE APPLICATION IN ANDROID

SUBMITTED TO DEPARTMENT OF SOFTWARE ENGINEERING
IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHLER OF SCIENCE IN SOFTWARE
ENGINEERING

PREPARED BY

<u>NO</u>	NAME OF STUDENTS	ID <u>NO</u>
1	SEWMEHON ASCHENEK	CIR/ 304/10
2	AMBESAW SIMACHEW	CIR /246/10
3	MERTIANOS ALEHEGN	CIR/ 250/10
4	ADUGNAW TESHALE	CIR/ 244/10

PROJECT ADVISOR: YAREGAL NIGATU (MSC.)

Wolkite University, Wolkite, Ethiopia

August 22, 2021

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Yaregal N. and having the title Travel guide application in android is the sole contribution of Ambesaw Simachew ,Adugnaw Teshale,Mertianos Alehegn, Sewmehon Aschenek. No part of the project work has been reproduced illegally (copy and paste) which can be considered as plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if the violation of this declaration is proven.

Date: _____

Group Members: -

NO	Full name	Signature
1	Ambesaw Simachew	_____
2	Adugnaw Teshale	_____
3	Mertianos Alehegn	_____
4	Sewmehon Aschenek	_____

APPROVAL FORM

This is to confirm that the project report entitled Travel guide application in android submitted to Wolkite University, College of Computing and Informatics Department of Software Engineering by Ambesaw Simachew ,Adugnaw Teshale,Mertianos Alehegn, Sewmehon Aschenek is approved for submission.

Advisor name	Signature	Date
-----	-----	-----
Department head name	Signature	Date
-----	-----	-----
Examiner 1 name	Signature	Date
-----	-----	-----
Examiner 2 name	Signature	Date
-----	-----	-----
Examiner 3 name	Signature	Date
-----	-----	-----

ACKNOWLEDGMENT

First and for most, we would like to thanks the almighty GOD for help us to be alive and to do all of this. Thanks, our family for everything, and we have very special thanks to our advisor Mr. Yaregal N for guiding us, correcting our mistakes, constant supervision, encouraging us to do more and providing necessary information regarding the project. We have also so many individuals, groups, instructors, department deans, faculty deans, and students that we want to thank for supporting while working on this project.

TABLE OF CONTENTS

DECLARATION	i
APPROVAL FORM	ii
ACKNOWLEDGMENT.....	iii
LIST OF TABLES	ix
LIST OF FIGURES	xi
EXECUTIVE SUMMARY	xiv
CHAPTER ONE	1
1. INTRODUCTION	1
1.1 Background study for Ministry of Culture and Tourism (Ethiopia)	1
1.1.1 Vision.....	2
1.1.2 Mission	2
1.2 Problem Statement	2
1.3 Objective of the Project.....	3
1.3.1 Main Objective	3
1.3.2 Specific Objective.....	3
1.3 Feasibility Analysis	3
1.4.1 Technical feasibility	4
1.4.2 Operational feasibility	4
1.4.3 Economic feasibility.....	5
1.4.5 Legal feasibility	6
1.5 Scope and Limitation of the Project.....	6
1.5.1 Scope Project	6
1.5.2. Limitation of The Project	7
1.6 Significance of the Study	7

1.7. Beneficiary of the Project.....	7
1.8. Methodology of the Project.....	8
1.8.1. Data Collection Tools/Techniques	8
1.8.2. System Analysis and Design	8
1.8.3. System Development Model	8
1.8.4. System Testing Methodology	9
1.8.5. Development Tools and Technologies	10
1.9 . Document Organization	12
CHAPTER TWO	13
2. DESCRIPTION OF THE EXISTING SYSTEM	13
2.1. Introduction of Existing System.....	13
2.2 Users of Existing System	13
2.3. Major Functions of the Existing System	14
2.4. Forms and Other Documents of the Existing Systems (if any).....	15
2.5. Drawbacks of the Existing System.....	16
2.6. Business Rules of the Existing System	17
CHAPTER THREE	18
3.PROPOSED SYSTEM	18
3.1 Functional Requirements.....	18
3.2. Non-functional Requirements	19
3.4.1. User Interface and Human Factors	19
3.4.2. Hardware Consideration	20
3.4.3. Security Issues	20
3.4.4. Performance Consideration	21
3.4.5. Error Handling and Validation	21

3.4.6. Quality Issues	21
3.4.7. Backup and Recover	22
3.4.8. Physical Environment.....	22
3.4.9. Resource Issues.....	22
3.4.10. Documentation.....	22
CHAPTER FOUR.....	23
4. SYSTEM ANALYSIS	23
4.1. System Model.....	23
4.1.1. Use Case Model.....	23
4.2. Object Model.....	45
4.2.1. Class Diagram.....	45
4.2.2. Data Dictionary.....	46
4.3. Dynamic Model.....	52
4.3.1. Sequence Diagram.....	52
4.3.2. Activity Diagram	58
4.3.3. State Chart Diagram	61
CHAPTER FIVE	65
5. SYSTEM DESIGN	65
5.1. Design Goals	65
5.1.1. Performance.....	65
5.1.2. Dependability.....	65
5.1.3. Maintenance.....	66
5.1.4. End-user.....	66
5.1.5. Priorities of Design Goal	66
5.2. Proposed System Architecture	68

5.2.1. Subsystem Decomposition and Description	69
5.2.2. Hardware/Software Mapping	71
5.2.3. Detailed Class Diagram	72
5.2.5. Access Control and Security.....	75
5.3. Packages	75
5.4. Algorithm Design.....	77
5.5. User Interface Design.....	78
CHAPTER SIX 6.....	82
6. IMPLEMENTATION AND TESTING	82
6.1 introduction	82
6.2. Implementation of the Database.....	82
6.2.1. Whenever you found appropriate, implement the following: Indexes, Views, Triggers, Stored procedures.....	86
6.2.2. Scheduled backup of database-tables with new data in month	88
6.2.3. Configure database level security.....	89
6.3 Implementation of the Class Diagram.....	90
6.4. Configuration of Application Security.....	92
6.4.1. Implement all input validations properly.....	92
6.4.2. Implement encryption/deception	94
6.4.3.Roles must be defined clearly.....	94
6.5.4.User accounts must be assigned with necessary access privileges.....	95
6.7. Implementation of User Interface	95
6.8. Testing.....	101
6.8.1. Testing Tools and Environment	101
6.8.2. Unit Testing	102

6.8.3 Integration Testing.....	104
6.8.4. System Testing	104
CHAPTER SEVEN	108
7. CONCLUSION AND RECOMMENDATION.....	108
7.1 Conclusion.....	108
7.2 Recommendation.....	108
Appendix I: Sample Source Code.....	109
References.....	123

LIST OF TABLES

Table 1. 1 Existing System Traveler Cost In Ethiopia.....	5
Table 1. 2 Proposed System Traveler Cost In Ethiopia.....	5
Table 1. 3 Software Tools.....	11
Table 2. 1 User Of Existing System.....	14
Table 4. 1 Use Case Description For Login.....	25
Table 4. 2 Use Case Description For User Registration.....	26
Table 4. 3 Use Case Description For Creating Profile.....	27
Table 4. 4 Use Case Description For Traveler View Package Tour.....	28
Table 4. 5 Use Case Information For Traveler Research Tour Package.....	30
Table 4. 6 Use Case Description For View Location.....	31
Table 4. 7 Use Case Description For Traveler View Weather Condition.....	32
Table 4. 8 Use Case Description For Admin Add Tour Package.....	34
Table 4. 9 Use Case Description For Admin Post Advertisement.....	35
Table 4. 10 Use Case Description For Admin Delete Tour Package.....	36
Table 4. 11 Use Case Description For Admin Deleted Posted Information.....	37
Table 4. 12 The Use Case Description For Adding Travel Service.....	38
Table 4. 13 The Use Case Description For Successful Deletion Of Travel Service.....	39
Table 4. 14 Data Dictionary For A Traveler.....	47
Table 4. 15 Data Dictionary For Tourism Site.....	48
Table 4. 16 Data Dictionary For Weather Condition.....	48
Table 4. 17 Data Dictionary For Admin.....	49
Table 4. 18 Data Dictionary for Hotel.....	50

Table 4. 19 Data Dictionary For Advertisement	51
Table 4. 20 Data Dictionary For Travelagency	51
Table 4. 21 Data Dictionary For Transportation.....	52
Table 5. 1 Access Control For User.....	75
Table 6. 1 Implementation Of The Database User Table	83
Table 6. 2 Implementation Of The Database Tourism Site Table	84
Table 6. 3 Implementation of the Database City Table	84
Table 6. 4 Implementation Of The Database Order Service.....	84
Table 6. 5 Implementation of the Database Hotel Table	85
Table 6. 6 Implementation Of The Database Special Event Table.....	85
Table 6. 7 Implementation Of The Database Travel Agency	86
Table 6. 8 Cloud Firestore Function Triggering	87
Table 6. 9 Unit Test For Authenticated Login Page	102
Table 6. 10 Unit Test For Register Travel Agency.....	102
Table 6. 11 System testing for authenticated login page	104
Table 6. 12 System Testing For Order Service For User Part	106

LIST OF FIGURES

Figure 2. 1Travel Information Form.....	16
Figure 4. 1 Use Case Diagram Of The System.....	25
Figure 4. 2 class diagram	46
Figure 4. 3 Sequence Diagram For Login.....	53
Figure 4. 4 Sequence Diagram For Register.....	54
Figure 4. 5 Sequence Diagram For Profile	55
Figure 4. 6 Sequence Diagram For Add Tour Package	56
Figure 4. 7 Sequence Diagram For Post Advertisement.....	57
Figure 4. 8 Sequence Diagram For View Tour Package	58
Figure 4. 9 Activity Diagram For Login.....	59
Figure 4. 10 Activity Diagram	
Figure4. 1 Use Case Diagram Of The System.....	25
Figure4. 2 class diagram	46
Figure4. 3 Sequence Diagram For Login.....	53
Figure4. 4 Sequence Diagram For Register.....	54
Figure4. 5 Sequence Diagram For Profile	55
Figure4. 6 Sequence Diagram For Add Tour Package	56
Figure4. 7 Sequence Diagram For Post Advertisement.....	57
Figure4. 8 Sequence Diagram For View Tour Package	58
Figure4. 9 Activity Diagram For Login.....	59
Figure4. 10 Activity Diagram For User Registration	60
Figure4. 11 Activity Diagram For Add Tour Package	60
Figure 4. 12 Activity Diagram For Provid Service.....	60

Figure4. 13 Login State Chart Diagram.....	61
Figure4. 14 Registration State Diagram.....	62
Figure4. 15 View Package State Diagram	62
Figure4. 16 Search State Chart Diagram	63
Figure4. 17 Add Tour Package State Diagram	63
Figure4. 18 Delete Package State Diagram	64
Figure4. 19 Advertisement State Diagram.....	64
For User Registration.....	60
Figure 4. 11 Activity Diagram For Add Tour Package	60
Figure 4. 12 Activity Diagram For Provid Service.....	60
Figure 4. 13 Login State Chart Diagram.....	61
Figure 4. 14 Registration State Diagram.....	62
Figure 4. 15 View Package State Diagram	62
Figure 4. 16 Search State Chart Diagram	63
Figure 4. 17 Add Tour Package State Diagram	64
Figure 4. 18 Delete Package State Diagram	64
Figure 4. 19 Post Advertisement State Diagram.....	64
Figure 5. 1 Proposed System Architecture For Traveler Guide Application.....	68
Figure 5. 2 Subsystem Decompositions.....	71
Figure 5. 3 System Deployment Diagram.....	72
Figure 5. 4 . Detailed Class Diagram	73
Figure 5. 5 Persistent Diagram.....	74
Figure 5. 6 Package Diagram.....	76
Figure 5. 7 User Interface Design For Login Page	79

Figure 5. 8 The User Interface Design For Registration.....	80
Figure 5. 9 The User Interface Design For Create Profile	81
Figure 6. 1 Show Firebase Database With All Table.....	83
Figure 6. 2 Implement encryption/deception	94
Figure 6. 3 Home page.....	96
Figure 6. 4 user interface for Login Page for all user	97
Figure 6. 5 User Interface For Tourism Site	98
Figure 6. 6 User interface for Main page every one user can access	99
Figure 6. 7 User Interface For Order Service	100
Figure 6. 8 User interface for Traveler page.....	101
Figure 6. 9 Integration Testing for travel guide application	104
Figure 6. 10 System Testing For Authenticated Login Page	105
Figure 6. 11 System Testing For Order Service For User Part.....	107

EXECUTIVE SUMMARY

In recent years, the development of worldwide travel has been outstanding. At the same time, due to the fast development of the travel business, it is important to offer tourists an enhanced travel platform that uses a mobile device as a means of travel guide that can be accessed anytime and anywhere. This application allows one to easily access the information of tourism sites in Ethiopia. This can be done by identifying problems of an existing system in the Ethiopia Tourism industry. Some of the issues that we will investigate are how hard it is to know the location of different tourism sites and how they find the right people to contact if they needed some help or make some payment, and another difficulty is that there are many fake tourist sites so in this application we shall have to only put tourism sites that have been approved by the Ethiopia Tourism Board.

The most interesting category is Travel Guides that combines the Information Resources and Location-Based Services category. We will carry out a data collection on the different tourists who would be found on some sites. The people that will be interviewed include the site manager, tourists, and tourist brokers who will help in the achieving in coming up with a legit solution and also use firebase and Android studio used to develop the Travel Guide Application system. They will provide first-hand information about the problems they face and how they thought problems can be solved. We will build a system to solve all existing problems in the tourism industry; the system will have many modules. There will be an Android Application that will be used by tourists to access all the information designed for them. The system administrator will be the one to update the tourist sites.

CHAPTER ONE

1. INTRODUCTION

The increasing use of smartphones and internet technology has encouraged developers to develop mobile applications. Mobile tourism is gaining popularity in the last few years which uses a mobile device as a means of travel guide. The traveler search for information about the places they want to visit using a travel guide application. The travel guide android application which is installed on the tourist's cell phones acts as tour advisors. Even if the tourist can find new places through the internet, it is tedious to obtain necessary information about the places to visit, when the tourist is not familiar with the place. This paper describes the application which enhances the tourism experience using mobile software technology. Mobile tour management services will assist a developer to develop a mobile application to improve travel experience based on real-time data values. Without having a classic guide, it will help one to get information of the places to visit, lodges to stay, restaurants to dine, check out the photos and current weather which narrates the location description which they are currently visiting [1].

In the earlier tourism system, whenever a traveler visits famous destinations, they hire a guide who describes them regarding the places. Tourists are not able to get travel information on the time when they are on a trip. The proposed system doesn't require such a classic physical guide. The visitor who is not aware of the places can get better guidance before visiting the place. The travel guide system android application provides the users with tour guidance information that they need anytime and anywhere.

The information that the tour guide android application provides helps the tourists to find the desired locations. It provides all the details of the desired locations and how to reach those locations. It also provides basic information to the user to decide the destinations. This paper is useful for the tourists who are unaware of the places they want to visit. The travel guide application makes it possible for the users to plan their trips, make reservations, and make other required arrangements for their trip. This would provide them with more time to enjoy their vacation [2].

1.1 Background study for Ministry of Culture and Tourism (Ethiopia)

The Ethiopian Ministry of Culture and Tourism is the ministry of the Government of Ethiopia responsible for researching, preserving, developing, and promoting the culture and

tourist attractions of Ethiopia and its peoples. inside the country and internationally. In doing so the Ministry closely works together with different national and international stakeholders. Subordinate bodies include the Authority for Research and Conservation of Cultural Heritage (ARCCCH), Ethiopian Wildlife Conservation Authority (EWCA), National Archives and Library Agency, and Ethiopian National Theatre [3], both. The Ministry publicizes the country's resources of tourist attractions and encourages the development of tourist facilities. It also licenses and supervises establishments of tourist facilities such as hotels and tour operators, and sets the standards for them.

Ethiopia is blessed with abundant natural tourist attractions, including nine World Heritage Sites, but the Ministry of Culture and Tourism still struggles to attract tourists in decent numbers owing to poor investment, security and does not have any cohesive tourism development or promotion strategy. As a result, most tourists fly over Ethiopia to Kenya, Uganda, Tanzania, South Africa to name a few countries.

1.1.1 Vision

The tide is turning for Ethiopia's tourism industry, as the Ethiopian Ministry of Culture and Tourism recently announced it aims to triple the number of foreign visitors, to more than 2.5 million, by 2020, with an ultimate goal of making Ethiopia one of Africa's top five tourist destinations by 2020.

1.1.2 Mission

Mission to study, preserve, develop and promote the cultural wealth and the national tourism attractions of the nations, nationalities, and peoples of Ethiopia and to build the positive images of Ethiopia to add sustainable socio-economic and political values with popular and stakeholder's participation.

1.2 Problem Statement

In the past years, tourists have been facing the problem of accessing tourism information such as locating national parks and other tourist sites in Ethiopia. Both national and international tourists have been having trouble when touring and less aware of what to find and expect and also locating places at different sites such as national parks, most especially about the accommodation.

The tourists are going to the travel agency to get the service. They have no information about the services they get such as hotel, car rental service, and another travel service.No

idea about the weather condition of the area or the place they want to visit. No ideas about the cultures and customs of a place. Not having enough information about the thing, they want to visit and where they found it.

They have used a different site to get the information, they face the following problem travelers have access to information regarding the places they visit. But this information lacks completeness. To explore the different aspects of a particular place – such as general facts, photographs, historical events, and other visitors' experiences – these travelers need to visit a separate website for each category of information. Therefore, we propose the development of a travel guide application whereby the tourist can have access to tourism information anytime anywhere.

1.3 Objective of the Project

1.3.1 Main Objective

To develop an Android Based travel guide application that will provide information about tourism sites in Ethiopia.

1.3.2 Specific Objective

Below are the specific objects of the project: -

- To study and analysis the Existing system
- To design a travel guide application, this will include the back-end and the UI interfaces of the component module.
- To implement the individual component module.
- To test and debug the application of each component module.

1.3 Feasibility Analysis

The feasibility study is essential to evaluate the cost and benefits of the new system. Need of the feasibility study:

- It determines the potential of the existing system.
- It is used to determine/finds out the problem of the existing system.
- To determine all goals of the new system.
- It finds all possible solutions to the problems of the existing system.

1.4.1 Technical feasibility

In the technical feasibility study, one has to test whether the proposed system can be developed using existing technology or not [4].

- It is planned to implement the proposed system using an Android-based applications system. So, it is evident that the necessary hardware and software are available for the development and implementation of the proposed system: - like CPU above 2.3 GHZ and RAM above 4GB and MS word: document tools and Enterprise Architect: use to make UML diagrams. So, the system will technically feasible.
- Also, the group members have enough knowledge to develop the proposed system/project.
- The system is going to be developed by technological development techniques such as JAVA, XML, Android. And the team can develop this system without any difficulty since the team has studied the required methodologies and tools. So, the system will be technically feasible.

1.4.2 Operational feasibility

It determines how the proposed system will satisfy the organization's needs and it also offers a Secure, accurate, and efficient system to the organization. The system performs all of its tasks in a user-friendly manner. In which, anyone without an advanced technical background can use the system [5]. The system is also compatible with all android platforms. So, the project is operationally feasible.

- The system is designed in such a way that it is easy to operate it has a well-defined user interface and is authenticated with user name and password.
- It's aimed at giving an easy way to generate reports.
- The functional accessibility is also structured in such a way it won't take much time for anyone to get access to a system.
- The system has a user manual (training manual) that the new user uses to use the system.
- The system provides better and effective service for the user.
- The system performs its activity without failure for 24 hours and 7 days so it is reliable.
- The system performs its activity in a timely processing manner.

1.4.3 Economic feasibility

Economic feasibility is the process of identifying the financial benefits and costs associated with the project being developed. It determines whether the investment that goes into the implementation of the project is recoverable or not. The cost-benefit analysis is a commonly used method in evaluating the effectiveness of the system. The cost of developing this system currently includes the minimum software cost to develop and run the system.

Tangible benefit: -: The team calculated the corresponding tangible benefits based on the technique called the Time Value of Money (TVM).

Table 1. 1 Existing System Traveler Cost In Ethiopia

Existing system Traveler cost in Ethiopia			
Type	Cost per day in ETB	For one month	Total price/salary per year
Taxi	400	X30	120,000
Cost for the person to indicate different place	300	X30	9000
Total materials cost			129000

Table 1. 2 Proposed System Traveler Cost In Ethiopia

Proposed system Traveler cost in Ethiopia			
Type	Cost per day in ETB	For one month	Total price/salary per year
Taxi	97	X30	2910
Cost for the person to indicate different place	0	X30	0

Total materials cost			2910
----------------------	--	--	------

Difference between before and after deployment money required for payment.

Cos reduction for materials=129000-2910 birr=126090 birr.

1.4.4 Cultural feasibility

Cultural feasibility study is defined as one that investigates scientific as well as ethical, behavioral, and social issues.

- The system will not harm any of the religious beliefs, ethical, behavioral, and social issues in the society
- Any traveler use this system must conserve the rule and the regulation of the society you vested.
- The system has no environmental factors to be considered.
- This system will be a great impact on the development of culture because by using this application the traveler will visit every cultural event in Ethiopia and this system becomes the source of income for a different cultural event.
- This system no impact on any of the enterprise's own culture rather it has more benefits.

1.4.5 Legal feasibility

The system is free from any political and legal rule associated with politics but the travel guide management system works with the rule and regulation of the ministry of culture and tourism bureau.

1.5 Scope and Limitation of the Project

1.5.1 Scope Project

The project is carrying out, will include the recognized tourism sites by the Ethiopia Tourism Board. The major aim of the application is to guide the tourists to locate different tourist sites across the country and showing contact information that the Tourist can use to contact the respective tourism agent or guider and also this system can contain information for locating the nearest hotel, locating the historical, cultural event and see the weather condition of Ethiopia is possible. This application contains two languages and developed using flutter i.e. It is possible to use both in the android and IOS operating system.

1.5.2. Limitation of The Project

- Due to the time limitation we cannot include hotel reservation. our team discussed about hotel reservation and its payment system by creating one database the transaction takes place but this is difficult to finish in this short time.
- Our system shows only a place which is stored on google map

1.6 Significance of the Study

This research and implementation of the project is important both economically, socially, and academically, some of the reasons why this study is significant are mentioned below:

- International Tourists their finding of accredited tourism sites will be made easy.
- Tourism sites will reduce the costs spent on advertising the information of the site.
- Remote tourist sites will be brought to the public, making them accessible and easy to find by tourists [6].
- Planning for trips in advance is made easier due to easy access to information [6].
- All Information about natural resources can be searched easily.
- People can gain access to the information about the place they wanted to visit at any time as far as the network is available.
- Gives accurate information
- Simplifies the manual work.
- Increase user satisfaction
- Easily understand about area and place they want to travel
- Get faster processing time and more accurate data for travel request

1.7. Beneficiary of the Project

The customers are benefited from this system since they save time, they get information from the traveler guide application system simply by visiting the site and booking. Here are the beneficiaries of the project.

- The travel agency- The system contains the location of the travel agencies and full information about the travel agency services such as hotel car rental service and any other service they provide so travelers easily access it.
- Customer or tourist- they can be benefited from this system by easily get any type of information about tourism to sit in Ethiopia, easily get hotel they want e.tc

- Ethiopian culture and tourism- it can be easily advertising the newly created tourism site so this will be greatly reducing advertising cost.
- Guider-in our country there are a lot of illegal guides have but this system only contains legal guider information and address so if travel wants a guide only contact legal one.
- The local service providers: -such as hotel, food-our system also contain information about the hotel and their service according to their stars so he traveler can easily access it.

1.8. Methodology of the Project

To accomplish this project on time and within the cost, we would follow different procedures which are described below.

1.8.1. Data Collection Tools/Techniques

If you want some information from those websites you need to see different websites to get full information about tourism site, the hotel you reserve, etc. So by this and another reason, we will propose a travel guide application that will provide full information about tourism site in Ethiopia, the hotel they reserve, the agency they want to contact, and also contains a contact information for guider for each place they want to visit.

1.8.2. System Analysis and Design

In the system analysis and design phase of a project, we will use the object-oriented approach that examines requirements from the perspective of the class and objects found in the problem domain. The reasons that we will use the object-oriented approaches are listed below.

- Everything in the android application is an object.
- We use java and XML to create a travel guide application.
- In XML we declare components as data objects.
- In XML we store components in terms of objects.
- Using java, we perform the operations of the object.

1.8.3. System Development Model

We chose iterative model to incorporating the necessary changes ideas comes from applicants and employer on every phase to its preceding phases. The change idea paths allow the phase to be reworked in which errors are committed and these changes are

reflected in the later phases. Because of the system is used by different kind of user we can update proposed system depending on user's desire, due to an iterative model allow to make change and update proposed system, we select iterative model.

1.Requirement gathering & analysis: In this phase, requirements are gathered from customers and check whether requirements will fulfil or not. Our team checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.

2. Design: In the design phase, team design the software by the different diagrams like ,activity diagram, class diagram, state transition diagram, etc.

3.Implementation: In the implementation, requirements are written in the coding language and transformed into computer programmers which are called Software.

4.Testing: After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common our team use a black box test methods.

5. Deployment: After completing all the phases, software is deployed on google play store.

7. Review: In this phase, after the product deployment, review phase is performed to check the behavior and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.

8. Maintenance: In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options

1.8.4. System Testing Methodology

Developing software is a complex process. No matter how hard we try to eliminate all faults simply by going through the phases of requirements, analysis, design, specification, and implementation, however, through good practice, we can make sure that the most series fault does not occur in the first place. Also, we need a separate testing phase, to eliminate all remaining faults before release. To simplify the testing process the project team followed the different types of tests that break the testing process up into distinct levels.

Unit Testing

Using unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The

unit test cases shall be designed to test the validity of the program's correctness. In this level of testing process.

Sample Tests

- Check whether the return type of the functions is correct.
- Check how the sub procedures or functions are called correctly.
- Check if the correct output is produced for different inputs.
- Check the efficiency of the code concerning the memory and CPU time.

Integration Testing

Once each individual part of the system is tested, every smallest unit is tested, different modules of the system are now integrated together and tested. We use integration to test their integration of different modules of proposed system (package module, post advertisement module, admin module) with each other in one.

Sample Teste

- Evaluate the functionality of the subsystem after combination all individual function.
- Identify the independence of each subsystem with other subsystems.

System testing

In this level of the testing process, we have examined how nicely the subsystems of the whole travel guide application are worked together to achieve the desired goal.

1.8.5. Development Tools and Technologies

Our system will use the following tools and technologies can be used for the implementation and after implementation

1.8.5.1 Frontend Technologies

The user interface is developed using the Android Studio integrated development environment since it easily designing the front end and connected it into firebase realizing rapid application development with constraints on the hand.

- XML language
- Android (java programming language)
- HTML, CSS, and JavaScript for web part

1.8.5.2 Backend Technologies

Firestore is an open-source mobile-based platform critical for establishing a flexible solution to the development of outstanding business apps that can grow your business by a great deal [7].

Firestore:- is best for Android app development as well as Web and iOS app development. Because it provides lots of functionalities such as,

- Real-time DB
- Cross-platform support
- Easy to integrate with Android, iOS, and Web.
- Crash Reporting
- Authentication
- Testing
- Hosting
- Grow & engage your audience with ad words, cloud messaging, target specific users

Software tools: are software installed on a computer for a different purpose from documentation up to the implementation

Table 1. 3 Software Tools

Software tools	Description
Microsoft Word 2016	For documenting the corresponding deliverables associated with the project.
Enterprise architect	For designing unified modeling language (UML) diagrams.
Adobe Photoshop	For design user interface.
Android studio	For design travel guide application
Firestore	For storing data in a real-time database
Google map	For locating a specific place

1.8.5.3. Documentation and Modeling Tools

- Microsoft word for documentation
- Enterprise Architect for design unified modeling language (UML)

1.8.5.4. Deployment Environment

For testing purpose, we want to keep the deployment under localhost. But since our application is a multiplatform app, we want to support data sharing between these platforms. So, we are planning on deploying the web app over the cloud on Heroku servers. Since Heroku provide a free trial version with 500 GB, we are planning to take that advantage. And as for the database and also the android app deployed on google play store.

1.9. Document Organization

- Chapter one defines and describes concepts about the introduction of the Travel Guide application, objective, feasibility analysis, Scope, limitation, significance, target Beneficiary, etc.
- The chapter discusses problems in the existing system, the major functionality of the existing system, the drawback of the existing system, the business rule of the existing system, etc.
- Chapter Three is about the Overview of the proposed system which includes functional and non-functional requirements.
- Chapter four is about system models such as use case model, use case diagram, use case scenario, and about object model such as class diagram, data dictionary and also about the dynamic model such as activity diagram, sequence diagram state chart diagram, etc.
- Chapter five is all about system design such as design goals, current system architecture, proposed system architecture, software & hardware mapping, etc.
- Chapter Six in this chapter will discuss the implementation and Testing
- In chapter seven we will discuss the conclusion and recommendation of the Travel Guide Application

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

The existing system, the Ethiopia tourism bureau advertising the tourism site mainly through newspapers, magazines, radio, and other simple ways that are available easily. The user can be accessing tourism information such as locating national parks and other tourist sites in Ethiopia. By visiting the travel agencies and enquire the information and the user needs to book the hotels manually and also get rental service by direct contact to the agency, and the tourists are guided by with having a classic guide. So they have no information about the service they get such as food, hotel, car, and parking area. No idea about the weather condition of the area or the place they want to visit etc. So we Proposed this project that will overcome all difficulties. The travelers are required to maintain considerable information on the place they visit, their transportation, the food-related information, travel information, extensions, etc.

2.2 Users of Existing System

The Ethiopian Ministry of Culture and Tourism is the ministry of the Government of Ethiopia responsible for researching, preserving, developing, and promoting the culture and tourist attractions of Ethiopia and its peoples, both inside the country and internationally .In doing so the Ministry closely works together with different national and stakeholders. In this system, a stakeholder has to go to various agencies to find details of places and to book hotels. The Ethiopian Ministry of Culture and Tourism has been defined as the following user roles:

Table 2. 1 User Of Existing System

User	The major functionality of the Existing system is done manually.
Ethiopia culture and tourism bureau	<ul style="list-style-type: none"> ● Developing and promoting tourism. ● Introducing newly created tourism site through the newspaper, radio, TV, etc. ● The report documented information about the number of travelers being traveled to the tourism areas. ● Collecting information about the travelers. ● Prepare a tour package. ● Provide hotel information and give information about guider to their traveler.
Travel agency	<ul style="list-style-type: none"> ● Give the service of guide and transportation for the tourists. ● Should report monthly traveler's information to the culture and tourism bureau manually. ● Provide travel information for tourists. ● Can provide services like Car rental service, hotel service, and guide service.
Tourist	<ul style="list-style-type: none"> ● Asking peoples to see the travel agency ● Agreeing with a travel agency to get a different travel service. ● Making payment for a travel agency. ● Visit the different historical and cultural place ● Book reservation (Accommodation).

2.3. Major Functions of the Existing System

- Register tourists manually first on paper.
- Give information to tourist what he/she wants.
- Generate tourist report.
- Accept tourist complain.

2.4. Forms and Other Documents of the Existing Systems (if any)

There are some forms and documents which are used by the existing system. The forms are used during the period of the travel process. The uses of these forms are to assure the correctness of their activity and to generate reports. In addition to this, these forms are used to increase the relationship with customers. Some of the forms are illustrated below.

Travel Information Form

Please provide us with the contact information below and we will arrange to reserve your ticket.

Title

Contact name
First Name Middle Name Last Name

Birth date
Month Day Year

Organization name

E-mail

Phone number -
Area Code Phone Number

Travel Details

Departure country

Destination country

Departure date & time - -
Month Day Year
 :
Hour Minutes

Return date & time - -
Month Day Year

:
Hour Minutes

Class of service

Please check the services that you need

Airline Tickets
 Rental Car
 Hotel Accomadations

Traveler's Details

Preferred hotel brand

Preferred airline

Additional information

Figure 2. ITravel Information Form

2.5. Drawbacks of the Existing System

In the existing system several drawbacks are existed those are:

- Most activities are done manually so this is vulnerable to man-made and natural disasters.
- Ethiopia tourism bureau cannot easily advertise the tourism sites that are found in Ethiopia
- Tourists do not have enough information about the destination, such as weather, tourism site, cultural place, and different events happening in Ethiopia and also the current destination.
- Tourists cannot get enough information about tourism sites.
- Inaccuracy in updating important information because of manual processing

- To provide information it may take a long time to search the necessary data which is intended for.
- Information is not the time for subsequent use and in a useful format
- High probability of losing important document due to less control over the data
- Since data is stored in a different place, it leads to data inconsistency
- The existing system is not easy to meet new information that is updating currently.
- Tourists can get private guidance by going to a travel agency so it is Exhausting activities for tourists.
- The system is inflexible to change the new or exceptional situation
- Tourists are not able to get travel information timely when they are on the move.

2.6. Business Rules of the Existing System

A business rule is effectively an operating principle or policy the software must satisfy. The Travel Guide Application governs and controls the workflow through the following business rules. These are rules and policy which are used to govern all actives in specified workflow, control the workflow, and performed in the work environment. Our system Travel Guide Application provides the following business rules for the user of the system.

- BR1: -The first rule is that the Traveler/customer should come to the Travel agency office and make some agreement and fill some form to get some travel service and continue other procedures.
- BR2: - A travel agency should be sold travel products and services.
- BR3: - Tourists go with the agency to the tourism bureau ask to view different tour packages.
- BR4: -Tourism Bureau can give full information to the traveler.
- BR5: -Tourist selects the place he/she wants to visit and requests from tour package then fill in personal information to visit any destination/places.
- BR4: -Request for family information if any family.
- BR5: -Request for payment form to fee required amount of money.
- BR6: -the customer recipe from the organization (travel agency).
- BR7: -After that start visiting with a guider.

CHAPTER THREE

3.PROPOSED SYSTEM

In this chapter, we are describing our proposed systems. our proposed system is a mobile-based traveler guide application. This application can develop using the flutter framework to support both android and iOS operating systems. The application can provide detailed information about Ethiopian tourism sites for the user. so to provide this information we use GPS and Google Maps in the application. The proposed traveler guide android application first requires the new user or the tourist to register and if the user has already registered they may log in to their account. Once the user logs in successfully they get the image and description of the selected place. The caching of all the information about the places is made for quick access by tourists without using the external source. The user can also view the map of the selected place for better guidance, the application provides weather forecasting facilities to the user.

The major aim of the application is to guide the tourists to locate different tourist sites across the country and showing contact information that the Tourist can use to contact the respective tourism agent or guider and also this system can contain information for locating the nearest hotel, locating the historical, cultural event and see the weather condition of Ethiopia is possible. This application contains two languages and developed using flutter i.e. It is possible to use both in the android and IOS operating system.

3.1 Functional Requirements

The major functionality of the application is to guide the tourists to locate different tourist sites across the country and showing contact information that the Tourist can use to contact the respective tourism agent or site.

- The system shall support GPS system.
- The system shall support two languages.
- The user shall register in the system.
- The system shall allow travelers can log in to the system to view the different information they want.
- The system shall allow the traveler can create a profile, update, and delete the profile as you want.
- The system shall show the location of the traveler's current location on the map.

- The system shall show the location of the tourism site, hotel, travel agency, contact information of the guider, historical place, and cultural events of Ethiopia they want to travel to.
- The system shall show the weather condition of the city they want to travel to.
- The system should accept only valid data from the users.
- The system shall allow the administrator to add, delete and update newly created tourism site information, tour package, hotel, travel agency, guider, and any other historical and cultural events.
- The system shall allow the administrator to post an advertisement or any other information.
- The system shall allow the traveler to see the hotel details like hotel star, number of room, cost of the room, Types of food they have, and contact no. Of the hotels.
- The system should allow displaying an error message when the user enters invalid data.
- The system shall show essential information of the region (e.g., police and ambulance phones, consulate or embassy contact information);
- The system shall show main country laws and regulations (e.g. Smoking and drinking is prohibited in public areas, regulations for attending churches, taking photos of government building) [8];
- The system shall show the transportation possibilities issue includes public transport schedules, taxi phones,ride sharing possibilities.

3.2. Non-functional Requirements

Nonfunctional requirements capture the required property or qualities of the system. On functional requirements include a broad variety of requirements that apply to many different aspects of the system for usability to performance. The following are the non-functional requirements that will be applied to the proposed system.

3.4.1. User Interface and Human Factors

Our system should provide a mobile user interface which is graphically and usually touch-sensitive display on a mobile device such as smart phone or tablet that allows the user interacts with the device app, features, content, and function this mobile interface is easy to understand, easy to use and user-friendly interface and users of the system easily use and perform their tasks.

We consider the following list while designing our interface:

- Keep the interface simple: We avoid unnecessary elements and keep the user interface clear in the language they use on labels and in message response to the user from the application.
- Create consistency and use common mobile UI elements: By using common elements in our UI, users feel more comfortable and can get things done more quickly, layout and design throughout the site and the application to help facilitate efficiency.
- Be purposeful in page layout: Consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help draw attention to the most important pieces of information and can aid scanning and readability.
- Strategically use color and texture: You can direct attention toward or redirect attention away from items using color, light, contrast, and texture to your advantage, for example, make text or button red color for danger (maybe like for delete button or error message).
- By carefully thinking about and anticipating the goals people bring to our site or application, we create defaults to reduce the burden on the user.
- Level of expertise of the users.
- Our system needs some kind of user expertise to use a travel guide application anyone who can read and write Amharic language or English language he/she can use this system easily otherwise it is difficult to use this system easily.

3.4.2. Hardware Consideration

The Software product to be developed and it run on existing computers , Android and web applicaion. Travel guide application systems can run on any modern hardware device. A phone with android operating system version 9.0 shall be required to run the application on the mobile phone. Application testing using a computer will require Computer storage above 500GB and above 4 GB RAM, monitor with a minimum resolution of 1024x768.

3.4.3. Security Issues

A user must log in to the system with a user name and password. The system allow login to only authorized users. i.e. users that have previously created an account through user name (E-Mail) and password. The system has three groups of users: The Admin,Travel

agency and Traveller .The Admin user has full privilege to perform on the system to add tour package ,post advertisement,manage hotel info. Whereas users can only perform limited operations based on the privilege given by the system. The data that is stored in the firebase is private. The proposed system security developed with hash function algorithm.

3.4.4. Performance Consideration

We are considered to develop the application should be easy and efficient code manipulation and have a clear database. Thus the response time should be very small. Because native android applications are usually run faster than websites and they perform actions much faster too and we use the native android application.

Our system uses firebase as the back end so it supports 200,000 simultaneous connections per database. This limit isn't the same as the total number of users of your app, because your users don't all connect at once. This document primarily covers micro-optimizations that can improve travel guide application performance when combined, but it's unlikely that these changes will result in dramatic performance effects.

3.4.5. Error Handling and Validation

The system will check user inputs to the system to handle the error. It handles and shows errors in a user-friendly manner, without exaggerating the user. We handle these errors on the client-side by using the flutter framework and you can handle them by setting up a Zone. We use front-end validation to reduce the loading time of pages. All errors caught by Flutter have routed to the flutter or. on error handler. When errors occur without a Flutter callback on the call stack, they are handled by the Zone where they occur. By default, a Zone only prints errors and does nothing else. You can customize these behaviors, typically by setting them to values in your void main () function.

3.4.6. Quality Issues

We will develop our project by divide a software system into multiple discrete and independent modules. And then implement them executing in parallel. by increasing cohesion in an element of a module and decreasing coupling of among module, we improve reliability, robustness and reusability features of our system. so Our system will be reliable i.e. It should work its function without any failure as well as if it gets an error the system will perform its function and also the system provides its service for 24 hours per 7 days.

In the quality assessing the users will be involved by a feedback mechanism in which they can give comments and rate on the application.

3.4.7. Backup and Recover

The system includes a secondary database which is a copy of the original database used for a backup purpose that is used for recovery if a problem occurs with the original one.

3.4.8. Physical Environment

The system travel guide application is deployed on a fire base server any of the data is stored in the fire base server so this server works well in all Temperatures any restriction is not given, but there is some problem with the firebase. One of the main problems with it is limited querying capabilities. The real-time database provides no way to filter capabilities because the whole DB is a huge JSON file, which makes it pretty difficult to make complex queries

3.4.9. Resource Issues

Our system well performs its function within a good internet connection it does not give better service if the internet connection is slow or no internet connection. Our system also uses different resources.

- Money for mobile internet.
- Travel guide application will be download on any smartphone, tablet, or any computer which has a NOx player or any player which downloads the application.

3.4.10. Documentation

User-level document is provided that the user can read the document to know how to use the system and what functionality should the system support to the user. Also, the development process is provided that the user can read to know about the process and what type of mode the developer uses to develop the system. We will prepare a short and precise help file on how to use the application for the application user. And also we include Software requirement specification document, Design document.

CHAPTER FOUR

4. SYSTEM ANALYSIS

System analysis is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish its purpose. The analysis model contains three models: functional, object, and dynamic models. The functional model can be described by use case diagrams. Class diagrams describe the object model. The dynamic model can also be described in terms of sequence, state chart, and activity diagrams. For this project, we have described the analysis model in terms of the functional model and dynamic models using use case, sequence diagrams, Activity diagram, State diagram, and class diagram.

4.1. System Model

To describe the abstract models of our system we use the following use case model with each of these use case models presenting of developing abstract models of the system. We can represent our system by using different system models such as use case models, object models, dynamic models, that describe the problem to be solved, and as system models are represented graphically, they are more understandable than more detailed natural language description of the system requirement.

4.1.1. Use Case Model

4.1.1.1 Use Case Diagram

Use case diagrams are used for capturing the functional requirements of the system. It is the functionality of the system or the service provided by the system. In the use case diagram, it better to consider the following elements.

- Use case - which is the functionality of the system which directly interacts with the user.
- Actor - anything which interacts with the system.
- A relationship between the use case.

Our system actor

The system has three actors and each actor can perform actions that are specific to a particular role.

Traveler

- Traveler can log in to the system and log out from the system.
- Traveler can create a profile
- Traveler can view the current location they found.
- Traveler can view the location of tourism site, hotel, travel agency, historical place and the contact information of guider.
- Traveler can view the weather condition of the city.
- Traveler can view Advertisement.

Admin

- Admin can log in and log out from the system.
- Admin can manage hotel information.
- Admin can manage events.
- Admin can give feedback.
- Admin can manage tour package.
- Admin can manage advertisements.

Travel Agency

- Travel Agency can login and logout from the system
- Travel Agency can manage transportation
- Travel Agency can manage hotel
- Travel Agency can also provide other travel services

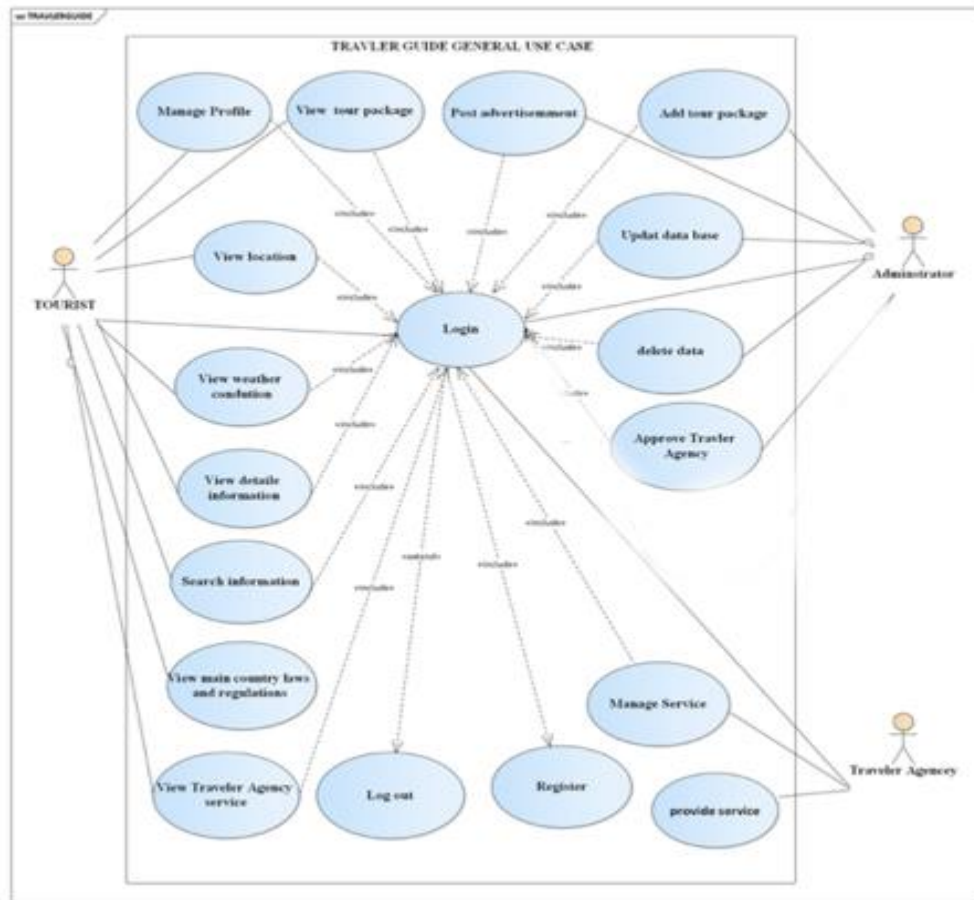


Figure4. 1 Use Case Diagram Of The System

4.1.1.2. Use Case Description

The use case description is used to detail description of the use case what and how the use case works to perform user and system functionality. In this project, we discussed the detail use case description .

Table 4. 1 Use Case Description For Login

Use case name	Login
Use case id	UC_01
Description	This use case describes how the user uses the proposed system to log in to the system.

Actor	<ul style="list-style-type: none"> All users 	
Precondition	<ol style="list-style-type: none"> Users should have a valid username and password (valid account). Users should have a Mobile phone that has this Traveler guide application installed on them. The Traveler and Admin should be registered first 	
The basic flow of events	Actor action	System response
	Step1: Open home page Step2: And then click the login button. Step4: enter user name and password	Step3: display login page Step5: checks user name and password Step6: Now the user has access to his account (logged in) Step7: use case ends
The alternative flow of action	4.1 The user enters invalid username or password <ol style="list-style-type: none"> The system displays an error message (or you entered the wrong username or password will be displayed) The use case continues back to step4. 	
Post condition	<ol style="list-style-type: none"> Successful completion The application shows user homepage according to their privilege Failure condition <ol style="list-style-type: none"> The user doesn't have access to the application if there is no internet connection. The application shows an error message for an invalid username or password and stays on the login page. 	

Table 4. 2 Use Case Description For User Registration

Use case name	Register	
Use case id	UC_02	
Description	This use case describes how the user uses this application to register and get a valid account to log in.	
Actors	<ul style="list-style-type: none"> All users 	
Precondition	Install the application on an android device	
The basic flow of events	Actor action	System response
	Step1: open home page Step2: And then click the registration button. Step4: Fill information	Step3: display page form Step5: checks information. Step6: Now the user has successfully registered and redirects to the login page Step7: use case ends
The alternative flow of action	4.1.If the user entered an invalid information <ol style="list-style-type: none"> The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed. The use case continues back to step4. 	
Post condition	<ol style="list-style-type: none"> Successful completion The application creates and adds a user account to the database. Failure condition <ol style="list-style-type: none"> The user doesn’t have access to the application if there is no internet connection. The application shows an error message for invalid personal information. 	

Table 4. 3 Use Case Description For Creating Profile

Use case name	Create Profile	
Use case id	UC_03	
Description	It allows creating a profile.	
Actor	Traveler	
Precondition	Must have photo and Email address	
The basic flow of events	Actor action	System response
	<p>Step1: open the home page.</p> <p>Step2: click the profile button</p> <p>Step4: Traveler choose photo</p> <p>Step6: Traveler fills information</p> <p>Step7: click the Update profile button.</p>	<p>Step3: The system displays the image upload button to the user.</p> <p>Step5: The system displays profile form</p> <p>Step8: the system displays a success message The profile updated successfully.</p> <p>Step9: The use case ends</p>
The alternative flow of action	<p>6.1 if the user enters an invalid information</p> <p>1. The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed</p> <p>2. The use case continues to step6.</p>	
Post- condition	The Travel Will Create Profile.	

Table 4. 4 Use Case Description For Traveler View Package Tour

Use case name	View Package	
Use case id	UC_04	
Description	It allows viewing packages in the system.	
Actor	Users	
Precondition	Must have an available package	
Basic flow event	Actor action	System response
	<p>Step1: user open the home page.</p> <p>Step2: user click login button</p> <p>Step4: the user fill the form and click login.</p> <p>Step5: user click the view package page link.</p> <p>Step7: user click any package button as you want.</p> <p>Step9:user can click location button and view tourism site, hotel cultural event, historical place, etc.</p>	<p>Step3: the system display login page</p> <p>Step6: the system displays different package tour like tourism site, cultural event, hotel.</p> <p>Step8: the system displays a detail description of that package and the individual component in that package.</p> <p>Step10: the system displays the location of the tourism site, hotel, cultural event, historical place.</p> <p>Step11: the use case ends</p>
Alternative Flow Action	<p>4.1 if the user enters an invalid user name and password</p> <p>1.The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed</p> <p>2. The use case continues to step4.</p>	

Post- condition	<p>1.Successful completion The system shows the selected tour package to the user.</p> <p>2. Failure condition</p> <p>a. The user doesn't have access to the application if there is no internet connection.</p> <p>b. The application shows not found message</p>

Table 4. 5 Use Case Information For Traveler Research Tour Package

Use case name	Search package Traveler Guide application	
Use case id	UC_05	
Description	This use case describes how the user uses this application to get relevant information like tourism site, hotel , Special events.	
Actors	<ul style="list-style-type: none"> Users 	
Precondition	<p>1, Install the application on android device</p> <p>1 The user should have a valid user name and password</p> <p>2 The user should have an internet connection</p>	
The basic flow of events	Actor action	System response
	<p>Step1: user open the home page.</p> <p>Step2: user click the search icon</p> <p>Step4: the user write texts in the text filed</p> <p>Step 5:user enter the search button</p>	<p>Step3: The system displays the search bar</p> <p>Step 6: the system displays the selected Tour package for the user</p> <p>Step 7:use case ends</p>

Alternative flow	<p>4.1: If the entered information is not found</p> <ol style="list-style-type: none"> 1. The system displays the information is not found 2. The use case continues to step4
Post condition	<ol style="list-style-type: none"> 1. Successful completion The application shows the selected tour package to the user. 2. Failure condition <ol style="list-style-type: none"> a. The user doesn't have access to the application if there is no internet connection. b. The application shows not found message

Table 4. 6 Use Case Description For View Location

Use case name	View location	
Use case id	UC_06	
Description	This use case describes how the user uses this application to get location tourism site, hotel , Special events.	
Actors	User	
Precondition	<ol style="list-style-type: none"> 1. Install the application on an android device 2. The user should have a valid user name and password 3. The user should have an internet connection 4 The user should open the location of your mobile phone 	
The basic flow of events	Actor action	System response
	<p>Step1: user open the home page.</p> <p>Step2: user click login button</p> <p>Setp4: user full fill the form and click login</p>	<p>Step3: The system display page</p> <p>Step6:The system displays a list of the tour package</p> <p>Step8: the system displays the selected package.</p> <p>Step10: the system displays the location of the package</p>

	<p>Step5: user click tour package button</p> <p>Step7:The user select package from the list</p> <p>Step 9:user click to view location button</p>	Step11:use case ends
The alternative flow of Action	<p>4.1 if the user enters an invalid email and password</p> <p>1.The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed</p> <p>2. The use case continues to step4.</p>	
Post condition	<p>1. Successful completion</p> <p>The application shows the exact location of selected tour package to the user.</p> <p>2. Failure condition</p> <p>a. The user doesn’t have access to the application if there is no internet connection.</p> <p>b. The application shows not found message</p>	

Table 4. 7 Use Case Description For Traveler View Weather Condition

Use case name	View weather condition
Use case id	UC_07
Description	This use case describes how the user uses this application to get the weather condition of tourism site city.
Actors	<ul style="list-style-type: none"> • User
Precondition	<p>1. Install the application on an android device</p> <p>2. The user should have a valid user name and password</p>

	<p>3. The user should have an internet connection</p> <p>4 The user should open the location of your mobile phone</p>	
The basic flow of events	Actor action	System response
	<p>Step1: open the home page.</p> <p>Step2 user clicks login button</p> <p>Step4 the user full fill the form and click login</p> <p>Step5: user clicks view weather condition</p> <p>Step7: The user select city from the list</p> <p>Step9: user click view current weather condition button</p>	<p>Step3: the system display form</p> <p>Step6:The system displays the list of the city in Ethiopia</p> <p>Step8: the system displays the selected city.</p> <p>Step10: the system displays the weather condition of the selected city</p> <p>Step11:use case ends</p>
Alternative flow	<p>4.1 if the user enters an invalid email and password</p> <p>1.The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed</p> <p>2. The use case continues to step4.</p>	
Post condition	<p>1. Successful completion</p> <p>The application shows the weather condition of a selected city</p> <p>2. Failure condition</p> <p>a. The user doesn’t have access to the application if there is no internet connection.</p> <p>b. The application shows not found message</p>	

Table 4. 8 Use Case Description For Admin Add Tour Package

Use case name	Add tour package.	
Use case id	UC_09	
Description	This use case allows the addition of a new tour package to the system.	
Actor	TourismBureau.	
Pre-condition	<ol style="list-style-type: none"> 1. Install the application on an android device 2. The TourismBureau should have a valid user name and password 3. The TourismBureau should have an internet connection 4 The TourismBureau should open the Add Tour Package button 	
Basic flow Events	Actor action	System response
	<p>Step1: open home page</p> <p>Step2 : TourismBureau click login button</p> <p>Step4: TourismBureau full fill the form and click login</p> <p>Step5: TourismBureau click add tour package link.</p> <p>Step7: TourismBureau select add tour package list.</p> <p>Step9: The TourismBureau fills the required data and click submits.</p>	<p>Step3: the system display login form</p> <p>Step6: The system displays the package tour list.</p> <p>Step8: the system display form.</p> <p>Step9: The system displays message successfully added.</p> <p>Step10: end-use case.</p>
The alternative flow of action	<p>Step9.1. If the user enters invalid information.</p> <ol style="list-style-type: none"> 1. System display please fill the form with in correct information. 2. The use case continues to step9. 	

Post-condition	<ol style="list-style-type: none"> 1.Successful completion The application shows successful the tour package is added. 2. Failure condition <ol style="list-style-type: none"> a. The tour package is not added if the internet connection is lost. b. The application shows package is not added
----------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 4. 9 Use Case Description For Admin Post Advertisement

Use case name	Post Advertisement	
Use case id	UC_10	
Actor	TourismBureau.	
Description	This use case allows posting the TourismBureau Advertise newly created tourism site, hotel, travel agency, guider.	
Pre-condition	<ol style="list-style-type: none"> 1. Install the application on an android device 2. The TourismBureau should have a valid user name and password 3. The TourismBureau should have an internet connection 4 The TourismBureau should open the Post Advertisement button 	
The basic flow of events	Actor action	System response
	Step1: The TourismBureau login to the system. Step2: The TourismBureau clicks the post info link. Step4: TourismBureau fills info. Step5: Submit the form.	Step3: The system displays the required form. Step6: The system post info and displays message successfully posted. Step8: use case end.
The alternative flow of action	Step4.1 If the user enters invalid information. <ol style="list-style-type: none"> 1. System display please fill the form with valid information. 2. The use case continues to step 4. 	

Post-condition	<ol style="list-style-type: none"> 1.Successful completion The application shows success the Advertisement is added. 2. Failure condition <ol style="list-style-type: none"> a. The Advertisement is not added if the internet connection is lost. b. The application shows no new posted information is added
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 4. 10 *Use Case Description For Admin Delete Tour Package*

Use case name	Delete tour package	
Use case id	UC_11	
Description	This use case allows deletion of an existing tour package.	
Actor	TourismBureau.	
Use case name	Delete tour package	
Pre-condition	<ol style="list-style-type: none"> 1. Install the application on an android device 2. The TourismBureau should have a valid user name and password 3. The TourismBureau should have an internet connection 4 The tour package must have been available. 	
Basic flow of Events	Actor action	System response
	Step1: open the home page. Step2: TourismBureau click login button Step4: TourismBureau full fill the form and click login Step5: TourismBureau click the delete tour package link Step7: The TourismBureau selects delete tour package. Step9: TourismBureau click the delete button Step11: the user clicks ok to confirm	Step3: the system display page the system displays the list of package Step6: the system displays the Exiting tour package. Step8:the system display selected package Step10: the system displays are you want to delete this package Step12: the system displays a message successfully deleted. Step13: end-use case.

Alternative cores of action	<p>4.1 if the TourismBureau enters an invalid email and password</p> <ol style="list-style-type: none"> 1. The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed 2. The use case continues to step4.
Post-condition	<ol style="list-style-type: none"> 1. Successful completion The application shows a successful delegation of a tour package. 2. Failure condition <ol style="list-style-type: none"> a. The tour package is not successfully deleted if the internet connection is lost. b. The application shows the existing package is not deleted.

Table 4. 11 *Use Case Description For Admin Deleted Posted Information*

Use case name	Delete Post Advertisement	
Use case id	UC_12	
Description	This use case allows deleting post Advertisement that the admin posted before like tourism site, hotel, travel agency, guider.	
Actor	TourismBureau.	
Pre-condition	<ol style="list-style-type: none"> 1. Install the application on an android device 2. The TourismBureau should have a valid user name and password 3. The TourismBureau should have an internet connection 4 The post Advertisement must have been available 	
Basic cores Action	Actor action	System response
	Step1: TourismBureau open home page Step2: TourismBureau clicks login button	Step3: The system display page Step6: The system displays posted information.

	<p>Step4: TourismBureau full fill the form</p> <p>Step5: TourismBureau clicks the delete post info link.</p> <p>Step7: TourismBureau selects the posted information.</p> <p>Step9: TourismBureau click the deleted button.</p> <p>Step11: TourismBureau click ok to delete information</p>	<p>Step8: The system display selected post info</p> <p>Step10: the system displays are you want to delete this information</p> <p>Step12: The system displays a message successfully deleted.</p> <p>Step13: use case end.</p>
Alternative cores of action	<p>4.1 if the TourismBureau enters an invalid email and password</p> <p>1.The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed</p> <p>2. The use case continues to step4.</p>	
Post-condition	<p>1.Successful completion</p> <p>The application shows success the Advertisement is deleted.</p> <p>2. Failure condition</p> <p>a. The Advertisement is not successfully deleted if the internet connection is lost.</p> <p>b. The application shows existing advertisement is not added</p>	

Table 4. 12 *The Use Case Description For Adding Travel Service*

Use case name	Add travel service
Use case id	UC_14
Description	This use case allows the addition of a new transport service to the system.
Actor	Travel agency.
Pre-condition	1. Install the application on an android device

	<p>2. The Travel agency should have a valid user name and password</p> <p>3. The Travel agency should have an internet connection</p> <p>4 The Travel agency should open the Add transport service button</p>	
Basic flow	Actor action	System response
Events	<p>Step1: login.</p> <p>Step2: click add transport service link.</p> <p>Step4: Travel agency select add travel service type list.</p> <p>Step6: The Travel agency fills in the required data.</p> <p>Step7: click submits.</p>	<p>Step3: The system displays the package tour list.</p> <p>Step5: the system display form.</p> <p>Step8: validate the information</p> <p>Step9: The system displays message successfully added.</p> <p>Step10: end-use case.</p>
The alternative flow of action	<p>Step6.1. If the Travel agency enters invalid information.</p> <ol style="list-style-type: none"> 1. System display please fill the form with in correct information. 2. The use case continues to step6. 	
Post-condition	<p>1.Successful completion</p> <p>The application shows successful the travel service is added.</p> <p>2. Failure condition</p> <ol style="list-style-type: none"> a. The travel service is not added if the internet connection is lost. b. The application shows no new travel service is added 	

Table 4. 13 *The Use Case Description For Successful Deletion Of Travel Service*

Use case name	Delete travel service
Use case id	UC_15
Description	This use case allows deletion of an existing travel service.
Actor	Travel agency.

Use case name	Delete travel service	
Pre-condition	<ol style="list-style-type: none"> 1. Install the application on an android device 2. The Travel agency should have a valid user name and password 3. The Travel agency should have an internet connection 4 The travel service must have been available. 	
Basic flow of	Actor action	System response
Events	<p>Step1: travel agency open the home page.</p> <p>Step2:travel agency click login button</p> <p>Step4: travel agency full fill the form and click login</p> <p>Step5: travel agency click the delete travel service link</p> <p>Step7: travel agency select type of travel service.</p> <p>Step9: travel agency click the delete button</p> <p>Step11: the user clicks ok to confirm</p>	<p>Step3: the system display login form</p> <p>Step6: the system displays the Exiting travel service.</p> <p>Step8:the sysem display selected travel service.</p> <p>Step10: the system displays are you want to delete this travel service</p> <p>Step12: he system displays a message successfully deleted.</p> <p>Step11: end-use case.</p>
Alternative cores of action	<ol style="list-style-type: none"> 4.1 if the Travel agency enters an invalid email and password 1.The System displays messages that contain “field cannot be empty or you entered a wrong value” will be displayed 2. The use case continues to step4. 	
Post-condition	<ol style="list-style-type: none"> 1.Successful completion The application shows a successful deletion of a travel service . 2. Failure condition a. The travel service is not successfully deleted if the internet connection is lost. 	

	b. The application shows the travel service is not deleted
--	------------------------------------------------------------

4.1.1.3. Use case Scenario

A use case scenario, or scenario for short, describes a real-world example of how one or more people or organizations interact with a system. They describe the steps, events, and/or actions which occur during the interaction. Use case scenarios can be very detailed, indicating exactly how someone works with the user interface, or reasonably high-level describing the critical business actions but not the indicating how they're performed.

Scenario name: Login.

Participating actors: Users

Initial assumption: internet connection is available. Also, the application is open.

The normal flow of events:

- 1.Users open home page
- 2.Users clicks on the (login) link on the home page.
3. The System display the login page.
- 4.User fulfills the username and password and then clicks on the (Login) Button.
- 5.The System validates input value.
- 6.User has access to his account (logged in).
- 7.End-use case.

Alternative flow:

- 4.1.The system displays messages that contain “field cannot be empty or you entered wrong username or password” will be displayed.

Scenario name: User registration.

Participating actors: Users

Initial assumption: traveler application system should be first displayed.

The normal flow of events:

1. Users clicks on the (register) link on the home page.
- 2.The system display the registration page.
- 3.Users fulfill the information that the system request and click on (register) button.

- 4.The system validates entered value from the user.
- 5.The system Registers the users and displays a Successfully registered message.
- 6.The system display the Login page.
- 7.Use case end

Alternative flow:

3.1.Users enters incorrect information and the system displays messages that contain “field cannot be empty or you entered wrong value” will be displayed.

Scenario name: Create Profile.

Participating actors: Users.

Initial assumption: Users is logged in into the system and on Create profile.

The normal flow of the event

1. Users click on a link from create a profile.
- 2.The system displays create profile form
3. Users fulfill the form and click upload profile
- 4.The system can upload the profile.
5. Use case end.

Alternative flow:

3.1. Users doesn't fulfill the fields and enter invalid information like incorrect Email.The system displays messages that contain “field cannot be empty or you entered wrong value” will be displayed.

Scenario name: View Package.

Participating actors: Users.

Initial assumption: Kediri is logged in into the system and View package.

The normal flow of an event

1. Users click on a link from view Package.
- 2.The system displays different package
3. Users select any type of package from the system
- 4.The system can view the selected package.

5. Use case end.

Alternative flow:

3.1.If Users don't select the appropriate package and if there is no internet connection.The system display you select an incorrect package and there is no internet.

Scenario name: Add package

Participating actors: TourismBureau.

Initial assumption: Kediri is logged in into the system and view Add package

The normal flow of an event

1. TourismBureau click on a link from Add package.
- 2.The system displays different package
3. TourismBureau select any type of package from the system
- 4.The system can view the selected package.
- 5.Then TourismBureau can add a new package from the selected package
5. Use case end.

Alternative flow:

3.1.If TourismBureau don't select the appropriate package and if there is no internet connection.The system displays in an appropriate package and there is no internet.

Scenario name: Delete Package.

Participating actors: TourismBureau.

Initial assumption: TourismBureau is logged in into the system and view the delete package

The normal flow of an event

1. TourismBureau click on a link from the Delete package.
- 2.The system displays different package
3. TourismBureau select any type of package from system
- 4.The system can view the selected package.
- 5.Then TourismBureau can delete the existing package from the selected package
6. Use case end.

Alternative flow:

3.1.If TourismBureau don't select the appropriate package and if there is no internet connection.The system displays in an appropriate package and there is no internet.

Scenario name: post advertisement.

Participating actors: TourismBureau.

Initial assumption: TourismBureau is logged in into the system and view post advertisement

The normal flow of an event

- 1.TourismBureau click on a link from post info.
- 2.The system displays advertisement form
- 3.TourismBureau can full fill the form and click the post info link
- 4.The system displays the advertisement is posted correctly
5. Use case end.

Alternative flow:

3.1.If TourismBureau don't fully fill the form correctly and if there is no internet connection.The system displays please full fill the form before posted and there is no internet.

Scenario name: Add travel service.

Participating actors: Travel agency.

Initial assumption: Travel agency is logged in into the system and view Add travel service

The normal flow of an event

1. Travel agency click on a link from Add travel service.
- 2.The system displays different travel service
3. Travel agency select any type of travel service from the system
- 4.The system can view the selected travel service form.
- 5.Then Travel agency can fill the form and click submite.
- 6.The system display travel service successfully added.
7. Use case end.

Alternative flow:

3.1.If Travel agency don't select the appropriate travel service and if there is no internet connection.The system displays inappropriate travel service and there is no internet.

Scenario name: Delete travel service.

5.1 .Travel agency doesn't fulfill the fields and enter invalid information.The System displays Messages that contain "field cannot be empty or you entered wrong value" will be displayed.

Participating actors: Travel agency.

Initial assumption: Travel agency is logged in into the system and view the delete travel service

The normal flow of an event

1. Travel agency click on a link from the Delete travel service.
- 2.The system displays different travel service list
3. Travel agency select any type of travel service list from system
- 4.The system show the selected travel service.
- 5.Then Travel agency can delete the existing travel service from the selected travel service
6. Use case end.

Alternative flow:

3.1.If Travel agency don't select the appropriate travel service and if there is no internet connection.The system displays inappropriate travel service and there is no internet.

4.2. Object Model

An object model is a logical interface, software, or system that is modeled through the use of object-oriented techniques. It enables the creation of an architectural software or system model before development or programming. In our system, we use object model like class diagram and data dictionary.

4.2.1. Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of travel guide Application.

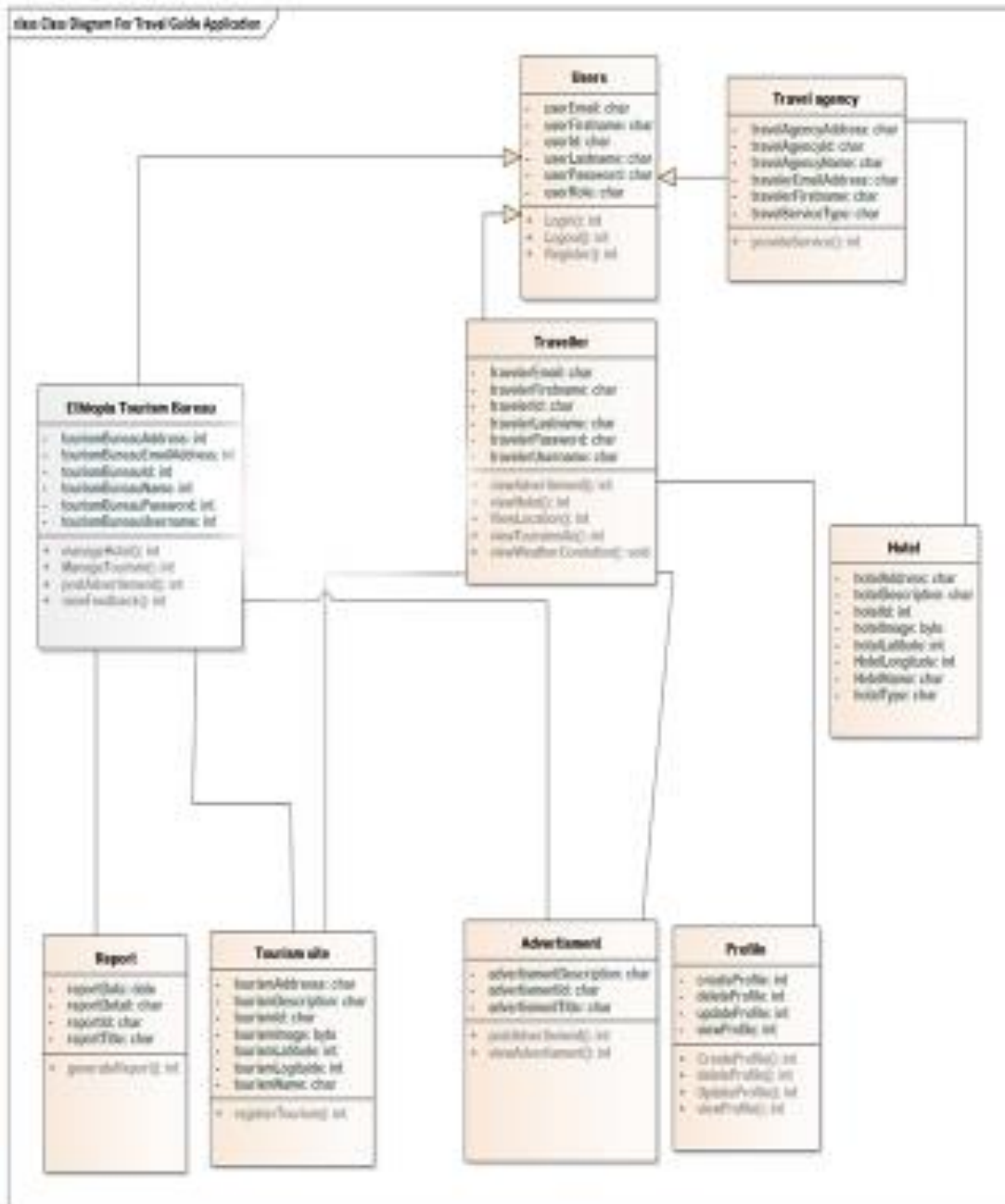


Figure4. 2 class diagram

4.2.2. Data Dictionary

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data. In our travel guide application system,

the following table describes the data dictionary. Table 4.14 below illustrates the data dictionary of our system.

Table 4. 14 Data Dictionary For A Traveler

Table name: Traveler				
Primary key: Traveler_Id				
Description: used to store information related to Traveler				
Attributes	Data type	Data size	Constraint	Description
TravelerId	String	30	Primary key	Unique identification for Traveler
TravelerFirstname	String	40	Not Null	Name of Travler
TravelerLastname	String	40	Not Null	Last Name of Traveler
TravelereMailaddress	String	50	No null	The email address for traveler
TravelerPassword	String	30	Not null	Traveler password for registration and login to the system
TravelerUsername	String	30	Not null	User name of Traveler
TravelerPhoneNumber	Int	30	Not null	Phone number of the traveler

Table 4. 15 *Data Dictionary For Tourism Site*

Table name: Tourism site				
Primary key: tourism_Id				
Description: used to store information related to tourism sites				
Attributes	Data type	Data size	Constraint	Description
TourismId	String	30	Primary key	Unique identification for tourism site
TourismName	String	40	Not Null	Name of tourism
TourismAddress	String	40	Not Null	Address of tourism site in Ethiopia
TravelerId	String	50	Foreign key	Unique identification from traveler primary key
TourismDescription	String	200	Not null	The description of tourism site
TourismImage	Byte		Not null	Image of tourism site
TourismBureauID	String	30	Foreign key	Unique identification from TourismBureau primary key
TourismLongitude	Int	50	Not null	The longitude location of tourism site
TourismLatitude	Int	50	Not null	The latitude location tourism site

Table 4. 16 *Data Dictionary For Weather Condition*

Table name: Weather condition				
Primary key: weatherId				
Description: used to store information related to weather				
Attributes	Data type	Data size	Constraint	Description

WeatherId	String	30	Primary key	Unique identification for weather conditions
City name	String	40	Not Null	Name of city
WeatherDetail	String	80	Not Null	Detail information for weather condition
TravelerId	String	50	Foreign key	Unique identification from traveler primary key

Table 4. 17 Data Dictionary For Admin

Table name: TourismBureau				
Primary key: TourismBureauID				
Description: used to store information related to TourismBureau				
Attributes	Data type	Data size	Constraint	Description
TourismBureauID	String	30	Primary key	Unique identification for Admin
TourismBureauName	String	40	Not Null	The name of TourismBureau
TourismBureauAddress	String	40	Not Null	The address of TourismBureau
TourismBureauEmailAddress	String	50	No null	The email address for TourismBureau
TourismBureauPassword	String	30	Not null	TourismBureau password for registration and login to the system
TourismBureauUsername	String	30	Not null	User name of TourismBureau

Table 4. 18 *Data Dictionary for Hotel*

Table name: Hotel				
Primary key: hotel				
Description: used to store information related to hotel				
Attributes	Data type	Data size	Constraint	Description
HotelId	String	30	Primary key	Unique identification for Hotel information
HotelName	String	40	Not Null	Name of hotel
HotelAddress	String	40	Not Null	Address of bset hotel in Ethiopia
TravelerId	String	50	Foreign key	Unique identification from traveler primary key
HotelDescription	String	80	Not null	The description of Hotel
HotelImage	Byte		Not null	Image of Hotel with its star
AdminId	String	30	Foreign key	Unique identification from Admin primary key
TourismLongitude	Int	50	Not null	The longitude location of hotel
TourismLatitude	Int	50	Not null	The latitude location hotel
HotelType	String	40	Not null	Type of hotel

Table 4. 19 *Data Dictionary For Advertisement*

Table name: Advertisement				
Primary key: AdvertisementId				
Description: used to store information that related to Advertisement				
Attributes	Data type	Data size	Constraint	Description
AdvertisementId	String	30	Primary key	Unique identification for Advertisement
TourismBureauId	String	40	Foreign key	Unique identification from TourismBureau primary key
AdvertismentTitle	String	80	Not Null	Title of Advertisement
TravelerId	String	50	Foreign key	Unique identification from traveler primary key
Advertisement Description	String	VARCHAR (n)	Not Null	Bestdescription for Advertisement of tourism site, hotel, historical place, etc.

Table 4. 20 *Data Dictionary For Travelagency*

Table name: travel agency				
Primary key: TravelagencyId				
Description: used to store information related to weather				
Attributes	Data type	Data size	Constraint	Description
TravelagencyId	String	30	Primary key	Unique identification for Travel Agency
TravelagencyName	String	40	Not Null	Name of Agency
ServiceType	String	50	Not Null	Type of service provided by Agency

TravelerId	String	50	Foreign key	Unique identification from traveler primary key
Description	String	80		The best description for travel agency
TourismBureauId	String	40	Foreign key	Unique identification from TourismBureau primary key

Table 4. 21 *Data Dictionary For Transportation*

Table name: Transportation					
Primary key: TransportationId					
Description: used to store information related to Guider					
Attributes	Data type	Data size	Constraint	Description	
TransportationId	String	30	Primary key	Unique identification for Transportation	
TransportationName	String	50	Not null	Name of Transportation	
TransportationType	String	40	Not null	Transportation type for a traveler	
TravelagencyTd	String	50	Foreign key	Unique identification from Travel Agencyprimary key	

4.3. Dynamic Model

A dynamic model represents the behavior of an object over time. It is used where the object's behavior is best described as a set of states that occur in a defined sequence. In our system we use sequence diagram, activity diagram, state diagram.

4.3.1. Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the

objects needed to carry out the functionality of the scenario so in the Travel Guide Application there will be a different process to do specific actions and we include some sequence diagrams to handle these interactions.

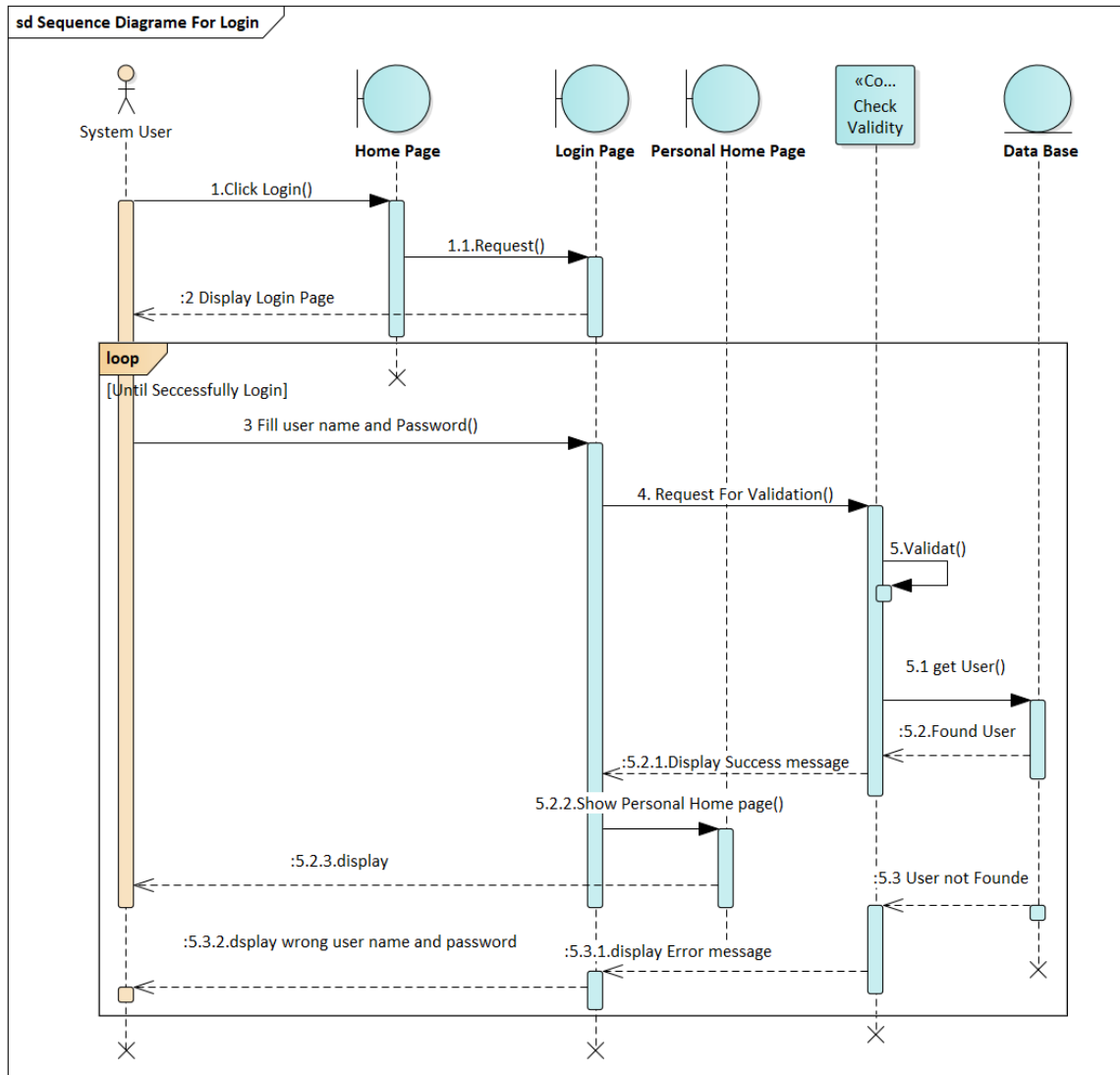


Figure4. 3 Sequence Diagram For Login

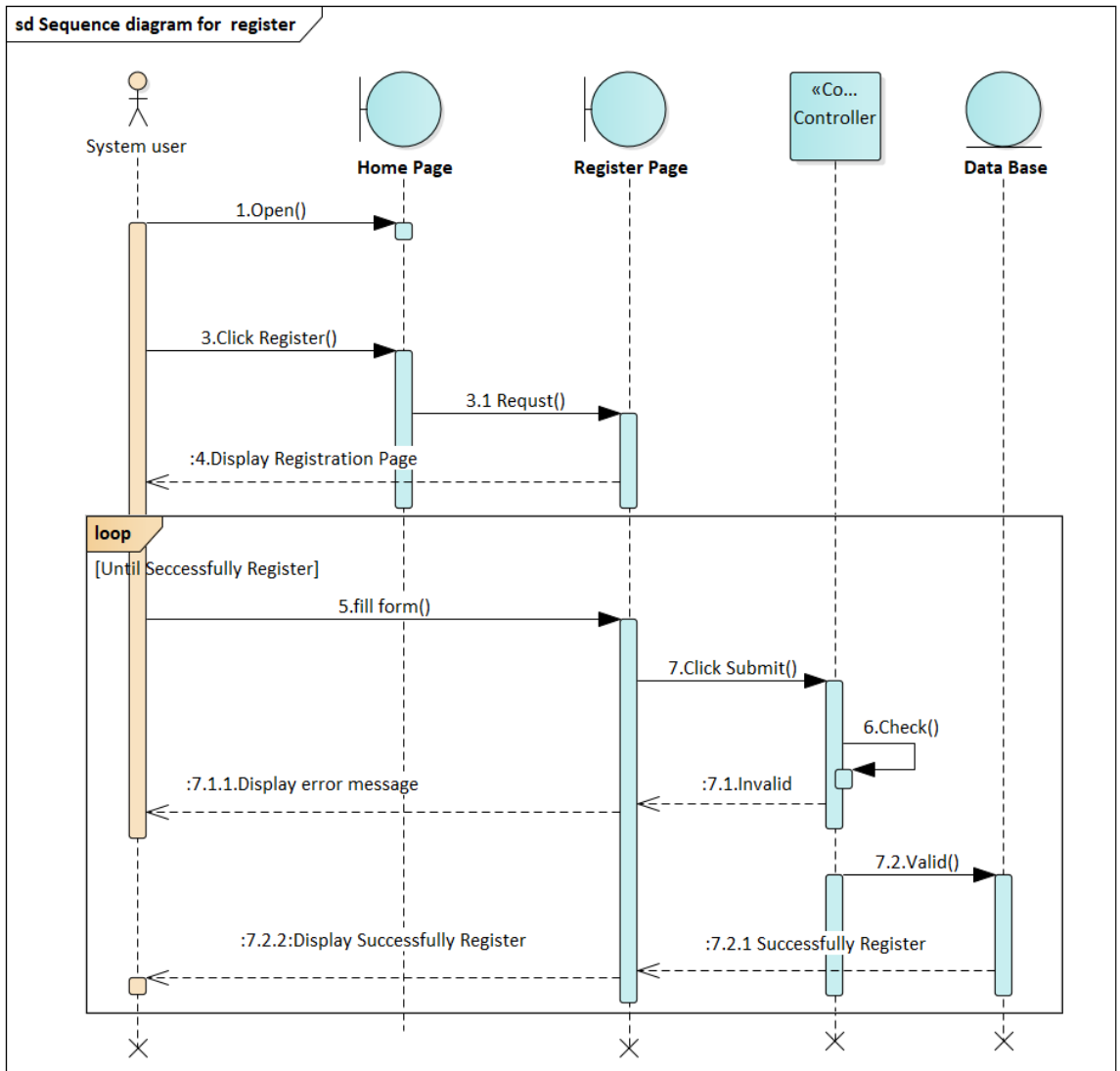


Figure4. 4 Sequence Diagram For Register

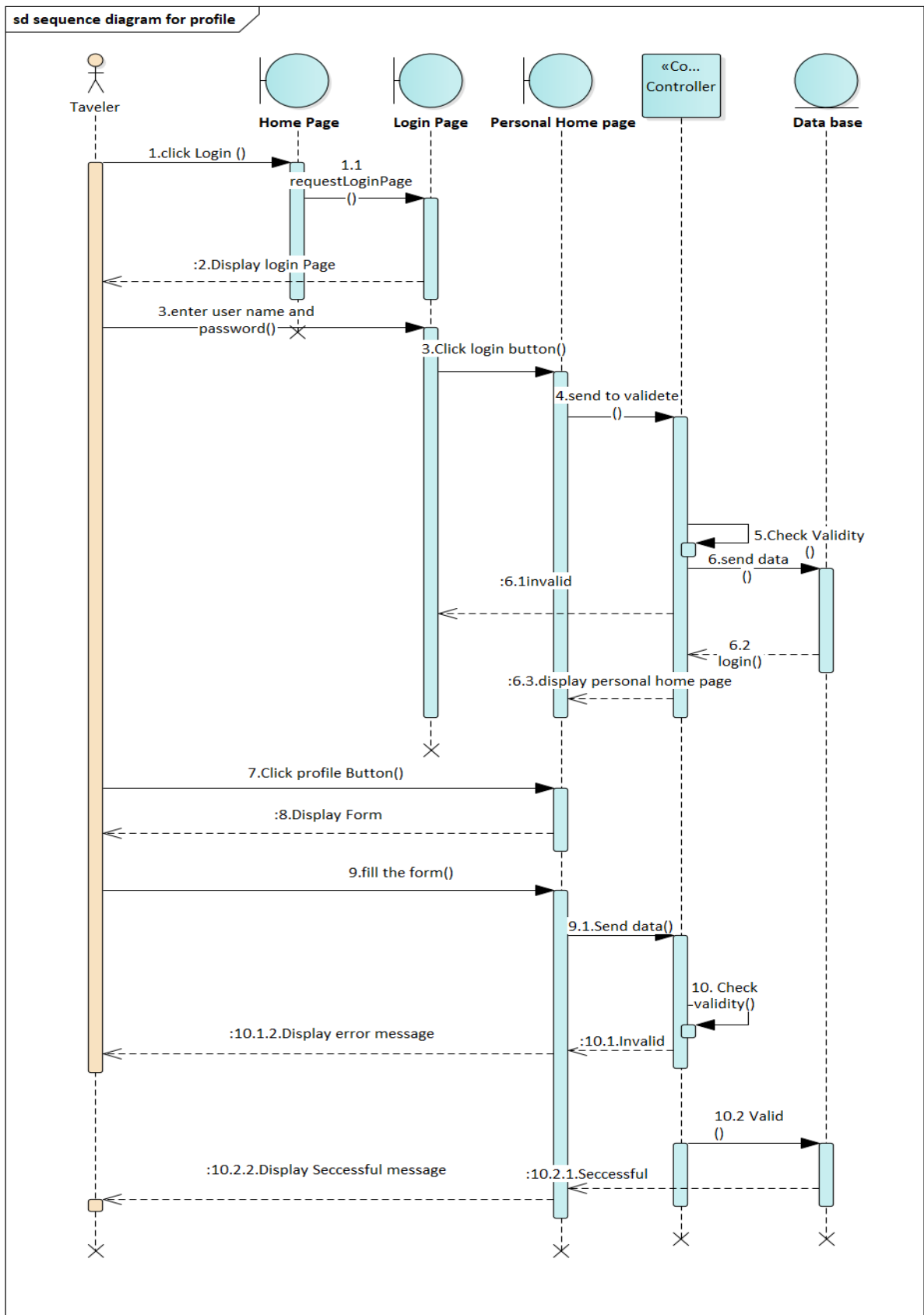


Figure4. 5 Sequence Diagram For Profile

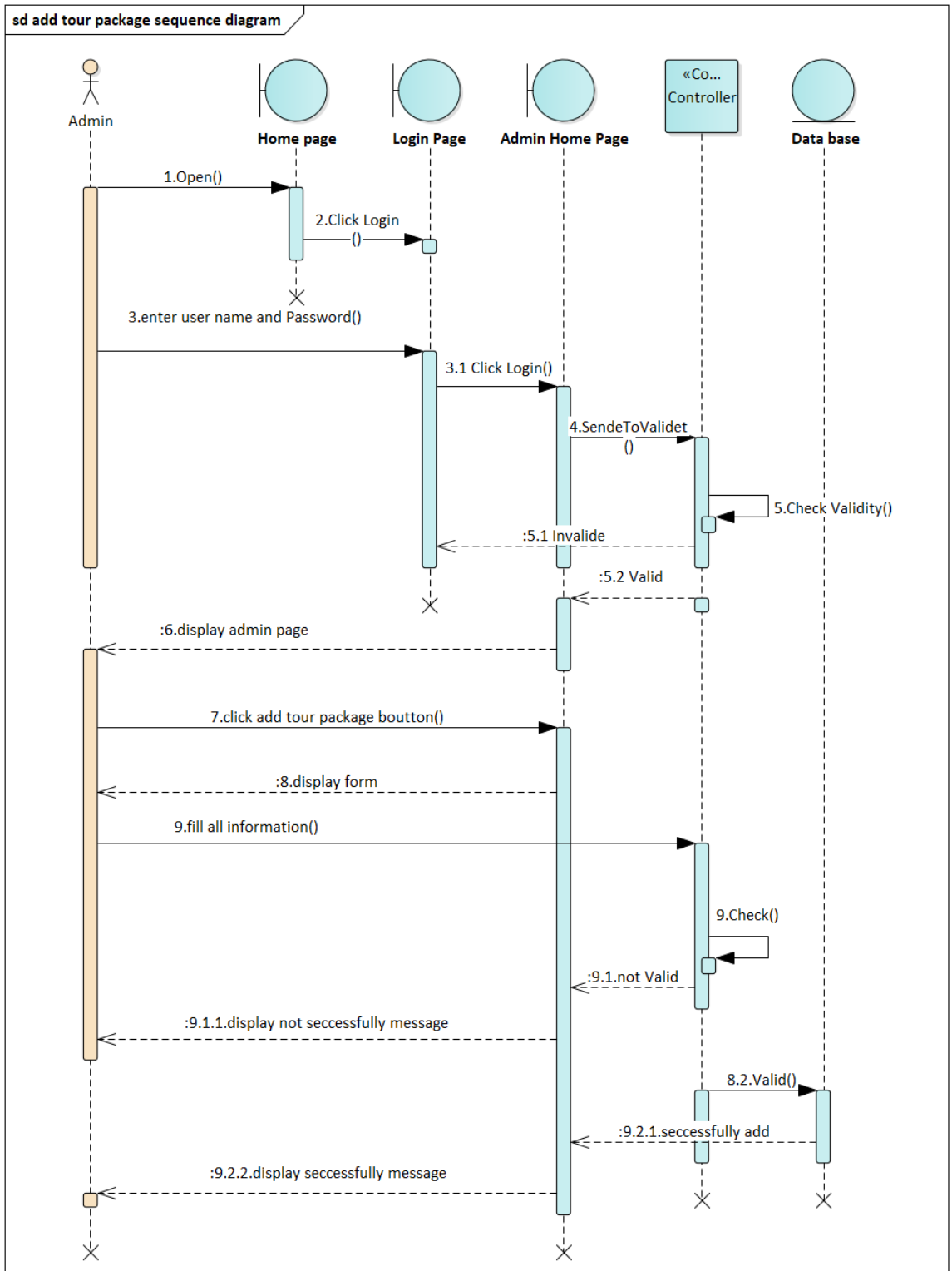


Figure4. 6 Sequence Diagram For Add Tour Package

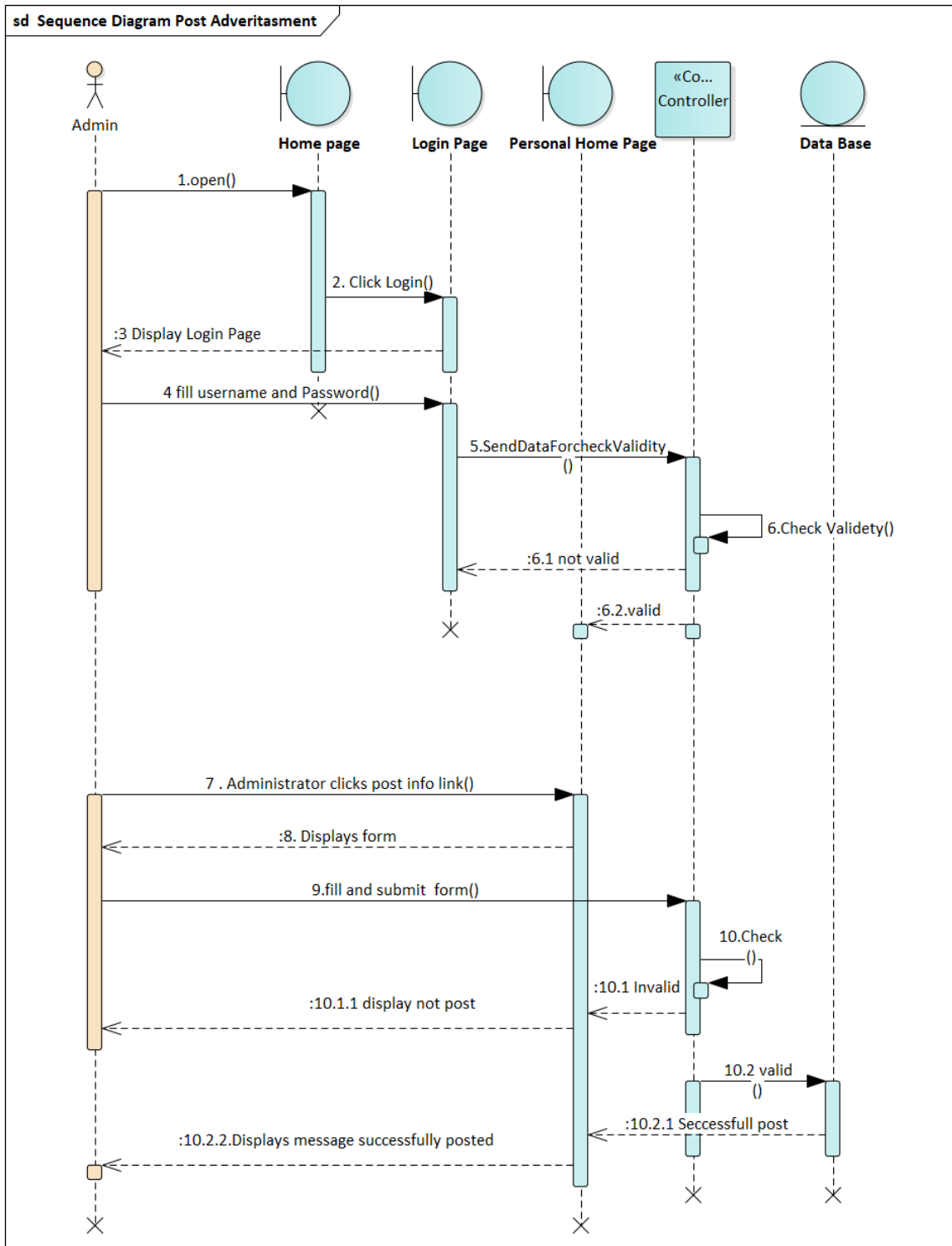


Figure4. 7 Sequence Diagram For Post Advertisement

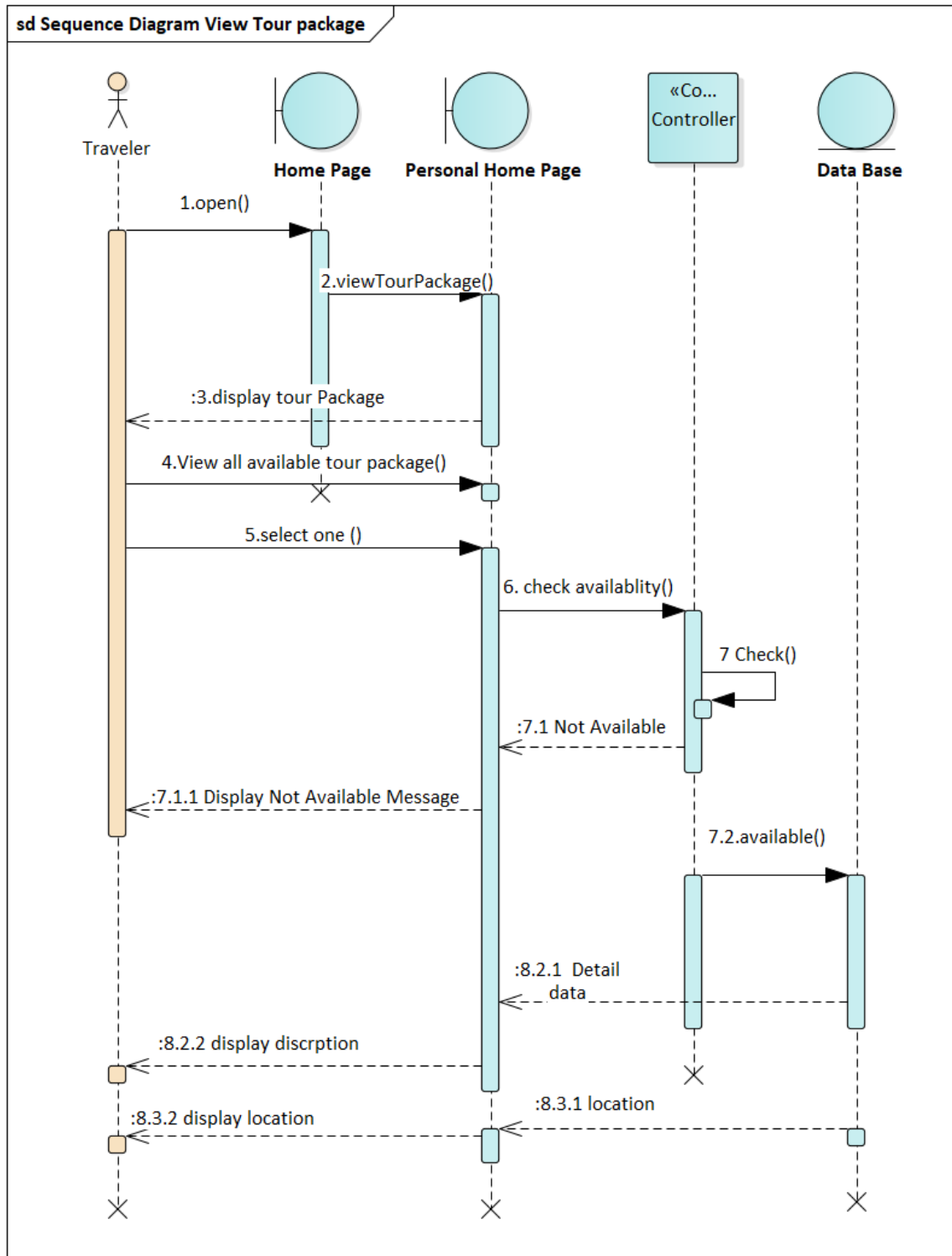


Figure4. 8 Sequence Diagram For View Tour Package

4.3.2. Activity Diagram

An activity diagram is a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system to perform many tasks there should be one or more activities to be done.

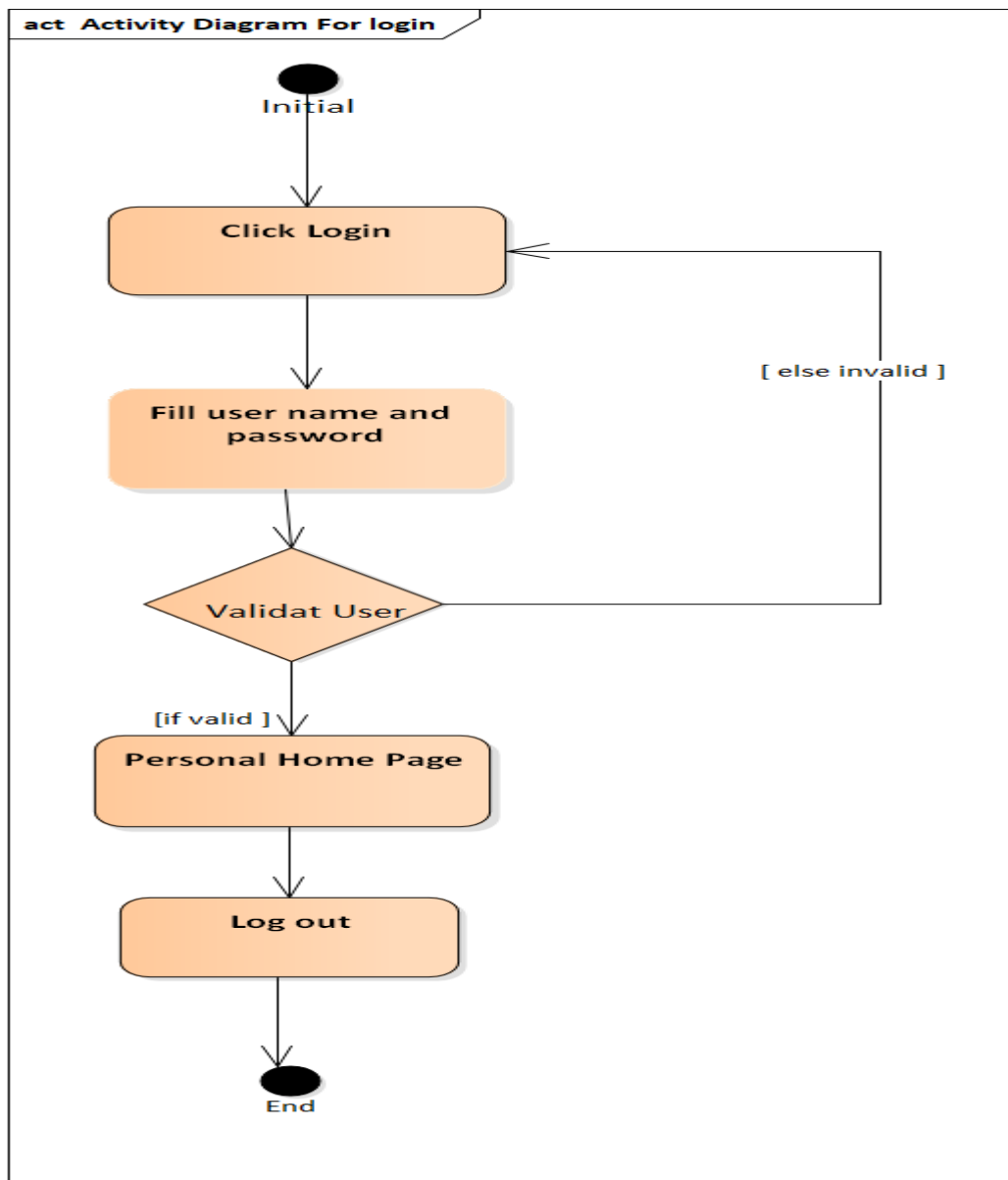


Figure4. 9 Activity Diagram For Login

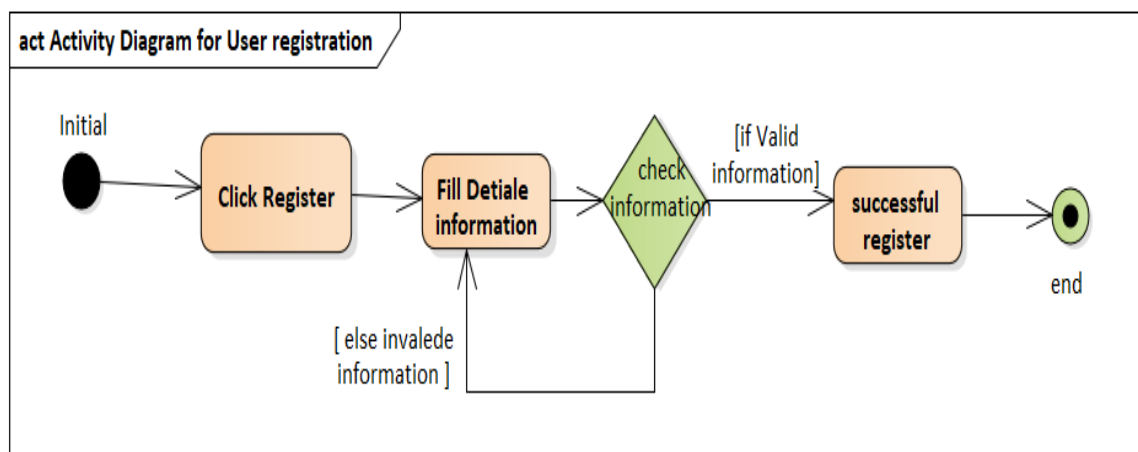


Figure4. 10 Activity Diagram For User Registration

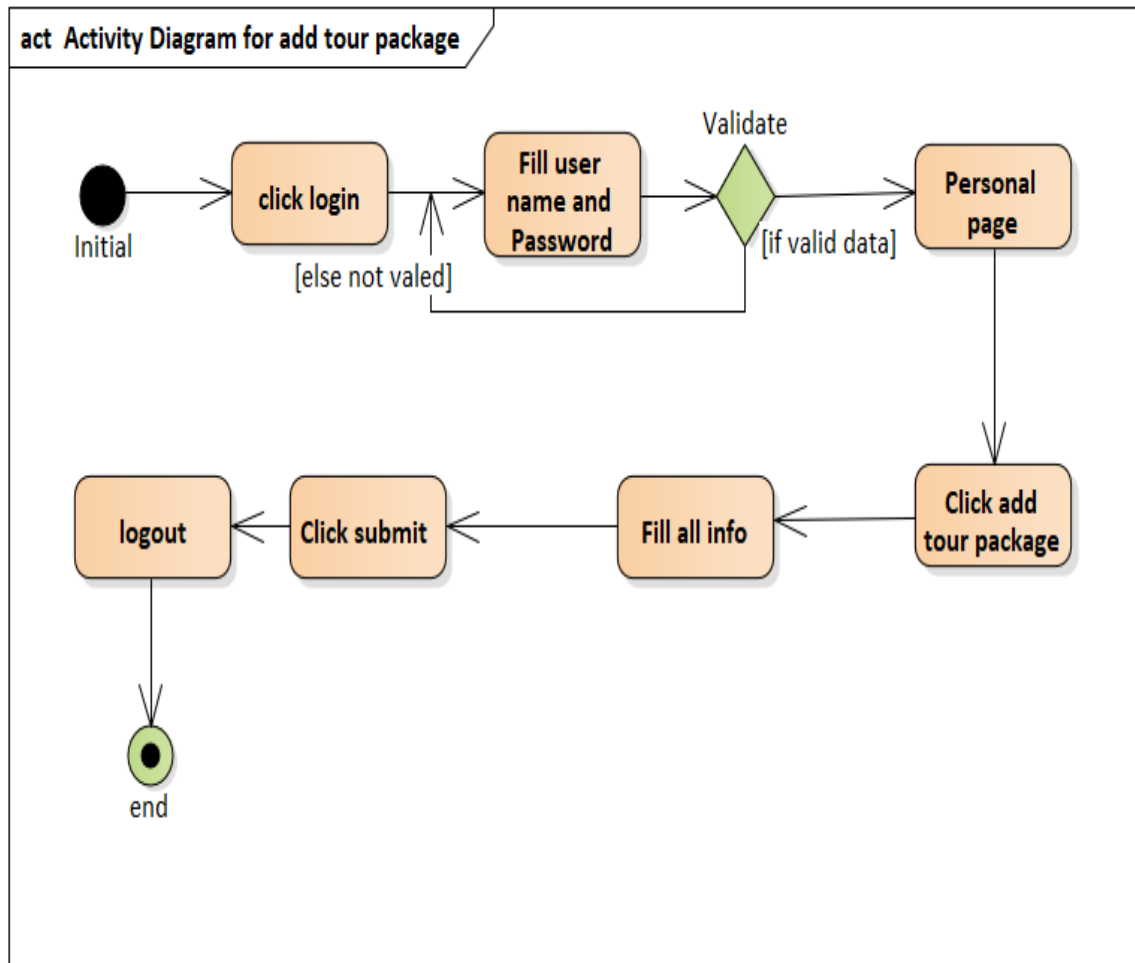


Figure4. 11 Activity Diagram For Add Tour Package

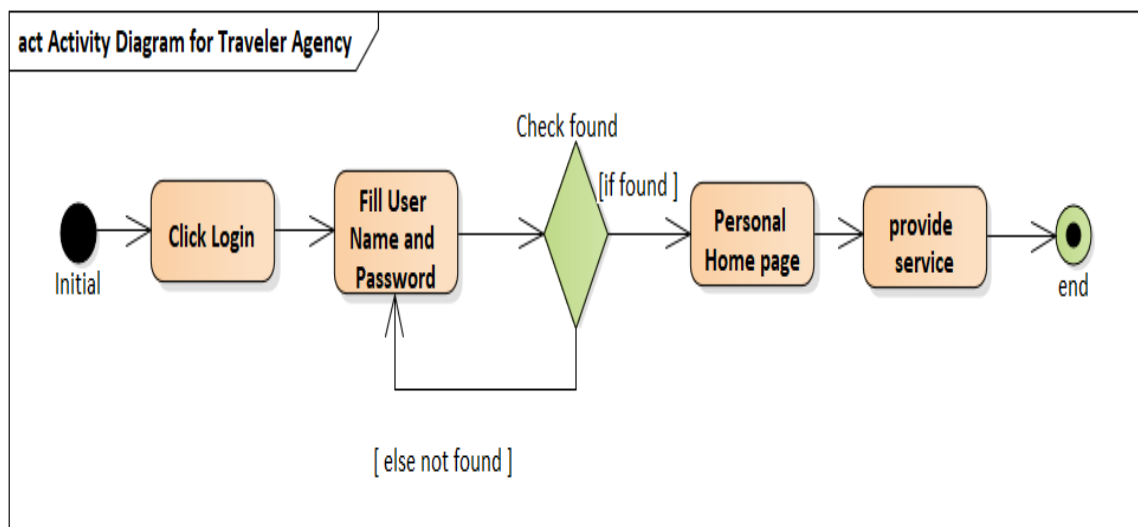


Figure 4. 12 Activity Diagram For Provid Service

4.3.3. State Chart Diagram

State chart diagram describes the flow of control from one state to another state. States are Defined as a condition in which an object exists and it changes when some event is triggered. So, the most important purpose of the State chart diagram is to model lifetime of an object from Creation to termination whenever performing different activities there will be some states that Are included in activities so in our project Travel Guide Application we have listed below Some states can be covered during the user interacts with the system.

Login

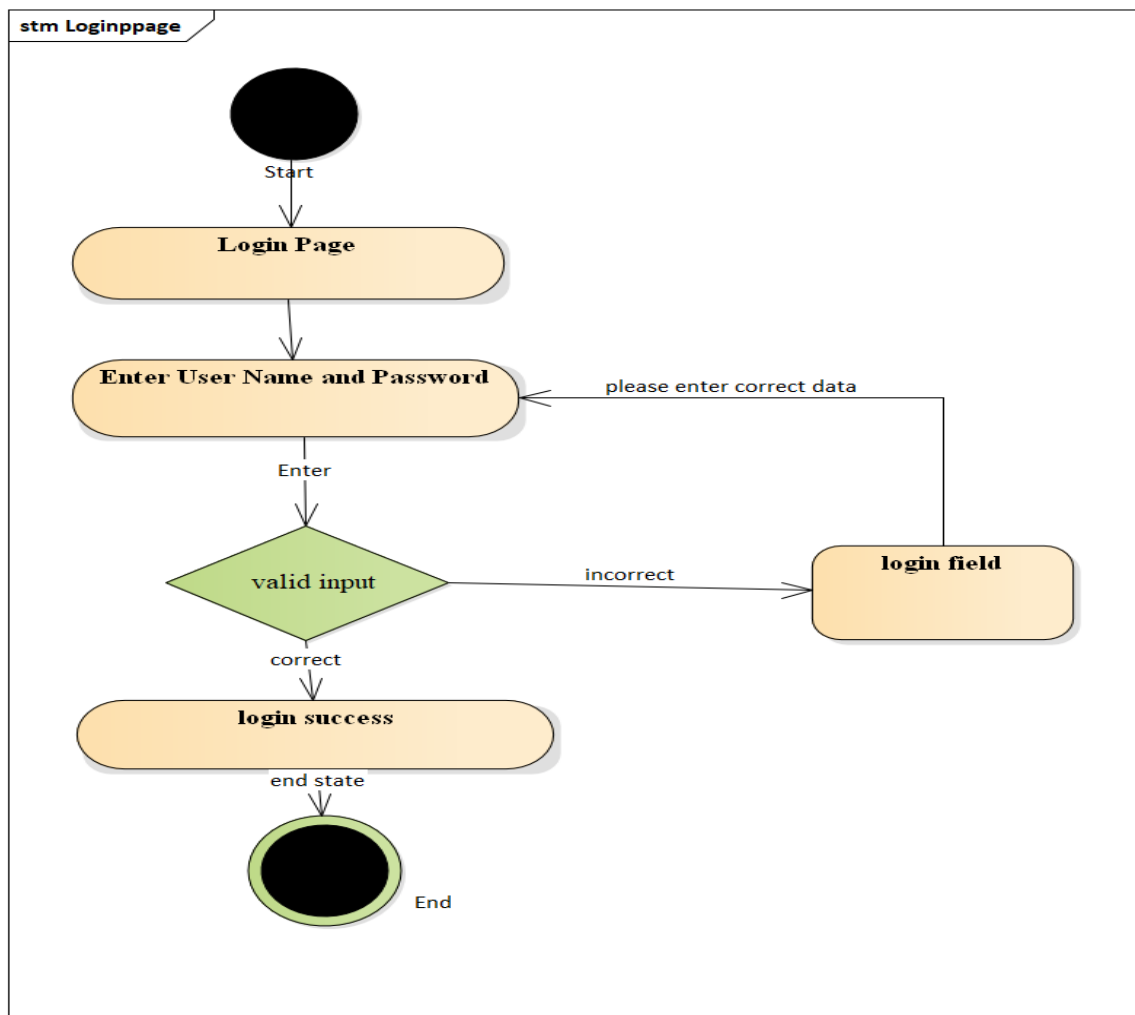


Figure4. 13 Login State Chart Diagram

Registration

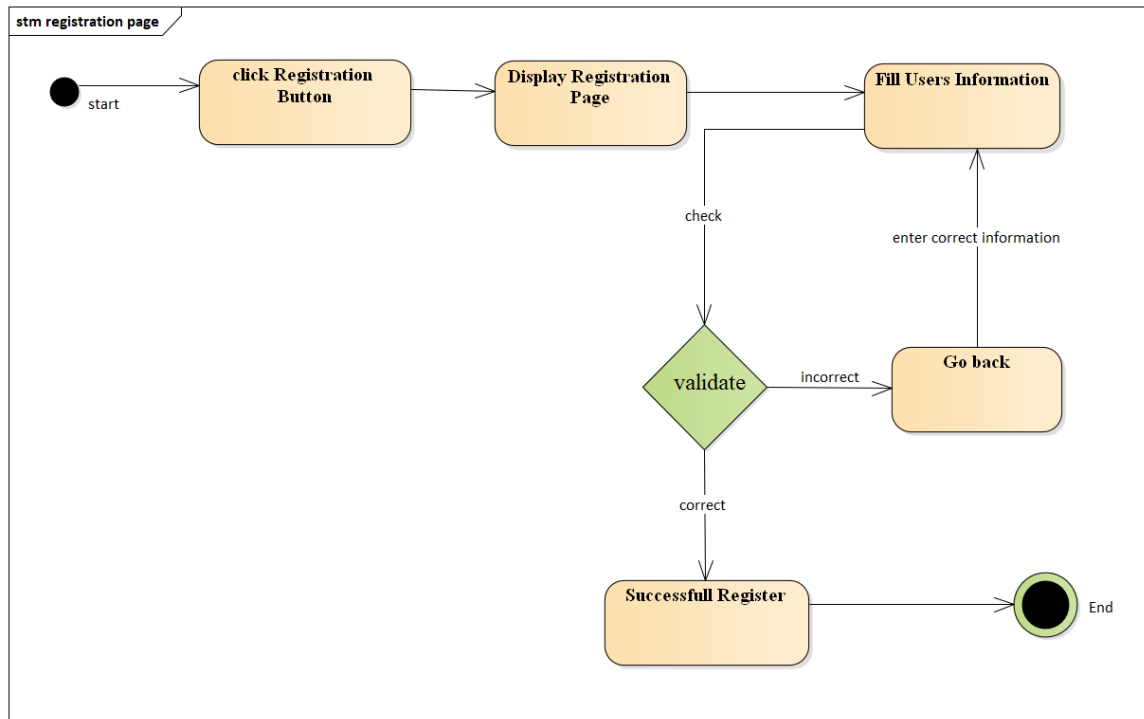


Figure4. 14 Registration State Diagram

View Package

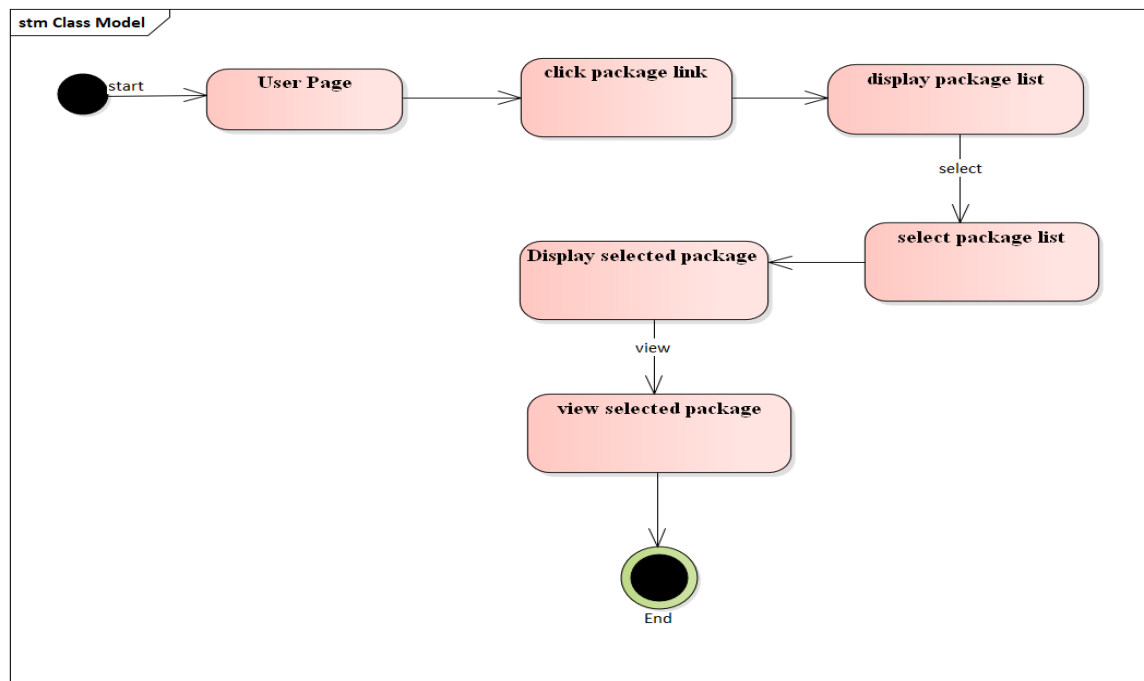


Figure4. 15 View Package State Diagram

Search

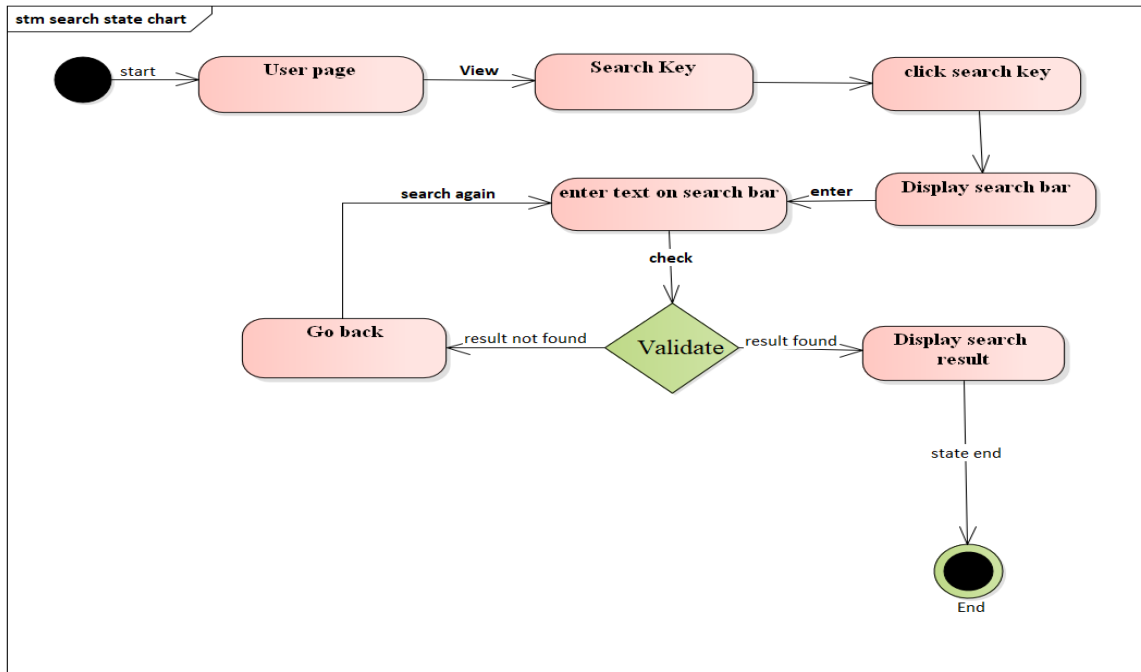


Figure4. 16 Search State Chart Diagram

Add Tour package

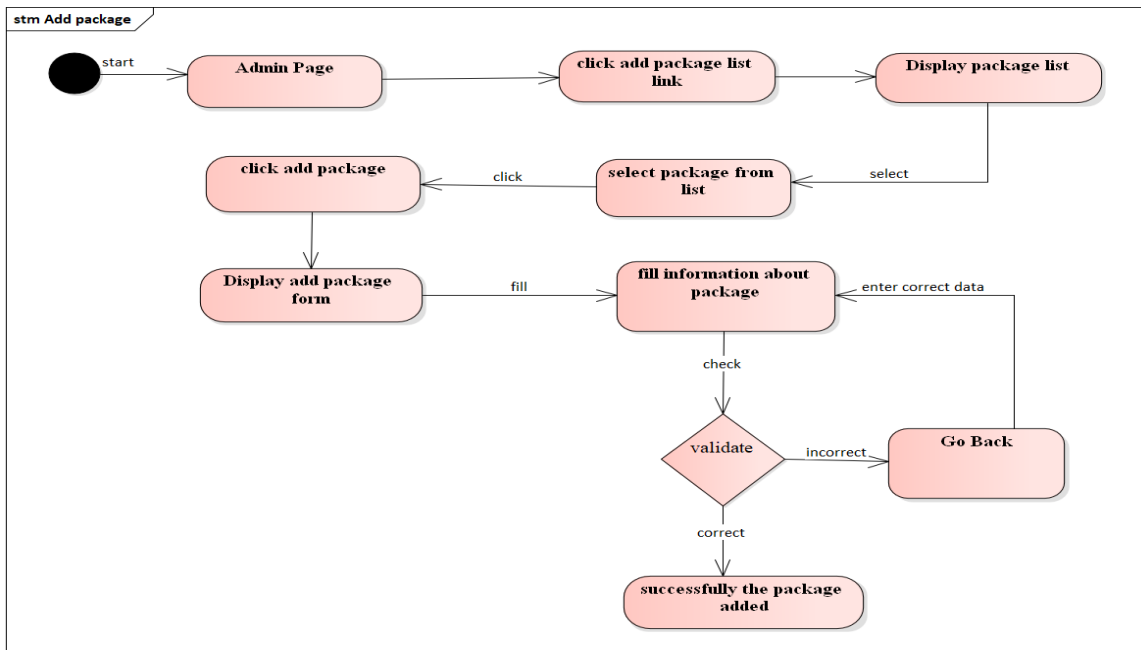


Figure4. 17 Add Tour Package State Diagram

Delete Package

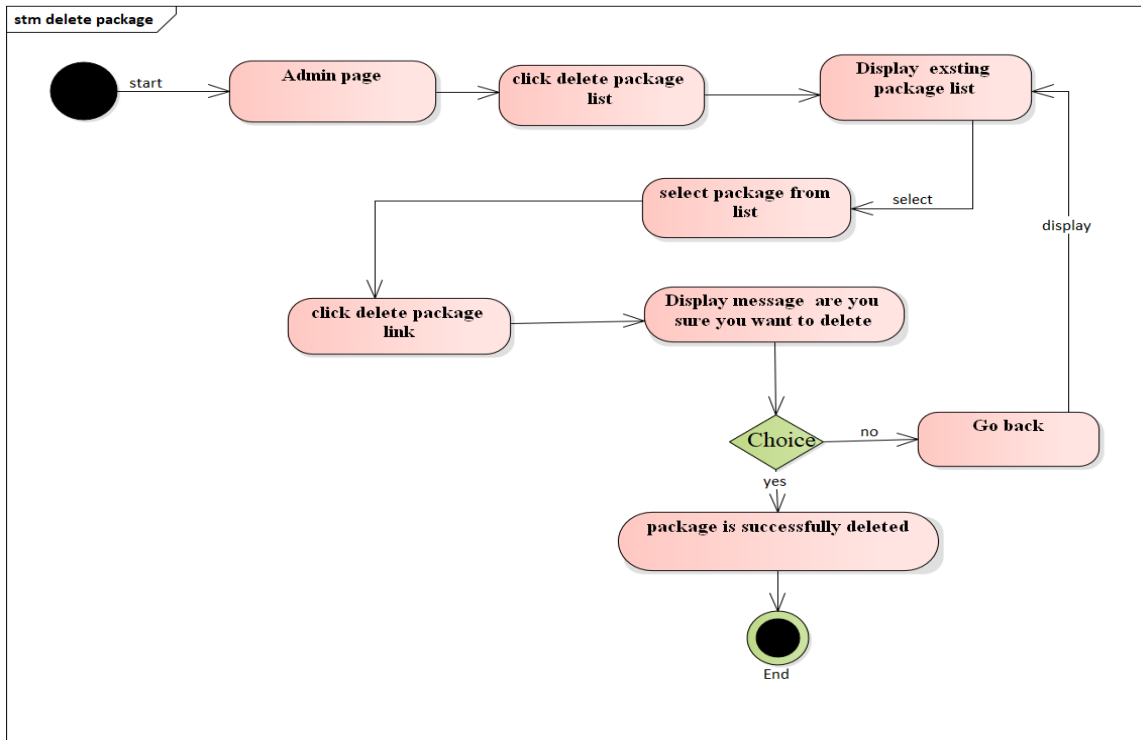


Figure4. 18 Delete Package State Diagram

Post Advertisement

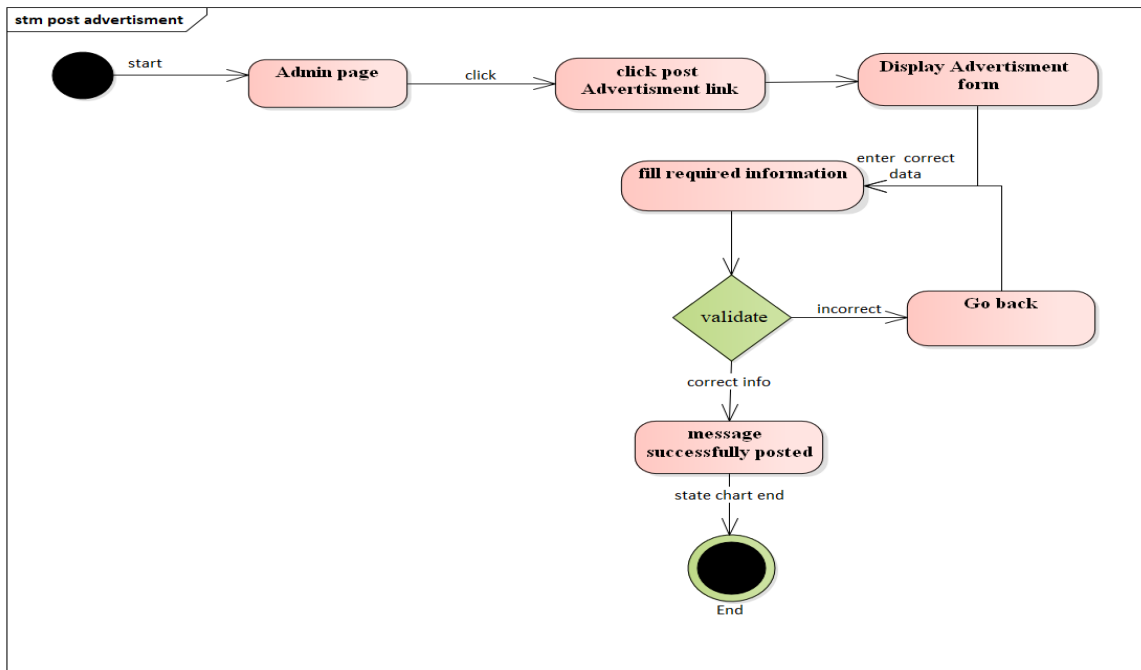


Figure4. 19 Advertisement State Diagram

CHAPTER FIVE

5. SYSTEM DESIGN

System design is the transformation of the analysis model into a system design model. Design is the process of describing, organizing, and structuring system components at architectural design level and detailed design level. Traveler guide application system design the overview of the proposed solution for existing system problem through the proposed system. Our proposed system is divided into different categories to the complexity of the system when we providing a solution for the specific problem.

5.1. Design Goals

Design goals are targets for design work. These are typically agreed upon by stakeholder as the Criteria for comparing design alternatives and evaluating design outcomes. The objectives of design are to model the system with high quality. The design goals are derived from nonfunctional requirements that mean the non-functional requirement is the description of the feature characteristics and attribute of the system as well as any constraints that may limit the boundary of the proposed system. The Design Goals specify the qualities of the system that should be achieved and addressed during the design of the system.

5.1.1. Performance

Users are becoming more unable to wait for a page to load, so the system can have as good a performance as possible. This can be achieved by quickly loadable interface components and a quick reaction time while browsing, modifying, deleting, adding, and accessing information from the database easily and quickly.

5.1.2. Dependability

The system should achieve the following dependability characteristics to resist crashes and be available and reliable.

- **Availability:** Our system uses firebase as the back end technology so it supports 200,000 simultaneous connections per database and response time should be very small. As the number of user increase the availability of the system decrease but in our case as the number of user increase it does not affect availability of the system. Our system also built using OOSAD as the basic that is the functionality of

travel guide application are independent to each other or the failurity of one function could not affect the functionality of another so our system increase Mean Time Between Failures.

- **Security:** A user must log in to the system with a user name and password. The system allow login to only authorized users. i.e. users that have previously created an account through user name (E-Mail) and password. The system has three groups of users: The Admin,Travel agency and Traveller .The Admin user has full privilege to perform on the system to add tour package ,post advertisement,manage hotel info. Whereas users can only perform limited operations based on the privilege given by the system. The data that is stored in the firebase is private. The proposed system security developed with hash function algorithm.
- **Reliability:** - the information provided by the system is as reliable as it is presented on the web page interface, and this is maintained by the persistent database.

5.1.3. Maintenance

Maintenance: - In a time of failure or need modification the system needs to be maintained. To be maintainable the system should meet the following maintenance criteria.Our system built using object oriented programing so each and every functionality of the system can prforme its oun function can't affect with each other. If the one module of the system failed to prform its function it is easy to mentain the individulal module rather than the whole system so Mean time to repair (MTTR) or average time it takes to repair a system is less.

5.1.4. End-user

The system's end users are classified into two main categories, these are:

- **Profession user (Administrator):** He/she will take technical as well as supportive pieces of training to help the layman users when problems arise.
- **Layman (Customer, Product Manager):** A user who has some basic computer knowledge, they can operate the system with simple training and with the help of user documentation.

5.1.5. Priorities of Design Goal

- **Functionality vs Usability**

The system favors usability compared to functionality yet it doesn't mean that the system loses its proper functionality. Rather the developing team will endeavor to

maintain the basic requirements of the system along with high usability. The system will avoid lavish functionalities, which are not a must for the system's existence, for the sake of encouraging basic system utilization

- **Performance vs portability**

The system selects portability compares to performance but not say our system does perform his task with low performance that annoys every user rather the system performs its operation without depending the specific type of operating system as a result of this users can install the system any place where ever it needed without bothering a kind of plate form, they going to use.

- **Security vs Usability**

Possible have lots of reason for selecting security instead of usability but again our system considers usability besides security. Relating security issues our system will drown line for each user which functionality of the system need to access by

Conclusion

Usability

- Our system gives more priority for usability because our system is a travel guide application it guides the traveler to visit different tourism site, historical place in Ethiopia.
- The system user interface is easy for traveler to use and also the system is clear to understand and use.
- Our system also used by the expertise and non-expertise of the traveler.

Portability

- Travel guide application can be used by different user which are located in different Environments.
- Our system will priorities portability than performance. Our system will be download from google play store in different android and iOS device in different Environment.

Usability

- Our system priorities usability than security because our system is a travel guide application it guides and give full and correct information related to tourism so the security feature is not more important, but have a security feature in the system.

5.2. Proposed System Architecture

Our proposed system is consisting of 3-tier architecture namely presentation layer, business logic layer, and data layer. The presentation layer is client layer and topmost layer of the application. This is the layer we see when we use this system. It is the interface to our system which takes information from the user. The main functionality of this layer is to communicate with the application layer. This layer passes the information which given in terms of keyboard action and mouse click to the application layer. Example when user want to login into the system first see two text box and login button to enter user name and password and click on login. Business logic layer it is application layer which interacts with data layer and sends information retrieved from database toward to presentation layer.it act as the mediator between presentation layer and data layer. From above example once, the user clicks on login button application layer interact with the database and send information towards to presentation layer. The third one is data layer which used to store data entered by the user. In general client of our system use browser to access the system using the internet.in this case when the user enters input and takes certain action application server process client request to interacting with the database server. Figure 5.1 below is proposed system architecture.

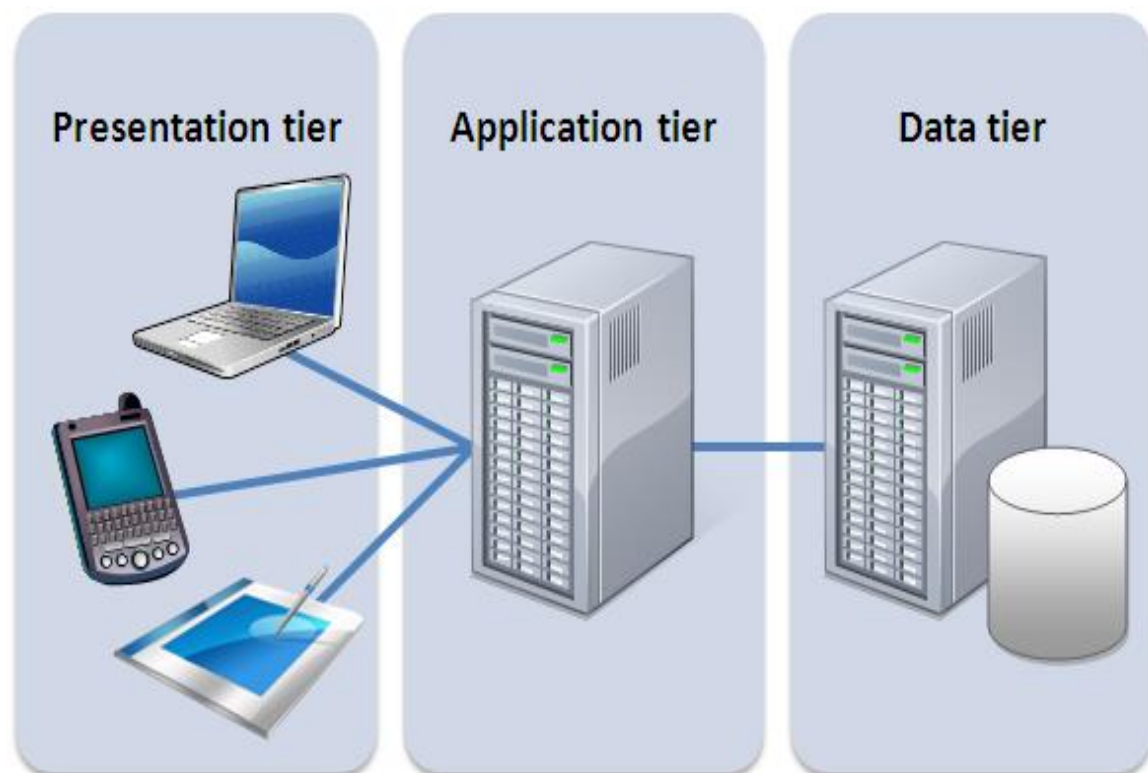


Figure 5. 1 Proposed System Architecture For Traveler Guide Application

5.2.1. Subsystem Decomposition and Description

Subsystem decomposition is the process of dividing the system into manageable subsystems from the analysis model of the proposed system. The goal of the system decomposition is to reduce the complexity of the design model and to distribute the class of the system into large-scale and cohesive components. Components are generally units of computation or data stores in the system. A component has a name, which is generally chosen to represent the role of a component or the function it performs. The different components of a system are likely to interact while the system is in operation to provide the service expected of the system. in our system the following subcomponent is available

Manage account subsystem: -in this subsystem managing information regard to account and performs.

- Create account.
- Delete account
- Update account

Database connection subsystem: this subsystem used to establish a connection between business class and database management system.

Travel registration subsystem: -this subsystem describes travel registration information

- Register traveler

Create profile subsystem: -this subsystem describes creating a profile for traveler's

- Create profile
- Update profile
- Delete profile

Package management subsystem: -this subsystem describes the management of information related to the package.

- Add package
- Update package
- Delete package
- View package
- Search package

Post advertisement subsystem: -this subsystem describe information related to the advertisement

- Post new advertisement
- Update advertisement
- Delete advertisement
- View advertisement

Report management subsystem: -this sub-system describes information related to the report.

- Generate report
- Delete report

Hotel management sub-system: - this subsystem describes information related to the hotel.

- Add hotel
- Update hotel
- Delete hotel
- View hotel information

Transport management subsystem: - responsible for describes information related to transport.

- Add transportation
- Update transportation
- Delete transportation

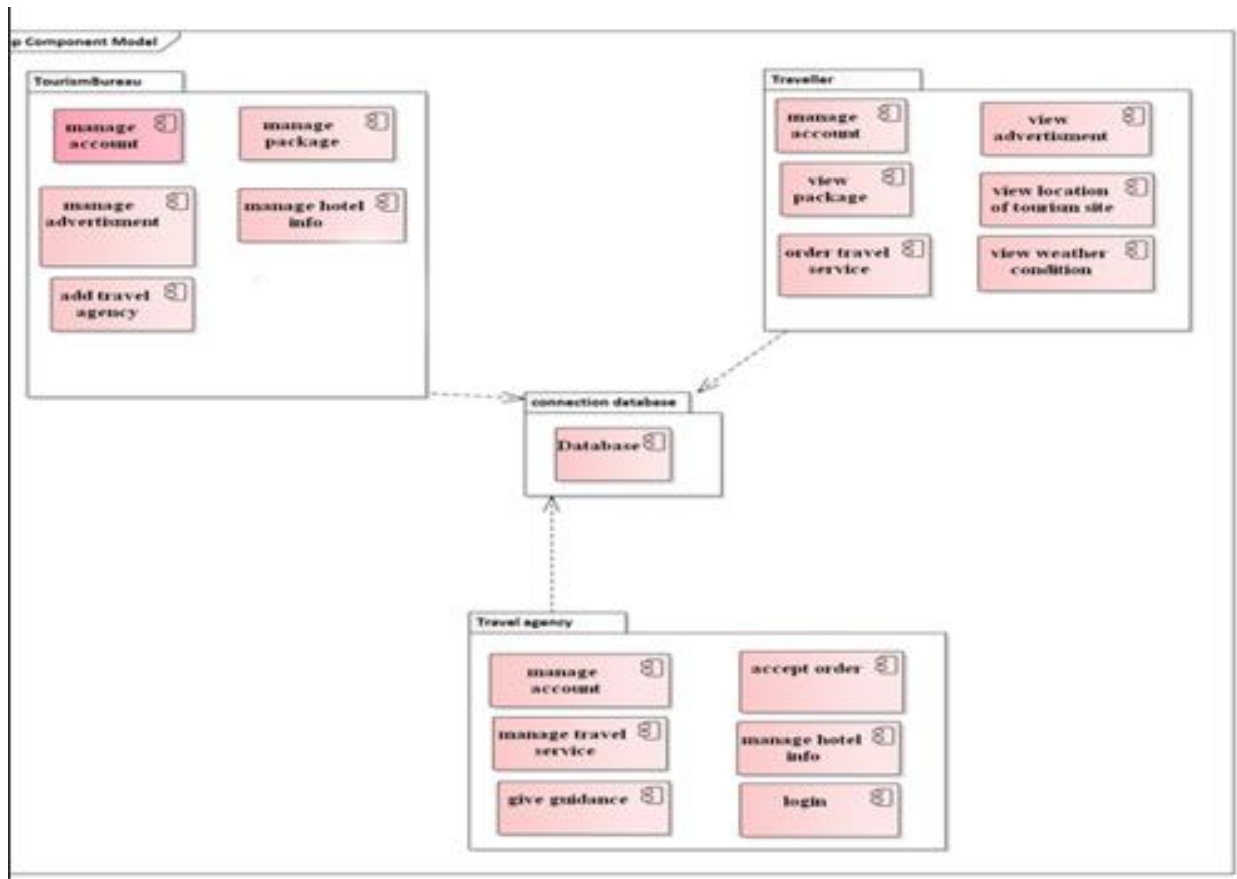


Figure 5. 2 Subsystem Decompositions

5.2.2. Hardware/Software Mapping

The deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware Used to connect the disparate machines. You want to create a deployment diagram for applications that are deployed to several machines. It also shows how the software and the hardware components work together. Deployment diagram used to show the hardware of the system, the software that is installed in the hardware, and also the middleware that is used to connect the disparate machines to one and other.it also shows how the software and hardware components of the system work together. Figure 5.4 below is shows the deployment diagram of our system.

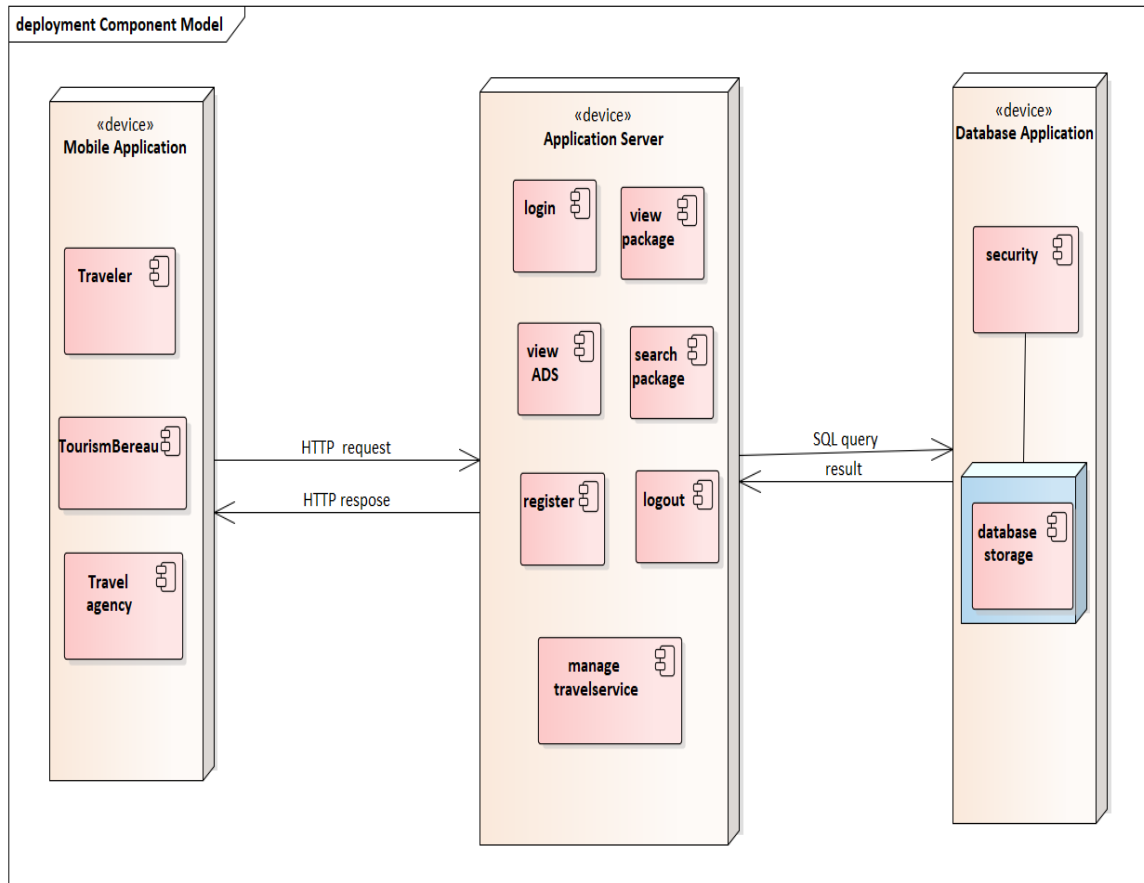


Figure 5. 3 System Deployment Diagram

5.2.3. Detailed Class Diagram

Detail class diagram is a class diagram that includes attributes, methods, attribute data types, visibility of attributes and methods, inheritance, association, aggregation, composition, dependency, and municipality (cardinality and optimality). The following figure uses the UML class diagram to specify attributes and operations with their visibility information.

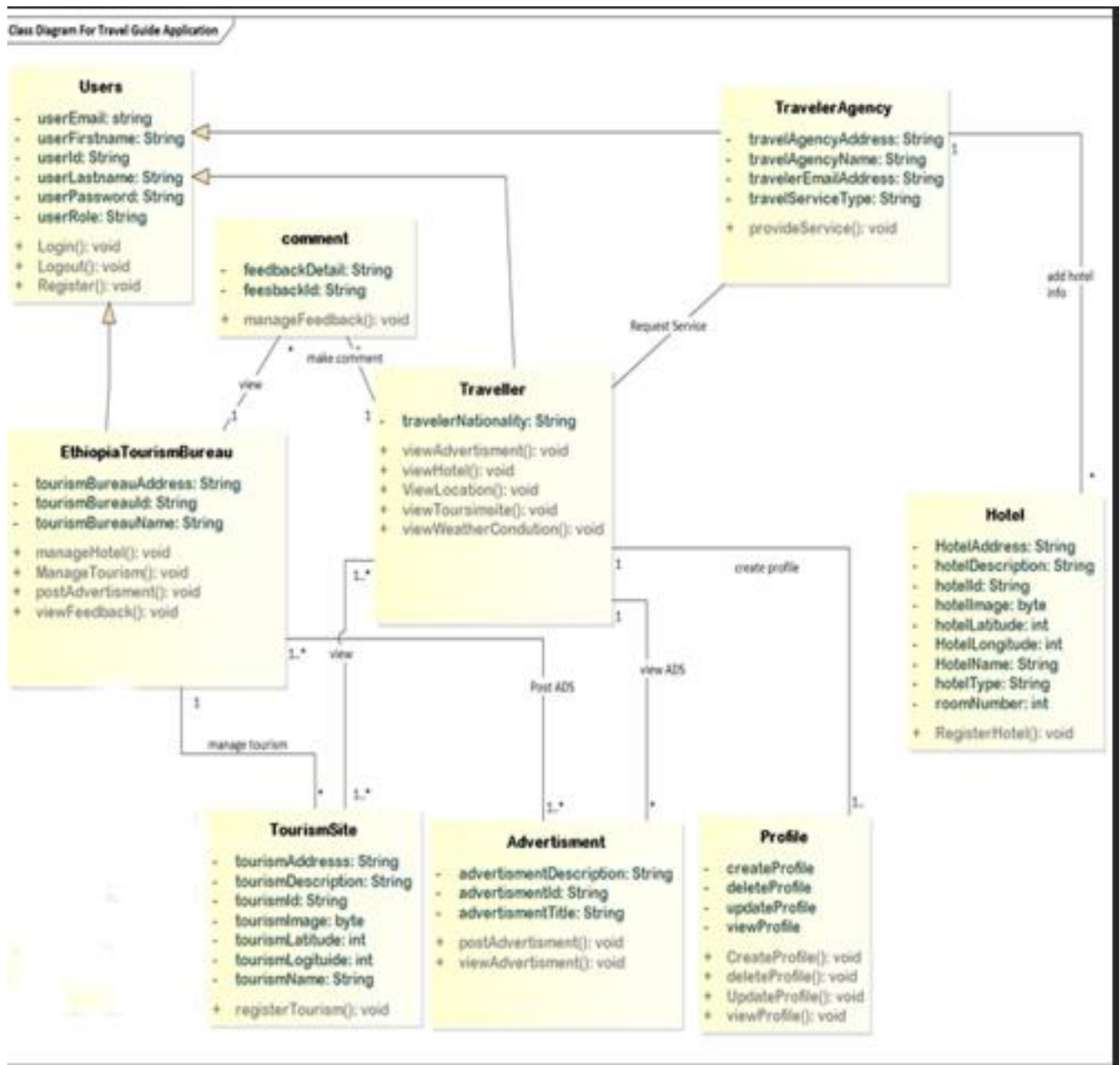


Figure 5.4. Detailed Class Diagram

5.2.4. Persistent Data Management

Persistence modeling is used to communicate the design of the database, usually the database to both the users and the developers. It is also used to describe the persistence data aspect of the system. The following tables indicate the persistence data management of the system

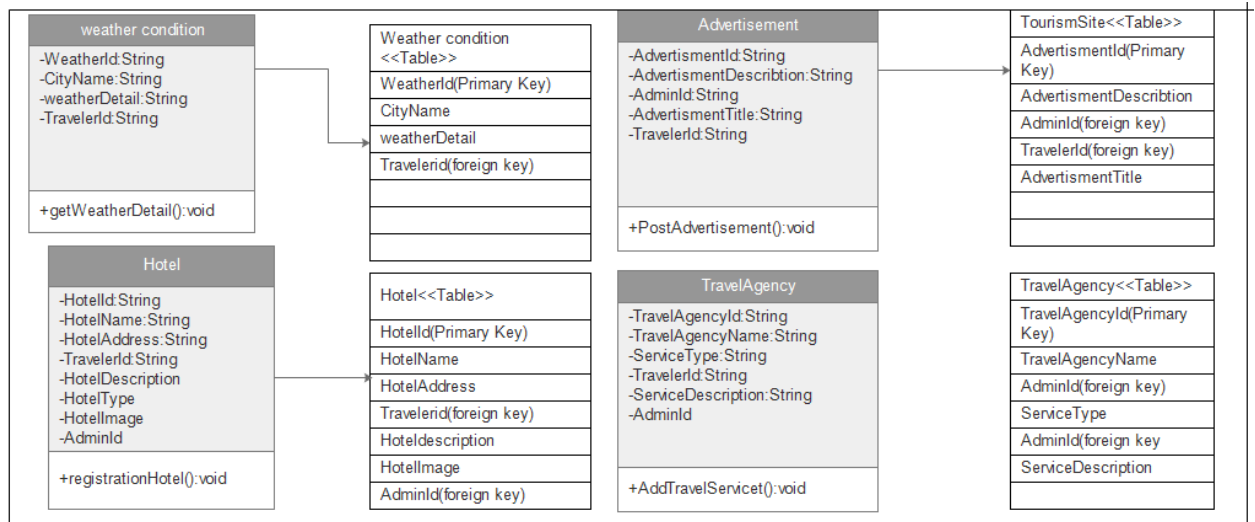
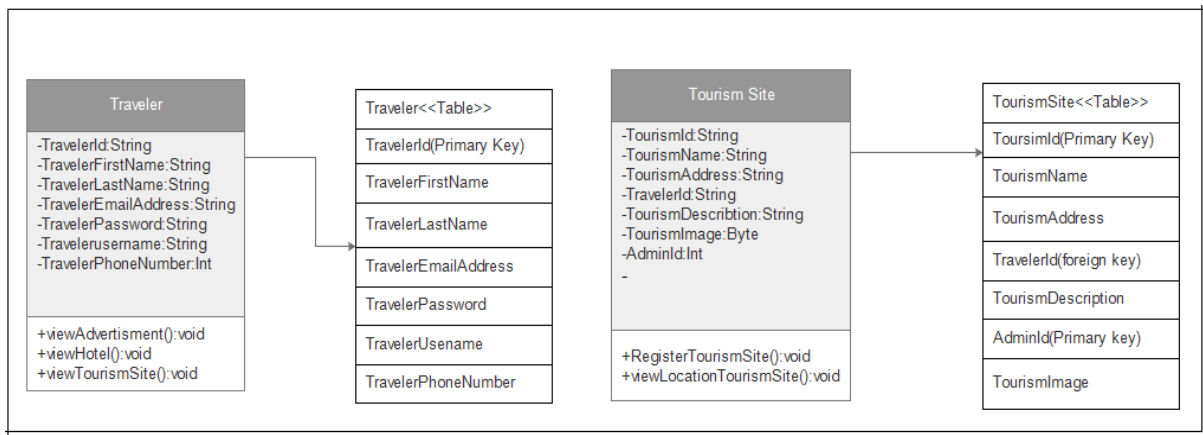
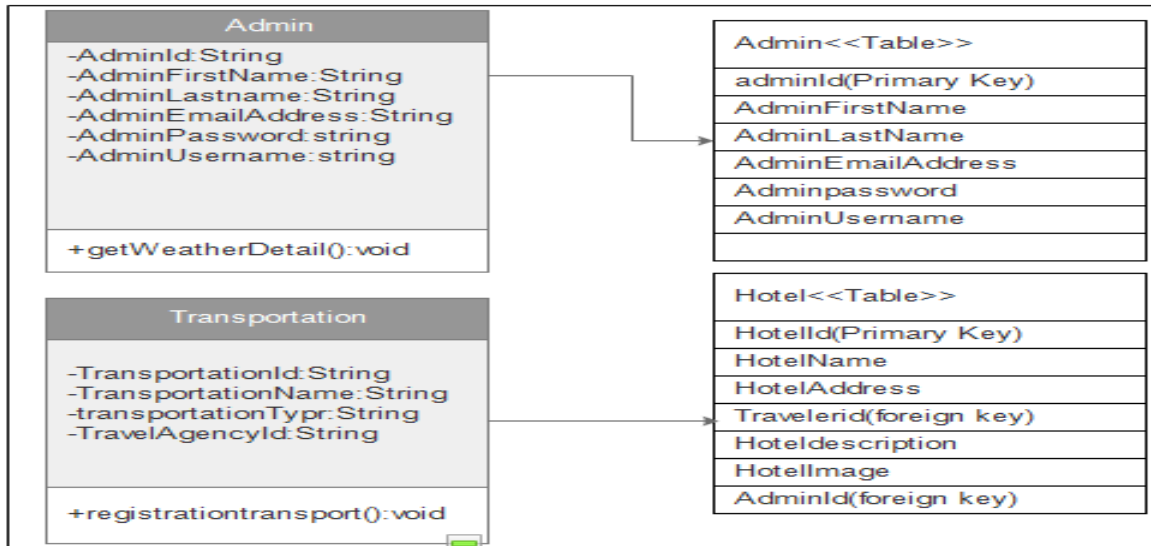


Figure 5. 5 Persistent Diagram

5.2.5. Access Control and Security

In our system, different actors have access to different information and data. Access control and security specify what the user can access or what cannot perform by some users. This access control is verified by username and password. System admin represents an authenticated user. The proposed system follows multi-user system. In multi-user system, different actors have access to different functions and data. Then it must be having: -

- Confidentiality: Only an authorized person can see the information. Private data is kept private; personal privacy is respected.
- Integrity: There are limits on who can change the data in this system.
- Availability: The system is available at all times to authorized users

Table 5. 1 Access Control For User

List of class	List of actors	Operation
Traveler	Traveler	<ul style="list-style-type: none"> • Create profile • View current location • View the location of tourism site, hotel, historical place and travel agency • View weather condition of city • Favorite tourism site • Search data
Admin	Admin	<ul style="list-style-type: none"> • Manage hotel information • View comment • Manage event • Manage tour package • Manage advertisement • Add Travel agency
Travel agency	Travel agency	<ul style="list-style-type: none"> • Manage transportation • Manage hotel • Add other travel service

5.3. Packages

The package is an organized and functionality-based set of related interfaces and classes. Packages organize classes that belong to the same category or provide similar functionality. In our system we category into database management package, report management package, Tour package management package, account management package, post

advertisement management package, Hotel management package, transport management package.

- **Database management package**

The database subsystem will be implemented by the firebase database management system which is used to store the persistent data.

- **Report management package**

It is a subsystem that helps an administrator to view the overall report of the system how many Travelers are visiting our country

- **Tour package management**

It is a sub-system that helps the user to see different tourism sites, hotels, travel agencies, and also the weather condition of Ethiopia.

- **Account management package**

It is a subsystem that helps an administrator to manage user account

- **Hotel management package**

It is a subsystem that helps the administrator can manage the hotel information and also the traveler can view the hotel information

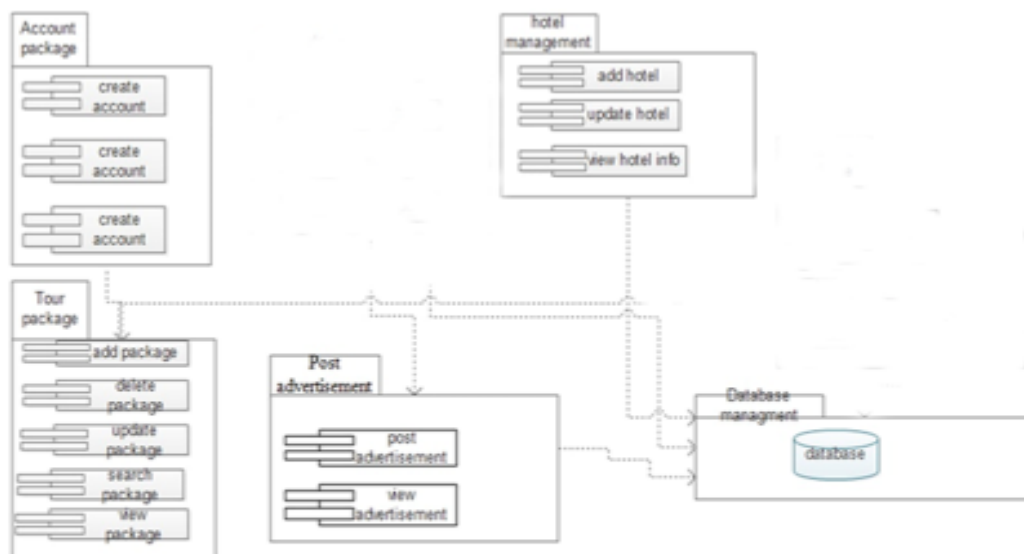


Figure 5. 6 Package Diagram

5.4. Algorithm Design

Algorithms are designed to show the flow of programs in the system. They are semantic driven rather than syntax driven. That means, the rule of syntax is not respected as other programming language but it has a complete meaning as that of syntax-based programming language. Also, Algorithms show the flow and steps of logic in each function. This design part is important in the coding part of the implementation. Some of the algorithms are listed below.

Create firstname,lastname,email,password and username variable

If firstname,lastname ,email,username and password matches accepted pattern.

 Add them to registration list

Else the system shows error message .

Create email,password variable

If email and password mathes accepted pattern

 Add them to login list

Else the system shows error message.

Composite pseudocode (MVC)

abstract class Component

 protected name

 public Component(name)

 public abstract Add (Component c)

 public abstract Remove (Component c)

 public abstract Display ()

Class Leaf: Component

 public Leaf(name)

 public override Display ()

class Composite: Component

 private list/array ChildComponents

```

    public Composite(name)

    public override Add(Component c)
        ChildComponents.Add(c)

    public override Remove(Component c)
        ChildComponents.Remove(c)

    public override Display()
        print all ChildComponents

class Main

    Composite menu = new Composite("TimeWaster Menu")
    menu.Add(new Leaf("Home"))

    Composite submenu = new Composite("Subscriptions")
    submenu.Add(new Leaf("Art tutorials"))
    submenu.Add(new Leaf("Music videos"))
    submenu.Add(new Leaf("Dumb cat vids"))

    menu.Add(submenu)

    menu.Add(new Leaf("Likes"))

    menu.Remove(new Leaf("Likes"))

    menu.Display()

```

5.5. User Interface Design

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design) so in our project Travel guide application system, we have designed user interfaces that increase the user experience.

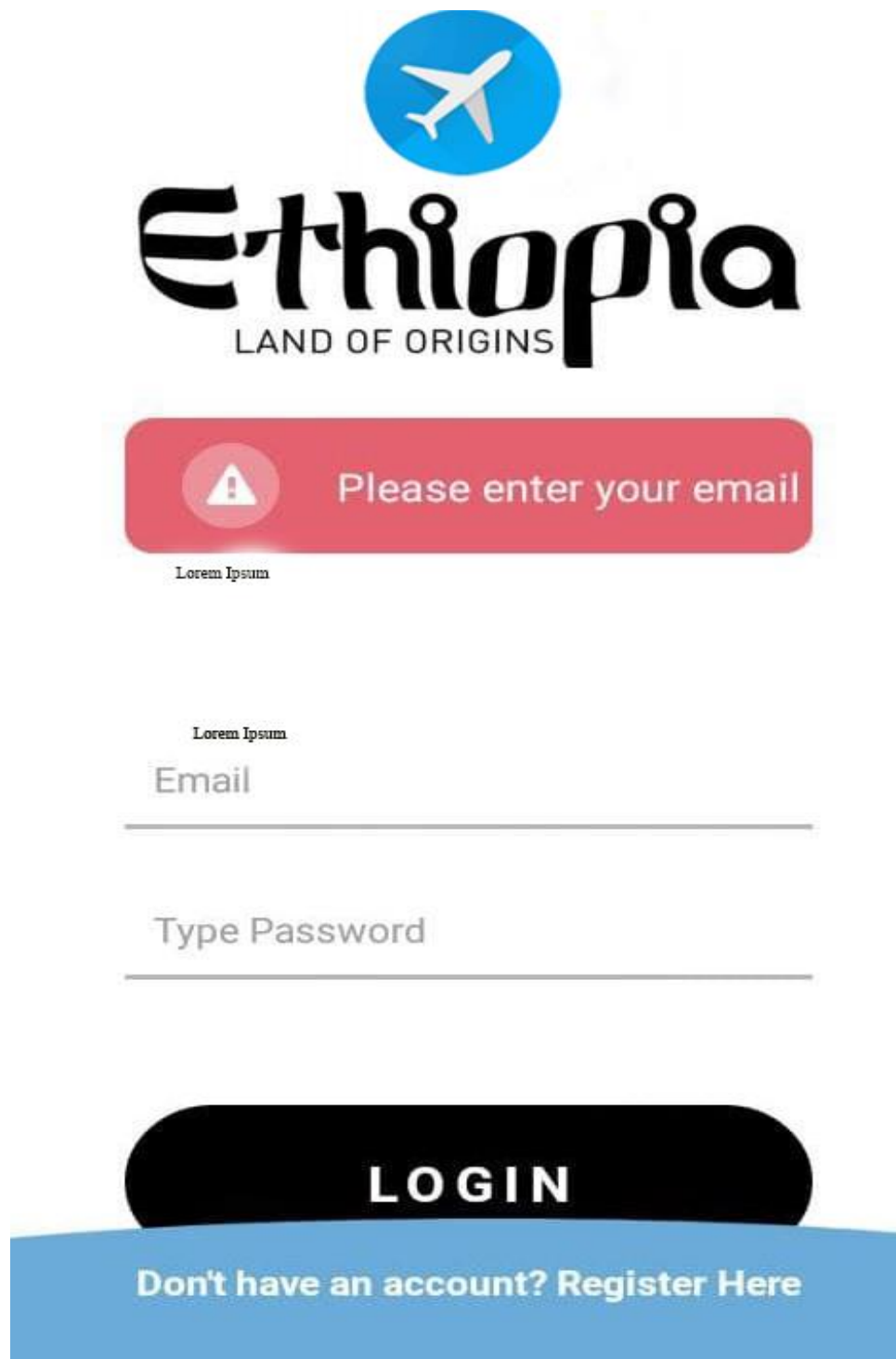


Figure 5. 7 User Interface Design For Login Page

This is the user interface design for our travel guide application the user must be full fill the user name and password before clicking the login button. After the user fill the forms by entering user name and password then the user clicks login button to view the full system function and its service. If the user enters wrong user name and password the system shows error message like as you see in the above figure.

CREATE NEW ACCOUNT

Firstname

Lastname

Email

Type Password

Re-Type Password

REGISTER

[Already have an account? Login Here](#)

Figure 5. 8 The User Interface Design For Registration

This is the user interface for design for registration page in this user interface design that the user must be first registration before interacting with the application in order to register themselves in this application the user must enter firstname ,lastname ,email and password.If the user enter short password and incorrect email address the system show error message please enter password>8 and correct email address.

Edit User



Update Profile

Figure 5. 9 The User Interface Design For Create Profile

This is the user interface design for the user create profile. In this user interface the user may be select the profile picture and enter the firstname lastname and email. If the user enters incorrect email address the system show you have entered incorrect email id, please enter the correct one.

CHAPTER SIX 6.

6. IMPLEMENTATION AND TESTING

6.1 introduction

This chapter basically highlights the issues discuss about the implementation phase. In this part of the project we implemented all the specific objectives, fundamental and non-functional requirement we pointed out the early on the early stage of the project. The coding convention has made it is possible for abstract objectives to become reality. After we implement the code the code was tested the system until most of the error have been detected and corrected.

Since implementation referee to the coding of all the document gathered starting from requirement analysis to Design phase. Business logic, information gathered are designed into the code so that the system will be implemented for the user to be tested.

The chapter also describe about the final testing system, hardware and software acquisitions, and installation process. Without leaving out about the hardware and software requirement, user manual preparation, training, installation processes, startup strategy, algorithm, and maintenance of the project. The design phase above is the key input to the implementation phase.

6.2. Implementation of the Database

To implement our project, we use Firebase Database for developing the database, we started from creating all the tables with which we pointed out on the design phase and assign primary key, foreign key. Below are sample of table that are taken from the database.

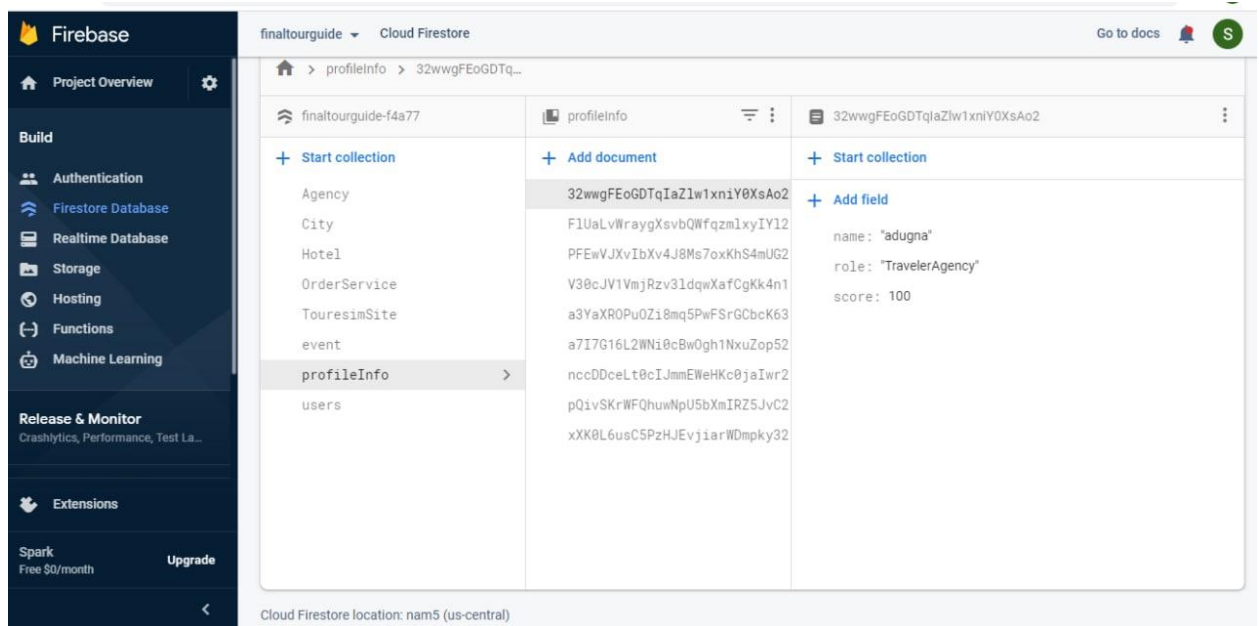


Figure 6. 1 Show Firebase Database With All Table

In our system there are above 8 database table can be created form Travel Agency, City, Hotel, Order Service, Tourism Site, event, profile Info and User.

You should choose appropriate database management system with justified reason and should perform the following activity. Tables which were identified and shown as persistent model in the design document should be created with their primary keys, foreign keys, check constraints and unique constraints.

Table 6. 1 Implementation Of The Database User Table

No	Name	Type	Primary key	Foreign key	check constraints	Unique constraints
	User Id		✓			
1	First Name	String			Only enter character	
2	Last Name	String			Only enter character	
	Phone number				>10 &&<15	
	Emergency contact name					
	Contact number	phone			Number digit>10 and less than 15	✓
3	Nationality	String				

Email	@
Password	>8character
Confirm password	>8character

Table 6. 2 Implementation Of The Database Tourism Site Table

No	Name	Data Type	Primary key	Foreign key	Check constraint	Unique constraint
1	TourismId	String	✓			
2	Name	String			ch>2 and ch<30	✓
	AdminId			✓		✓
3	Description	String			Not null	
	TravelerId			✓		
	Address				Not null	
4	Latitude					✓
5	Longitude					✓
	Category					✓

Table 6. 3 Implementation of the Database City Table

No	Name	Data Type	Primary key	Foreign key	Check constraint	Unique constraint
1	CityId	String	✓	✓		✓
2	City Name	String			ch>2 and ch<30	✓
3	City Address	String			Not Null	
4	City Description	String			Not Null	
5	Latitude	String				✓
6	Longitude	String				✓

Table 6. 4 Implementation Of The Database Order Service

No	Name	Data Type	Primary key	Foreign key	Check constraint	Unique
----	------	-----------	-------------	-------------	------------------	--------

No	Name	Data Type	Primary key	Foreign key	Check constraint	Unique constraint
1	OrderServiceId	String	✓			
2	First Name	String			ch>2 and ch<30	
3	Last Name	String			ch>2 and ch<30	
4	TravelerId	String		✓		
4	Phone Number	Int			Number digit>10 and less than 15	✓
5	Nationality	String			Not null	
	Source	String			Not null	
	Destination	String			Not null	
	Service Type	String			Not null	

Table 6. 5 Implementation of the Database Hotel Table

No	Name	Data Type	Primary key	Foreign key	Check constraint	Unique constraint
1	Name	String	✓	✓	ch>2 and ch<30	✓
2	Description	String				
3	Address	String				
4	Latitude	Double				✓
5	Longitude	Double				✓
	Cost	Int				
	Room	String				✓

Table 6. 6 Implementation Of The Database Special Event Table

No	Name	Type	Primary key	Foreign key	Check constraint	Unique constraint
	EventId	String	✓			
1	Name	String			ch>2 and ch<30	✓
	AdminId			✓		✓

2	Description	String		Not null	
3	Address	String		Not null	
4	Latitude	String			✓
	Longitude				✓
5	Celebration date	date		Not null	

Table 6. 7 Implementation Of The Database Travel Agency

No	Name	Type	Primary Key	Foreign Key	check constraints	Unique constraints
	AgencyId	String	✓			
1	Name	String			ch>2 and ch<30	✓
2	Description	String			Not null	
3	Address	String			Not null	
	AdminId	String		✓	Not null	
4	Phone Number	Int			Number digit>10 and less than 15	
5	Email	String			@	
	TravelerId	String		✓		
	Password	String			>8character	
	Service Type	String				

6.2.1. Whenever you found appropriate, implement the following: Indexes, Views, Triggers, Stored procedures.

6.2.1.1. Indexes

Cloud Firestore ensures query performance by requiring an index for every query. The indexes required for the most basic queries are automatically created for you.

6.2.1.2. Cloud Firestore triggers

With Cloud Functions, you can handle events in Cloud Firestore with no need to update client code. You can make Cloud Firestore changes via the Document Snapshot interface

- Cloud Firestore function triggers

The Cloud Functions for Firebase SDK exports a `functions.firestore` object that allows you to create handlers tied to specific Cloud Firestore events.

Table 6. 8 Cloud Firestore Function Triggering

Event Type	Trigger
<code>providers/cloud.firestore/eventTypes/document.create (default)</code>	Triggered when a document is written to for the first time.
<code>providers/cloud.firestore/eventTypes/document.update</code>	Triggered when a document already exists and has any value changed.
<code>providers/cloud.firestore/eventTypes/document.delete</code>	Triggered when a document with data is deleted
<code>providers/cloud.firestore/eventTypes/document.write</code>	Triggered when <code>onCreate</code> , <code>onUpdate</code> or <code>onDelete</code> is triggered.

Example.

- Sample code for Cloud Firestore function triggering

```
void create() async {
  // save to db
  CollectionReference ref = FirebaseFirestore.instance
    .collection('notes');
  // .doc(FirebaseAuth.instance.currentUser.uid)
  // .collection('notes');

  var data = {
    'title': title,
    'description': des,
    'created': DateTime.now(),
  };

  ref.create(data);
}
```

- Update data triggering

```

    TextButton( child: Text(getTranslated(context,"Delete")),
    onPressed: () {
    db.collection("Hotel").doc(snapshot.data.docs[index].id).delete().catchError((e)=>print(
    ));
    Navigator.of(context).pop();
    },
    )

```

- **Delete data triggering**

```

Map<String, dynamic> data = {
  "image": imageURL,
  "event_Name": nameInputController.text,
  "Description": descriptionInputController.text,
  "Address": addressInputController.text,
  "Location": GeoPoint(latu, longu),
  "selaviration_date":dateInputController.text,
};
snapshot.data.docs[index].reference.update(data);
clearText();|

```

6.2.2. Scheduled backup of database-tables with new data in month

We have a Firebase database with around 8 tables and approximately 1 megabytes of data, where the existing data is truncated in month and replaced by new data. The table-names and columns are unchanged month-to-month. But I need a backup of the data from Month 1 before the data from Month 2 is loaded into the database.

The original database is in this example called [Data_ Master] and my backup database is called [Data Backup]

```
final FirebaseFirestore ref=FirebaseFirestore. Instance;
```

example table name is **notes**

```
final Collection Reference db.=ref. collection('notes');
```

```
TRUNCATE TABLE [Data Backup]
```

```
ref.doc (
```

```
  db.doc [Data Backup]. collection(xxxxxx).doc ();
```

```
db. collection (Test table). whereEqualTo ("Data Backup", true). get ().
```

```
addOnSuccessListener {
```

```
xxxxxxx }
```

In my example, we will do this script 100 times in a stored procedure, which is a big task initially but when it's stored in a procedure, it's manageable. But we were thinking, whether there is a better/faster way, where we can get table copied/backup individually in Firebase database so they are accessible without further do. I'm looking for inputs, so all inputs are welcome.

6.2.3. Configure database level security

Firebase Security Rules allow you to control access to your stored data. The flexible rules syntax means you can create rules that match anything, from all writes to the entire database to operations on a specific document.

6.2.Default rules: Locked mode

When you create a database or storage instance in the Firebase console, you choose whether your Firebase Security Rules restrict access to your data (**Locked mode**) or allow anyone access (**Test mode**). In Cloud Firestore and Real-time Database, the default rules for **Locked mode** deny access to all users. In Cloud Storage, only authenticated users can access the storage buckets.

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

While we don't recommend leaving your data accessible to any user that's signed in, it might be useful to set access to any authenticated user while you're developing your app.

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

6.3 Implementation of the Class Diagram

```
AgencyRegistration({Key key}) : super(key: key);

//data base
final CollectionReference profileList =
FirebaseFirestore.instance.collection('profileInfo');
final CollectionReference agency =
FirebaseFirestore.instance.collection('Agency');

Future<void> createUserDataAgency(
    String name, String agency, int score, String uid) async {
    return await profileList
        .doc(uid)
        .set({'name': name, 'role': agency, 'score': score});
}

Future<void> createUserfullDataAgency( //this is method that used for d
    private String name,
    private String description,
    private String address,
    private int phoneNumber,
    private String transportationService,
    private String guiderService,
    private String hotelService,
    private String uid) async {
    return await agency.doc(uid).set({
        'name': name,
        'description': description,
        'address': address,
        'phoneNumber': phoneNumber,
        'transportation': transportationService,
        'guider': guiderService,
        'hotel': hotelService
    });
}
```

```

//Travel Agency home page dashboard
class TavelAgencyHomePage extends StatefulWidget{

  TavelAgencyHomePage({Key key, this.title, this.uid}) : super(key: key);
  final String title;
  final String uid;
  @override
  _HomePageState createState() => _HomePageState();
}

```

```

class _HomePageState extends State<TavelAgencyHomePage> {
  private var agencyName;
  private String name;
  private String lastName;
  private String returnDate;
  private String departureDate;
  private String destination;
  private String nationality;
  private String phoneNumber;
  bool hotel;
  final db = FirebaseFirestore.instance;
  @override
  void initState(){
    super.initState();
    agencyName=widget.title;
  }
}

```

```

// authentication for Login and registration page
class AuthenticationService {
  final FirebaseAuth _auth = FirebaseAuth.instance

```

```

//

```

```

////// sign with email and password

```

```

future loginUser(String email, String password) async {
  try {
    UserCredential result = await _auth.signInWithEmailAndPassword(
      email: email, password: password);
    return result.user;
  } catch (e) {
    print(e.toString());
  }
}

```

```

//User Homepage design

```

```

class UserHomePage extends StatefulWidget {
  UserHomePage({Key key, this.title, this.image, this.uid}) : super(key: key);

  //update the constructor to include the uid
  final String title;
  final String uid;
  final String image;

  @override
  _UserHomePage createState() => _UserHomePage();
}

class _UserHomePage extends State<UserHomePage> {
  private String name;
  private String image;

  void initState() {
    super.initState();
    name = widget.title;
    image = widget.image;
  }
}

```

6.4. Configuration of Application Security

In the configuration of application level security, we do message security which is service invoked at the interface between an application and queue manage which is connected. From the security aspect of any system, we should have to take in to account the following activities.

6.4.1. Implement all input validations properly

//Email validation

```
String emailValidator(String value) {
    Pattern pattern =
        r'^(([\<>()[]\.\.,;\s@"]+(\.[^<>()[]\.\.,;\s@"]+)*|(\.+\"))@((\[[0-9]{1,3}\. [0-9]
    RegExp regex = new RegExp(pattern);
    if (!regex.hasMatch(value)) {
        return getTranslated(context, 'Email format is invalid');
    } else {
        return null;
    }
}
```

Analyzing...

//Password validation

```
String pwdValidator(String value) {
    if (value.length < 8) {
        return getTranslated(context, 'Password must be longer than 8 characters');
    } else {
        return null;
    }
}
```

return null;

},

//Name Validation

```
TextFormField(
    controller: _lastNameController,
    validator: (value) {
        String pattern = r'^[A-Za-z]+$';
        RegExp regex = new RegExp(pattern);
        if (value.isEmpty) {
            return getTranslated(context, 'Last Name cannot be empty');
        } else if (value.length < 2 || value.length > 30) {
            return getTranslated(context, "Last Name is Between less than 30 and grater than 2 characters.")
        } else if (!regex.hasMatch(value)) {
            return getTranslated(context, 'Last Name is not valid!');
        } else
            return null;
    },
```

Indexing...

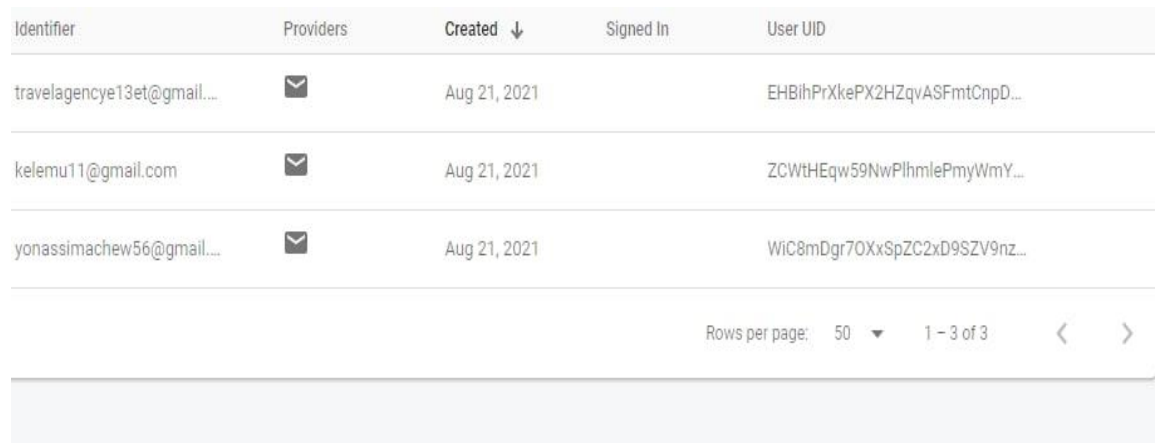
//Phone validation

```
TextFormField(
    keyboardType: TextInputType.number,
    controller: _emergencyContactPhone,
    validator: (value) {
        String pattern = r'^[0-9]+';
        RegExp regex = new RegExp(pattern);
        if (value.isEmpty) {
            return getTranslated(context, 'Phone Number cannot be empty');
        } else if (!regex.hasMatch(value)) {
            return getTranslated(context, 'Phone Number is not valid!');
        } else if (value.length < 9 || value.length > 20) {
            return getTranslated(context, "Value are not Between 9 and 20 characters !");
        } else
            return null;
    },
```

6.4.2. Implement encryption/deception

Our system uses Firebase Database as backend device. This back end device can authenticate the password by calling Hash function and pass the input from the field as a parameter, the input will be encrypted to cyber text and save to firebase database.

Below illustrates the example of Hash function algorithm.



Identifier	Providers	Created ↓	Signed In	User UID
travelagencye13et@gmail...	✉	Aug 21, 2021		EHBihPrXkePX2HZqvASFmtCnpD...
kelemu11@gmail.com	✉	Aug 21, 2021		ZCWtHEqw59NwPlhmlePmyWmY...
yonassimachew56@gmail...	✉	Aug 21, 2021		WIC8mDgr70XxSpZC2xD9SZV9nz...

Figure 6. 2 Implement encryption/deception

6.4.3.Roles must be defined clearly

In this project we have three roles namely Ethiopia Tourism Bureau, Traveler and Travel. And each role has their own privileged pages and the access is determining by their roles.

1. Traveler
2. Admin/Ethiopia Tourism Bureau
3. Travel Agency

6.5.4. User accounts must be assigned with necessary access privileges

```
if (_key.currentState.validate()) {
  try {
    _auth1.signInWithEmailAndPassword(
      email: _emailController.text,
      password: _passwordController.text)
      .then((currentUser) => FirebaseFirestore.instance
        .collection("profileInfo")
        .doc(currentUser.user.uid)
        .get())
      .then((DocumentSnapshot result) {
        if (result['role'] == "admin") {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(
              builder: (context) => Admin()));
        } else if (result['role'] == "users") {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(
              builder: (context) => UserHomePage(
                title: result["name"], image: result["score"],
                uid: currentUser.user.uid,
              ));
        } else if (result['role'] == "agency") {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(
              builder: (context) =>
                TavelAgencyHomePage(
```

6.7. Implementation of User Interface

For the actor that we are defined previously we create a user interface with different access privilege.

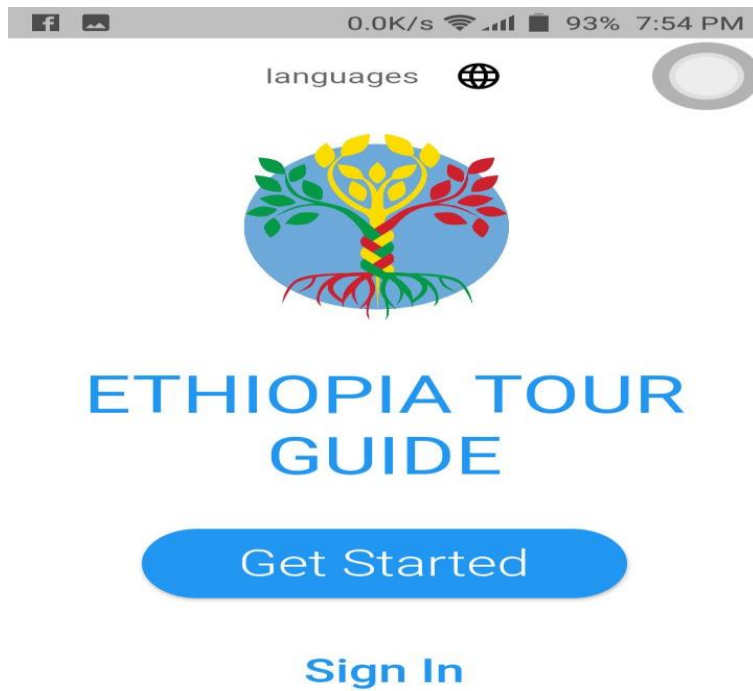


Figure 6. 3 Home page

This is the home page for Travel guide Android Application if any one use this system must first meet this home page. This is page contain button such us get started and Sign as a button if the user is new he/she must click Get Started button if not the user clicks Sign in button based on their role the user can use this system.

This page also contains one button which is called language if the user wants to change the language he/she can select the language.

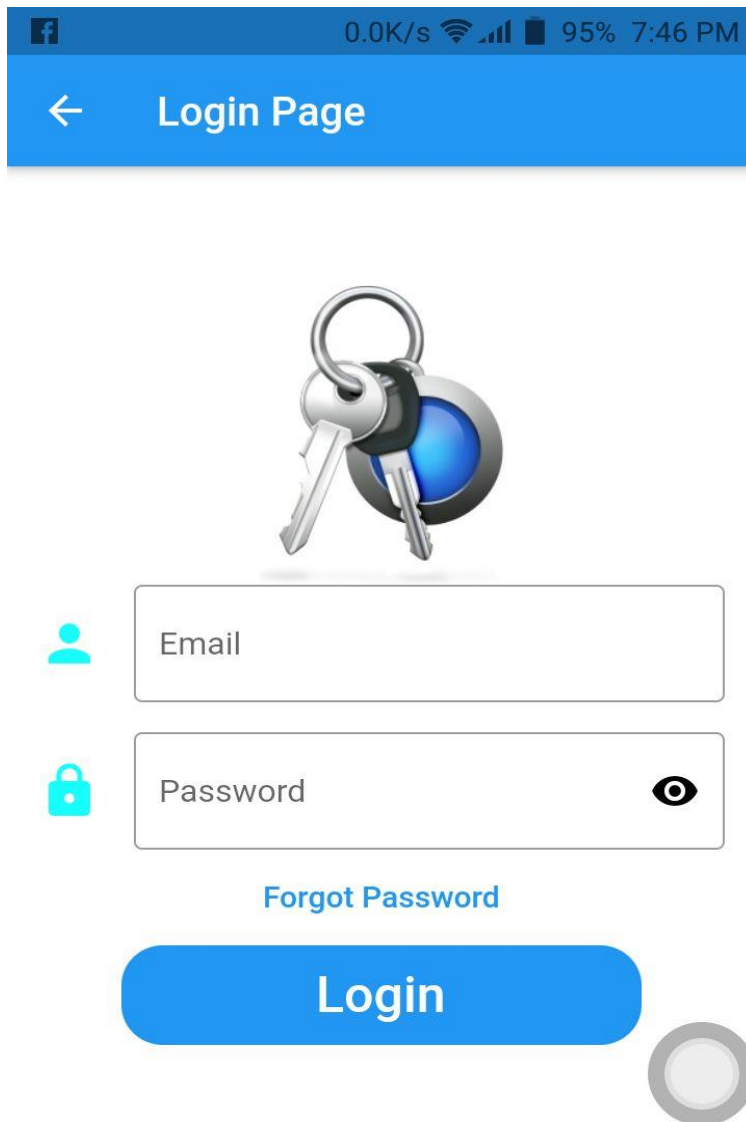


Figure 6. 4 user interface for Login Page for all user

This is the login interface that is all the actor can access it. By starting from this page all users (Actors) will be authenticated by the system and access their privileged pages.

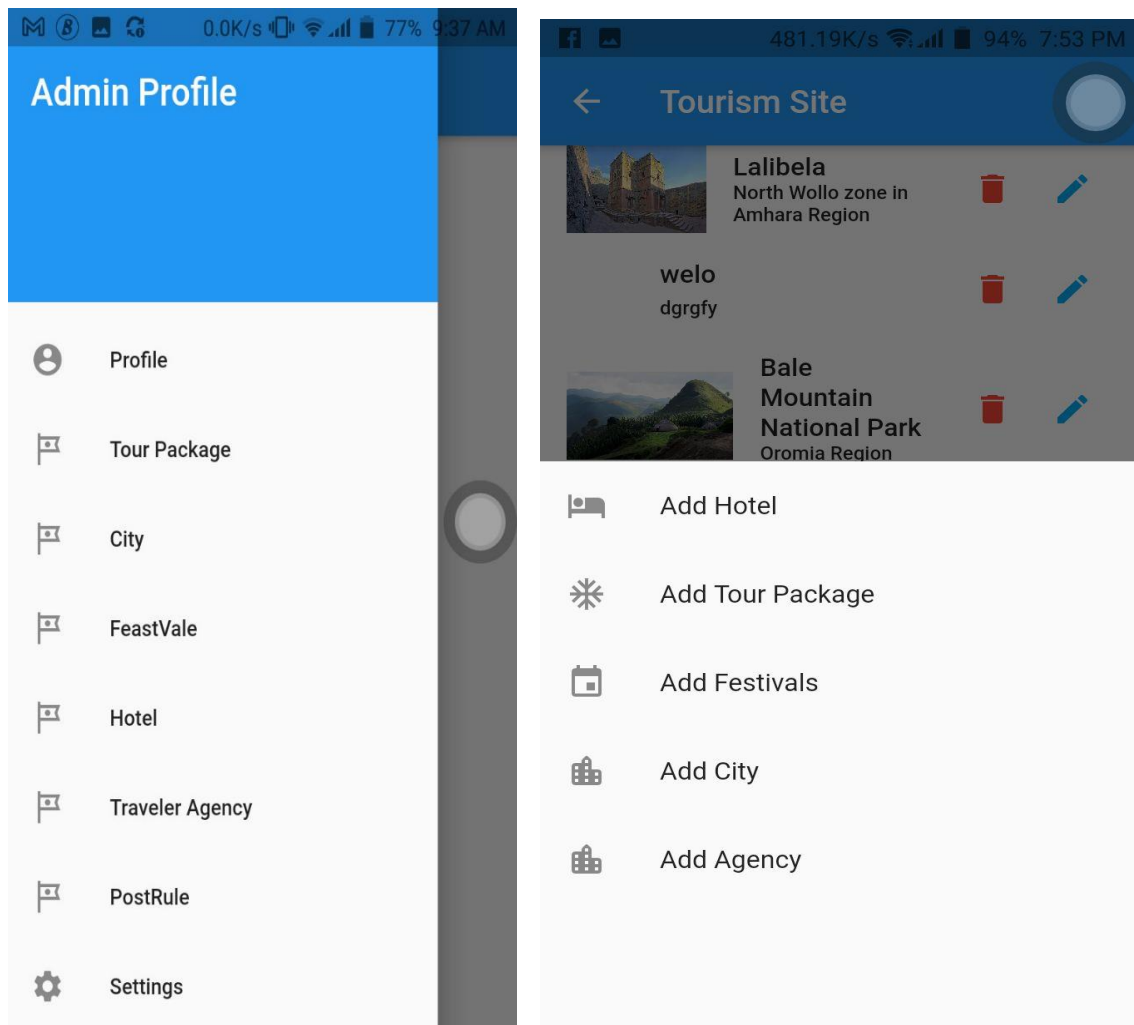


Figure 6. 5 User Interface For Tourism Site

- This is the interactive page for tourism site that the Admin can be login in and
- This is the Admin page of Travel Guide Android Application. In this page the Admin can manage Tour Package, City, Feast vale, hotel and Travel agency.
- The Admin also have full privilege to perform any crud function, i.e. create, read, update and delete all this type of action performed by Ethiopia Tourism bureau

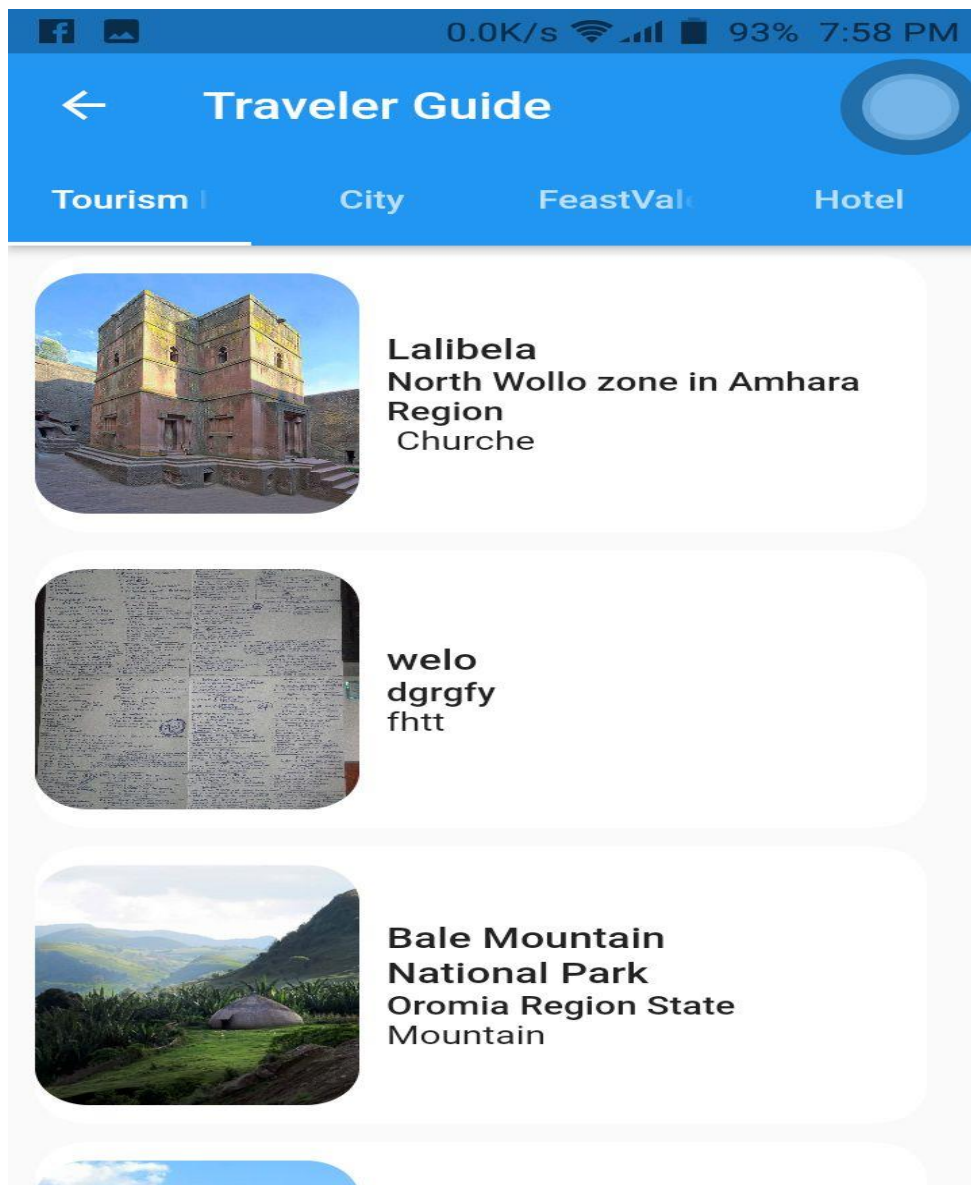


Figure 6. 6 User interface for Main page every one user can access

- This page is the main page of Travel Guide Application in android .in this page any one can accesses without creating username and password.

Agency Choose Agency

Name

Last Name

Phone number

Nationality

Source

Destination

Transportation

Figure 6. 7 User Interface For Order Service

- This is the order page for user if anyone have user name and password and also have much interested to visit different place in Ethiopia can access this page and send his/her request to different Travel Agency based on their interest.

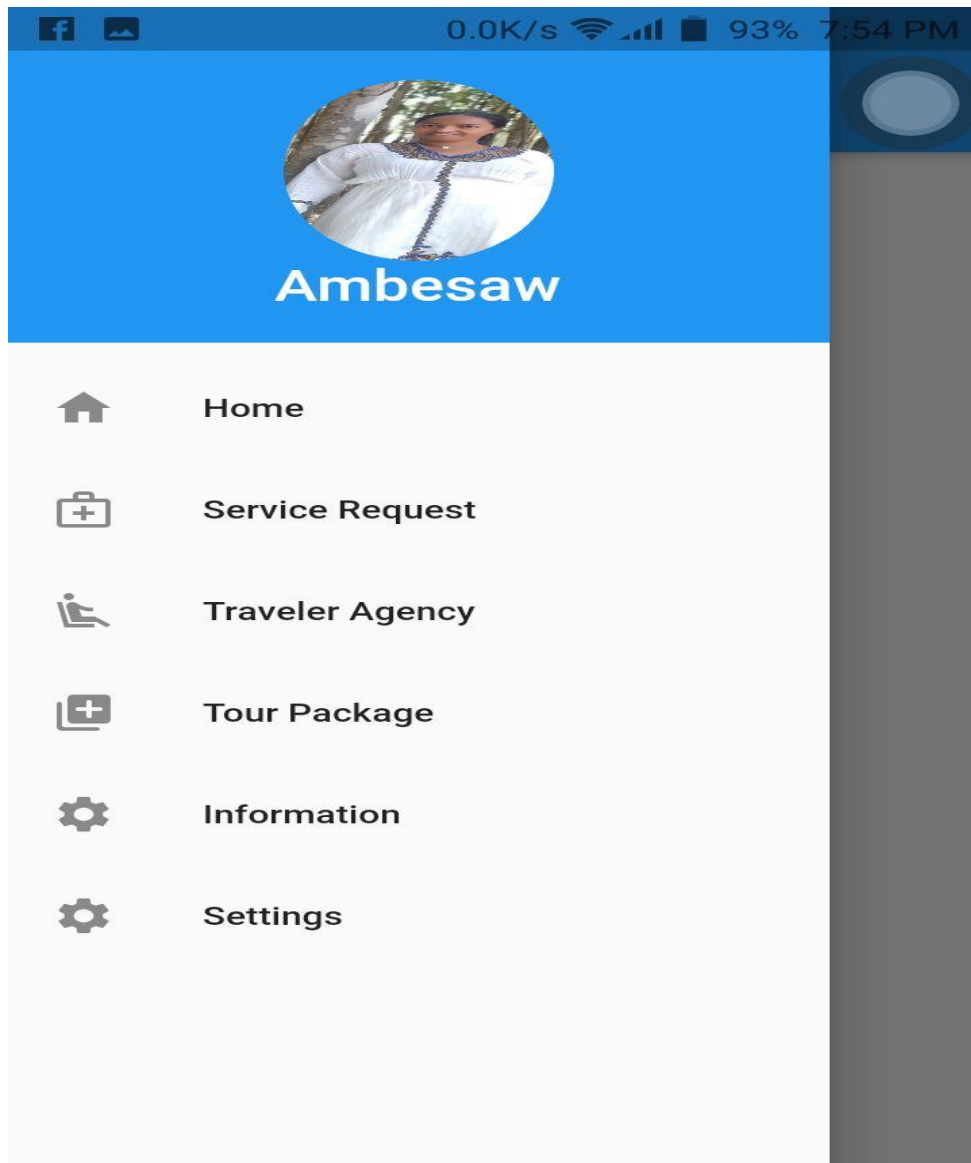


Figure 6. 8 User interface for Traveler page

- This is the traveler page of Travel Guide Application using android any one Travel can access this page if and only they must register first in the system to get access.

6.8. Testing

In this section of the project we made different tests on the system to make sure the system finally ready to be presented and be used ultimately by the users below are few of the testing we used and there inevitable results.

6.8.1. Testing Tools and Environment

The system is going to be implemented for Ethiopia Tourism Bureau since our system is Android based it will be store in google play store and accessed by any one of the user.

From user's point of view, they can use it from anywhere through by downloading from google play store, Here the hardware requirements that are need to be part of the system are minimum.

6.8.2. Unit Testing

Using unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness. In this level of testing process.

- It is also called component testing
- It is performed on standalone module to check whether it is developed correctly or not.
- Example if you take login whether it perform correctly or not.

Table 6. 9 Unit Test For Authenticated Login Page

Unit test =authentication of login page

Assumption =login to the user's appropriate page

Test data=username and password(empty,valid,invalid)

Steps to be executed	Test data	Expected result	Actual result
Empty username and password	Username="", Password=""	Please Enter valid address and password	aPassword field required and validusername field
Invalid username or password	Username="Ambesaw" Password="wku@gmail"	Please enter a valid email address and a valid password	Enter valid username and password
If valid username and password	Username= Yonassimachew56@gmail.com Password=Ambesaw@12	Successfully logged in to system	Successfully login to your Account

Table 6. 10 Unit Test For Register Travel Agency

Unit test =Register Traveler Agency

Assumption =Admin register Traveler agency

Test data=Name,Description,Address,PhoneNumber,Email>Password(empty,valid,invalid)

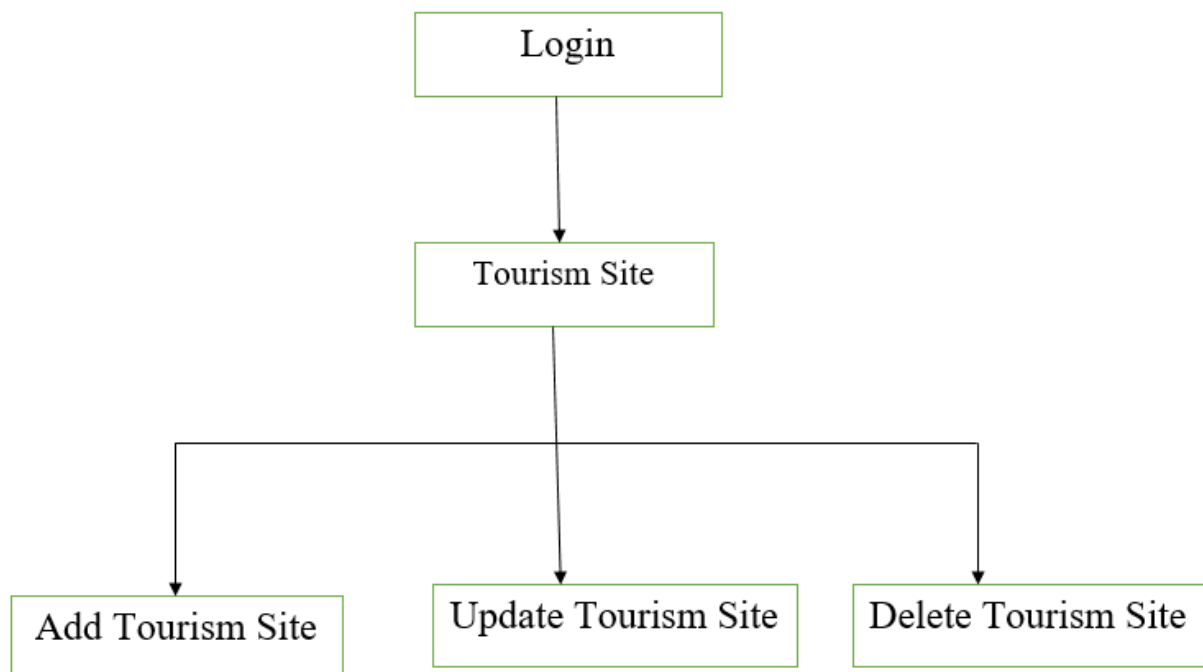
Steps to be executed	Test data	Expected result	Actual result
	Empty username and password	Name, Description, Address cannot empty.	phoneName, andDescription, phone Number, password and Address field required
	Invalid username and password	name="234", orDescription="55", Address="123" Phone number="097562" Email =" Ambes.com" Password ="1234"	Name, description and Address cannot be number. phone Number>10 Email format is invalid. Password is greater than 5 and 20 character. Email format is invalid. Password is greater than 8 character
If username and password	validUsername= Yonassimachew56@gmail.com Password=Ambesaw@12	Travel agency Register in the system	SuccessfullyTravel agency register successfully and get their username and password

6.8.3 Integration Testing

Once each individual part of the system is tested, every smallest unit is tested, different modules of the system are now integrated together and tested. We use integration to test their integration of different modules of proposed system (package module, post advertisement module, admin module) with each other in one.

Sample Teste

- Evaluate the functionality of the subsystem after combination all individual function.
- Identify the independence of each subsystem with other subsystems.



In the integration testing check whether the integrated function work correctly or not as we see from the above figure our group member check the functionality of individual could not affect with each other when we integrate the individual model so in this system all of integrated module was tested

Figure 6. 9 Integration Testing for travel guide application

6.8.4. System Testing

In this level of the testing process, we have examined how nicely the subsystems of the whole travel guide application are worked together to achieve the desired goal.

In this testing, the team perform over all functional testing by checking whether it meets the required goal.

Table 6. 11 System testing for authenticated login page

System Test =Authentication of login attempt for user part



Assumptions =Redirect to UserHomePage.dart .dart

Test data

✓ Username (valid username, invalid username, empty)

Password(valid password,invalid password, empty)

Steps to beData
executed

Empty usernameEmpty data
and password
field and click
login

Enter invalidUsername=" alem11",
username andPassword="abes"
password

Enter validUsername="
username andyonassimachew562gmail.com",
password Password="Ambesaw@12"

result

Please enter a valid
Email address and
password

Please enter a valid
email and password

Redirect=
UserHomePage.dart

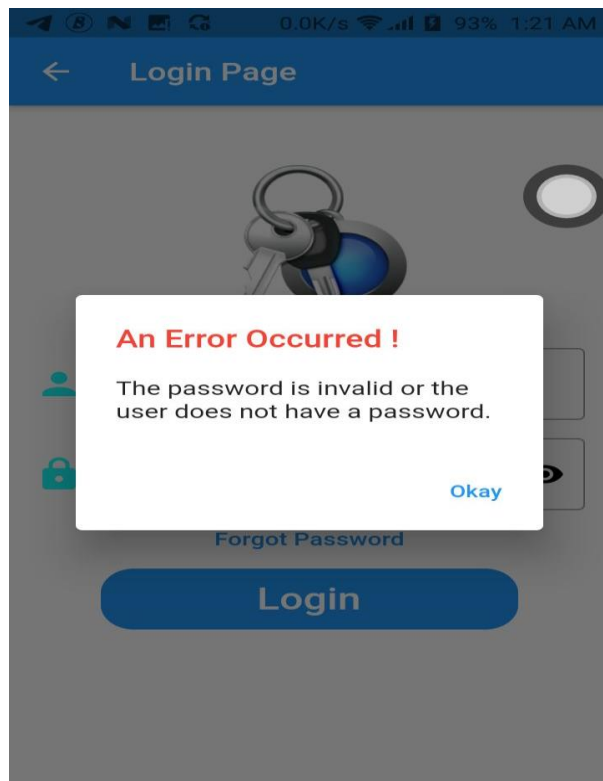
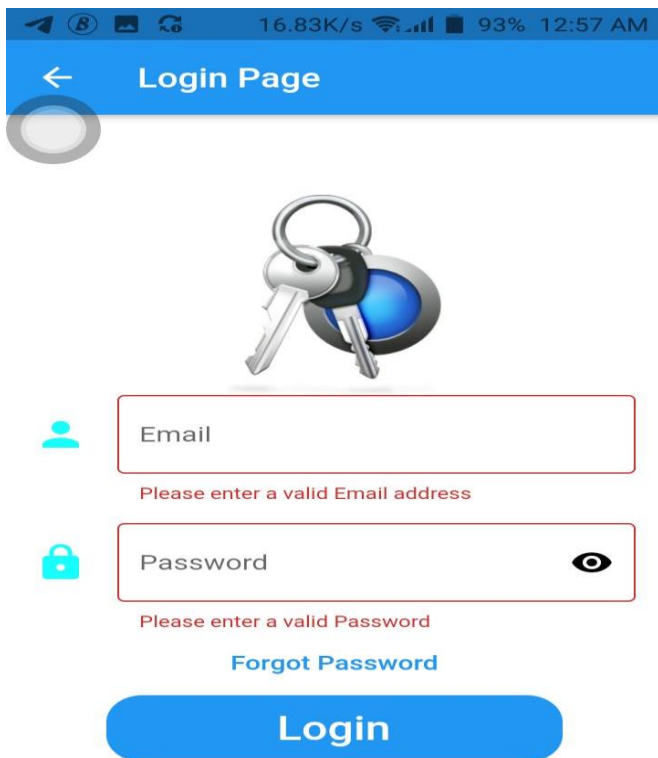


Figure 6. 10 System Testing For Authenticated Login Page

Table 6. 12 System Testing For Order Service For User Part

System Test =Order Service for user



Assumptions =Redirect to travelerAgencyDashBord.dart **.dart**

Test data

- Name (valide, invalid, empty)
- Last Name (valide, invalid, empty)
- Phone number (valide, invalid, empty)
- Nationality (valide, invalid, empty)
- Name(valide,invalid,empty)

Steps to be executed	Data	result
Empty Name and all others filled and click login button	Empty data	All filled are not empty
Enter invalid username and password	Phone number=<10 digit and valid for other the other field	Please enter a valid email and password
Enter All field with valid data and click register	All fields with valid data	Display all field register success

0.0K/s 94% 1:26 AM

← Order Service

Name

Name cannot be empty

Last Name

Name cannot be empty

Phone number

Phone Number cannot be empty

Nationality

Name cannot be empty

Source

Destination

0.0K/s 94% 1:29 AM

← Order Service

Agency beteseb travel and tourism

Name ambesaw

Last Name simachew

Phone number 0986542

Value are not Between 9 and 20 characters !

Nationality Ethiopia

Source wolkie

Destination bahir dar

Transnortation

Figure 6. 11 System Testing For Order Service For User Part

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1 Conclusion

It is known that developing a system for an organization is not easy. But the team have tried its best and developed interesting android and iOS application which is called a flutter application for Ethiopia Tourism Bureau. This application allows one to easily access the information of tourism sites in Ethiopia. This can be done by identifying problems of an existing system in the Ethiopia Tourism industry. Some of the issues that we will investigate are how hard it is to know the location of different tourism sites and how they find the right people to contact, and another difficulty is that there are many fake tourist sites. So in this application we shall have to only put tourism sites that have been approved by the Ethiopia Tourism Board. We will build a system to solve all existing problems in the tourism industry; the system will have many modules. Since this system is android and iOS application everyone who live in the world can download the application from google play store and easy use it to visit different place in Ethiopia, to order service from Travel agency, to visit city in Ethiopia, to see the weather condition in Ethiopia and plane tripe according to weather, to see different special event in Ethiopia like timket , and this application also gives access to know the current location of the user .The system is flexible, accurate and attractive GUI and the team confidently can say that the system accomplish its objective and completed successfully.

Generally, we conclude that the system is an android application that works in both android and iOS and it supports two languages i.e. English and Amharic and also the system shows the starting and End destination of the user.in this system the user can send travel request to travel agency it is also easy and user friendly i.e. anyone can access and easy to use the application without any Guide.

7.2 Recommendation

While doing this system the team has faced different type of challenges. But by the cooperation of all the group members and the advisor, the team is now able to reach to the final result. All the group members strongly fought these challenge.

We would like to recommend some of the extra feature to this system which we could not do due to time limitation and unavoidable reasons and challenges including the following feature.

- The system Travel guide android application allow the user to send request to travel agency but the travel agency not retrieve the acceptance or reject directly to the user page rather it send the message to his/her phone as a message.
- If this application contains video stream more about tourism the application become more powerful, but this application not contain video tour.
- The system could not consider the payment cost for hotel reservation and for travel agency.
- Payment transaction with bank database.

Appendix I: Sample Source Code

```
//Authentication Service for all user login
```

```
import 'package:final_tour_guide_app/travelerAgency/teravelerAgencyRegistration.dart';
```

```
import 'package:firebase_auth/firebase_auth.dart';
```

```
import 'DatabaseManager.dart';
```

```
class AuthenticationService {
```

```
  final FirebaseAuth _auth = FirebaseAuth.instance;
```

```
// registration with email and password
```

```
  Future createUser(
```

```
    String name, String lastName, int phoneNumber,String emergencyContactName,String  
    emergencyContactPhone,String nationality ,String email
```

```
    ,String password,String confirmPass, String imageURL ) async {
```

```

try {
    UserCredential result = await _auth.createUserWithEmailAndPassword(
        email: email, password: password);

    User user = result.user;

    await DatabaseManager().createUserData(name, 'users', imageURL, user.uid);

    await DatabaseManager().createUserfullData(name, lastName, phoneNumber,
emergencyContactName, emergencyContactPhone, nationality, user.uid);

    return user;
} catch (e) {
    print(e.toString());
}
}

```

//zara yetechemer

```

Future createNewUseragency(String name, String description,String address ,int
phoneNumber, String email, String password,String transportationService,String
guiderService,String hotelService) async {

```

```

    try {
        UserCredential result = await _auth.createUserWithEmailAndPassword(
            email: email, password: password);

        User user = result.user;

        //await DatabaseManager().createUserData(name, 'users', 100, user.uid);

        await AgencyRegistration().createUserDataAgency(name, "agency", 100,user.uid);

        await AgencyRegistration().createUserfullDataAgency(name, description, address,
phoneNumber, transportationService, guiderService, hotelService,user.uid);

        return user;
    } catch (e) {

```

```

        print(e.toString());
    }
}

// sign with email and password

Future loginUser(String email, String password) async {
    try {
        UserCredential result = await _auth.signInWithEmailAndPassword(
            email: email, password: password);
        return result.user;
    } catch (e) {
        print(e.toString());
    }
}

// signout

Future signOut() async {
    try {
        return _auth.signOut();
    } catch (error) {
        print(error.toString());
        return null;
    }
}

```

```

}

//Tourism Site Registration Sample code

import 'dart:io';

import 'package:final_tour_guide_app/stting/language_constants.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:image_picker/image_picker.dart';
import 'package:firebase_storage/firebase_storage.dart';

class TourismRegistration extends StatefulWidget {
  TourismRegistration({Key key}) : super(key: key);

  @override
  _TourismRegistration createState() => _TourismRegistration();
}

class _TourismRegistration extends State<TourismRegistration> {
  //String uid;

  final GlobalKey<FormState> _registerFormKey = GlobalKey<FormState>();
  TextEditingController nameInputController;
  TextEditingController descriptionInputController;
  TextEditingController addressInputController;
  TextEditingController latitudeInputController;
  TextEditingController longitudeInputController;
  TextEditingController imageInputController;

```

```

TextEditingController categoryInputController;

File image;

String imageURL;

@override
initState() {

    imageInputController = new TextEditingController();

    nameInputController = new TextEditingController();

    descriptionInputController = new TextEditingController();

    addressInputController = new TextEditingController();

    latitudeInputController = new TextEditingController();

    longitudeInputController = new TextEditingController();

    categoryInputController=new TextEditingController();

    super.initState();

}

final picker = ImagePicker();

Future getImage() async {

    final pickedFile = await picker.getImage(source: ImageSource.gallery);

    setState(() {

        if (pickedFile != null) {

            image = File(pickedFile.path);

        } else {

            print('No image selected.');
```

```

});
}

//image upload

Future uploadFile() async {
  Reference storageReference =
  FirebaseStorage.instance.ref().child(image.path);
  UploadTask uploadTask = storageReference.putFile(image);
  print('File Uploaded');
  imageURL = await (await uploadTask).ref.getDownloadURL();
}

//clear text

void clearText() {
  nameInputController.clear();
  descriptionInputController.clear();
  addressInputController.clear();
  latitudeInputController.clear();
  longitudeInputController.clear();
  categoryInputController.clear();
}

@override
Widget build(BuildContext context) {
  return Scaffold(

```

```

appBar: AppBar(
  title: Text("Register Tourism Site"),
),
body: Container(
  padding: const EdgeInsets.all(20.0),
  child: SingleChildScrollView(
    child: Form(
      key: _registerFormKey,
      child: Column(
        children: <Widget>[
          TextFormField(
            decoration: InputDecoration(
              labelText: 'Name', hintText: " Name"),
            controller: nameInputController,
            validator: (value) {
              String pattern=r'^[a-zA-Z]';
              RegExp regExp=new RegExp(pattern);
              if (value.length < 2) {
                return "Name is not less than 2 character";
              }
              else if (value.isEmpty){
                return "Please enter Name.";
              }else if(value.length>40){
                return "Name is not grater than 40 character";
              }else if(!regExp.hasMatch(value))

```

```

        {
            return "Enter valid Name";
        }
        return null;
    })),
    TextFormField(
        decoration: InputDecoration(
            labelText: 'Description', hintText:"Description"),
        controller: descriptionInputController,
        validator: (value) {
            String pattern=r'^[a-zA-Z0-9]';
            RegExp regExp=new RegExp(pattern);
            if (value.length < 10) {
                return "Description is not less than 2 character";
            }
            else if (value.isEmpty){
                return "Please enter Description.";
            }
            else if(!regExp.hasMatch(value))
            {
                return "Enter valid Description";
            }
            return null;
        }
    ),

```

```

TextFormField(
  decoration: InputDecoration(
    labelText: 'Address', hintText:"Address"),
  controller: addressInputController,
  validator: (value) {
    String pattern=r'^[a-zA-Z]';
    RegExp regExp=new RegExp(pattern);
    if (value.length < 2) {
      return "Address is not less than 2 character";
    }
    else if (value.isEmpty){
      return "Please enter Address.";
    }
    else if(!regExp.hasMatch(value))
    {
      return "Enter valid Address";
    }
    return null;
  }
),

```

```

TextFormField(
  decoration: InputDecoration(
    labelText:'Latitude', hintText: "Latitude"),
  controller: latitudeInputController,
  validator: (value) {

```

```

        double lat=double.parse(value);

        //      String      pattern=r'^(-{1}?(?:([0-9]{0,10}))|([0-9]{1}))?(?:([0-
144 9]{0,9}))?(?:\.[([0-9]{1,6}))?$',

        // RegExp regExp=new RegExp(pattern);

        if (value.length < 2) {

            return "Latitude is not less than 2 character";

        }

        else if (value.isEmpty){

            return "Please enter Latitude.";

        }else if(value.length>10){

            return "Latitude is not grater than 10 character";

        }

        /*else if(!regExp.hasMatch(value))

        {

            return "Enter valid Latitude";

        }*/

        else if(lat > 90 && lat < -90){

            return "Latitude is between 90 and -90 ";

        }

        return null;

    }

),

    TextFormField(

        decoration: InputDecoration(

```

```

        labelText: 'Longitude', hintText: "Longitude"),
controller: longitudeInputController,
keyboardType: TextInputType.number,
validator: (value) {
    double lat=double.parse(value);

    //      String      pattern=r'^(-{1}?(?:([0-9]{0,10}))|([0-9]{1}))?(?:([0-
9]{0,9}))?(?:\.[([0-9]{1,6}))?$',

    // RegExp regExp=new RegExp(pattern);
    if (value.length < 2) {
        return "Longitude is not less than 2 character";
    }
    else if (value.isEmpty){
        return "Please enter Longitude.";
    }else if(value.length>10){
        return "Longitude is not grater than 10 character";

    }
    /*else if(!regExp.hasMatch(value))
    {
        return "Enter valid Longitude";
    }*/

    else if(lat > 180 && lat < -180){
        return "Longitude is between 180 and -180 ";
    }
    return null;
}

```

```

    }
),
TextFormField(
  decoration: InputDecoration(
    labelText: 'Category', hintText: " Category"),
  controller: categoryInputController,
  validator: (value) {
    String pattern=r'^[a-zA-Z]';
    RegExp regExp=new RegExp(pattern);
    if (value.length < 2) {
      return "Category is not less than 2 character";
    }
    else if (value.isEmpty){
      return "Please enter Category.";
    }else if (value.length > 40){
      return "Category is not greater than 40 character";
    }
    else if(!regExp.hasMatch(value))
    {
      return "Enter valid Category";
    }
    return null;
  }
),
ElevatedButton.icon(
  icon: Icon(Icons.photo),

```

```

onPressed: () {
    getImage();
},
label: Text('Pick image From Gallery'),
),
RaisedButton(
    child: Text("Register"),
    color: Theme.of(context).primaryColor,
    textColor: Colors.white,
    onPressed: () async {
        //uploadFile();
        var po=FirebaseStorage.instance.ref().child(image.path);
        UploadTask uploadTask = po.putFile(image);
        imageURL=await (await uploadTask).ref.getDownloadURL();
        if (_registerFormKey.currentState.validate()) {
            double latu = double.parse(latitudeInputController.text);
            double longu = double.parse(longitudeInputController.text);

            Map<String, dynamic> data = {
                "tourism_Image": imageURL,
                "tourism_Name": nameInputController.text,
                "tourism_Description": descriptionInputController.text,
                "tourism_Address": addressInputController.text,
                "tourism_Location": GeoPoint(latu, longu),
                "tourism_Category":categoryInputController.text,
            };

```

```

FirebaseFirestore.instance
    .collection("TouresimSite")
    .add(data);
clearText();
//.catchError((err) => print(err))
//.catchError((err) => print(err));
} else {
showDialog(
    context: context,
    builder: (BuildContext context) {
    return AlertDialog(
        title: Text(getTranslated(context,"Error")),
        content: Text(getTranslated(context,"The text is no fill")),
        actions: <Widget>[
            TextButton(
                child: Text(getTranslated(context,"Close")),
                onPressed: () {
                    Navigator.of(context).pop();
                },
            )
        ],
    );
});
},
),      ],      ), ))); }}

```

References

- [1] Aishwarya Bhat, Joylin Rosario, Ayesha Thasneema, Judith Jennifer Rodrigues, "Design and Development of Tour Management," *International Journal of Engineering Research & Technology*, 2018.
- [2] Design and Development of Tour Management System using Android, "Aishwarya Bhat, Joylin Rosario, Ayesha Thasneema, Judith Jennifer Rodrigues," *International Journal of Engineering Research & Technology*, 2018.
- [3] "OVERVIEW," 2021. [Online]. Available: <https://www.devex.com>.
- [4] S. P. Deepika, M. Ananthi, P. Rajkumar, "Video-Based Face Detection Process," *International Journal of Advance Research, Ideas and Innovations in Technology*, p. 84, 2018.
- [5] M. A. PATLE, "Onlin Tour & Travelers," UNIVERSITY OF MUMBAI, MUMBAI, 2014.
- [6] G. Venetta, "RESEARCH," 2017. [Online]. Available: https://www.academia.edu/33581857/UGANDA_TOURISM_MOBILE_GUIDE_APP_proposal_docx.
- [7] Muzammil, "Importance of Firebase in Mobile App Development," 10 September 2020. [Online]. Available: <https://www.aalpha.net/blog/firebase-in-mobile-app-development/>.
- [8] Alexander Smirnov, Alexey Kashevnik, Nikolay Shilov, Nikolay Tesly, Anton Shabaev, "Mobile Application for Guiding Tourist Activities: Tourist Assistant," *ISSN*, 2014.
- [9] U. LINGALA, "ANDROID AND ANDROID," 2012.
- [10] M. Ch., "Iterative Application Development Methodology: How We Ensure App Quality," 05 FEB 2017.

