Driving School App (A License to Drive)

By

Capt Fahan Saleem Taas

GC Zia ur Rehman Ch

GC Jawad Ali

Submitted to the Faculty of Computer Software Engineering, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a

B.E. Degree in Computer Software Engineering

June 2018

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is officially state that the thesis work contained in this report

"Driving School Guide (A license to drive)"

is carried out by:

Farhan Saleem Taas, Zia Ur rehman Ch, Jawad Ali

under my supervision and under my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Computer Software Engineering from National University of Sciences and Technology (NUST).

Approved By: Signature_____

Supervisor: Lt Col. Dr. Adil Masood Sadique Phd.

MCS, Rawalpindi

ABSTRACT

Driving license app is design to make an app for the traffic rules understanding and preparation of testing for driving license. This app will contain theory test which is quite necessary before going towards licensing. The project is designed to allow users to test their abilities after getting proper guideline about traffic rules. The system will be accessible from android mobile phones. This project explores the use of current technologies in software engineering world.

Overall, our project presents a complete understanding of driving rules and regulations for theory tests, on a mobile platform Android other than a website. This project involves some deliverables which are documents and an end app.

COPYRIGHT STATEMENT

We hereby declare the work contained in the report and the intellectual content of this report are the product of our work. This thesis report has not been formally published in any structure not does it include any verbatim of the published resources which could be treated as violation of international copyright degree. We also affirmed that we do recognize the terms “plagiarism’ and ‘copyright’ that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our supervisor, whose endless support gave us strength and courage to complete this work.

To our beloved parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

All glory goes to Allah almighty who lead us to extent, May all glory, honor and adoration be unto thy Name.

Our special thanks go to our supervisor Lt Col. Dr. Adil Masood Sadique for his guideline, in completing this project successfully. We wish to add our acknowledgement to the people from MCS, where it all started, especially Dr. Bilal Rauf, Madam Ayesha Naseer, Dr. Hammad Afzal and Dr. Tauseef Ahmad Rana.

A deep gratitude towards Dr. Adnan Ahmad Khan (Head of Computer Software Department) for his guidance and facilitation for the project.

Table of Contents

List of Tables

Table names **Page number**

List of Figures

Figure names **Page number**

Chapter 1

1. Introduction

1.1. Motivation

Android's market share in the mobile operating system market is rapidly increasing. By the end of 2010 android is becoming the second famous worldwide operating system. We have decided to leverage the android platform to help the users in driving perspective. There are many android applications regarding driving guides but no one offers the online theory testing rather than websites. Our app would be different from other driving apps with respect to its features regarding traffic rules learning and theory test for licensing.

1.2. Purpose

The aim of the project is design to make an app for the traffic rules understanding and preparation of testing for driving license. In any driving school user gets a booklet which guides the user to understand the traffic rules and driving guidelines, we have automated that booklet in the form of a mobile app. Basically it will provide guideline to learn basic traffic sign and rules which are necessary for the safe driving. This app will cover almost all aspects of a driving school and act as a guide towards learning.

This app will contain theory test which is quite necessary before going towards licensing. The project is designed to allow users to test their abilities after getting proper guideline about traffic rules. The system will be accessible from android mobile phones. This project explores the use of current technologies in software engineering world.

1.3. Project Scope

This project is basically built for all cities in Pakistan which contain a data regarding traffic, in which we check out road safety measurements, updates, and particular best driving schools in major cities. It is very help full for those who are new to driving and for those who want to apply for driving tests. There are tutorials for driving which enable users to learn through simulation.

Overall, our project presents a complete understanding of driving rules and regulations for theory tests, on a mobile platform Android other than a website. This project involves some deliverables which are documents and an end app.

1.4. Deliverables

Deliverable Name	Deliverable Summary Description
Software requirement Specification (SRS) Document	Detailed description of functional and non-functional requirements and system features.
Design Documentation	Complete description of HOW the system will do. Design models are included.
Testing Documentation	Complete system is tested corresponding to specification. System is tested at all levels of Software Development Life Cycle (SDLC).
Code	Complete Code with API.
External Documentation/Thesis	Complete working system.

Table 1: Deliverables

Chapter 2

2. Literature Review

2.1. Introduction

We have proposed to implement a Driving School an app, by using this android application user will be able to prepare him or her before applying for driving test. In any driving school user gets a booklet which guides the user to understand the traffic rules and driving guidelines, we are going to automate that booklet in the form of a mobile app. Basically it will provide guideline to learn basic traffic sign and rules which are necessary for the safe driving. This app will cover almost all aspects of a driving school and act as a guide towards learning.

2.2. Problem Domain

- New drivers or learners face a lot of problem while getting their driving license.
- There is no guideline for drivers to handle real time scenario problems during driving in Pakistan
- There is no sample test for driving license is available in the market
- People are not aware of the general traffic rules and signs.
- The procedure to get a traffic license issued it tedious and time consuming.
- Traffic licenses are easily forged and fake licenses are issued.
- There is no current system for on spot verification of licenses
- Lives are at stake.

2.3. Related Work

There are many android applications regarding driving guides but no one offers the online theory testing rather than websites. There are tutorials for driving which enable users to learn through simulation. Many apps provide displaying traffic conditions to finding parking spots, locating the cheapest gas stations near you or remembering where you park your car and many other requirements for your driving or travelling needs. Many developers had tried best to provide the best solutions regarding driving and licensing guide.

2.4. Technological requirements

Our project requires following software and hardware requirements.

2.4.1. Software Requirements

The Software(s) required for the implementation of our project includes:

- JAVA / Android Platform OR
- Android studio SDK version 1.0.0.
- Adobe Photoshop for Graphic designing
- Unity version 5.6.2017 platform for real time simulation.

2.4.2. Hardware Requirements

The Hardware required for the implementation of our project includes:

- Smart Phone (Available for Android).
- Data Cable for the mobile and application interfacing.

Chapter 3

3. System Requirement Specification

3.1. Purpose

The purpose of this report is to fully document the specifications and requirements for android app which is driving school (for driving licensing). The audience of this document will be the technical professionals developing the software.
The Driving School android application will teach you about the basic rules of traffic and will give information about the traffic police. In any one needs help about anything regarding driving; it will provide user with the basic information. It will guide about the theory test for driving. This app will provide user with full information about safe and responsible driving, guidelines for children, list of best driving schools in major cities, and sample sign test and questionnaire for learners.

3.2. Document Conventions

The conventions used to prepare the document is given bellow

1 Font – Times New Roman, size 12

2 Main headings, Bold size 18

3 Sub headings, Bold size 14

4 Sub-sub headings, Bold size 12

3.3. Intended Audience and Reading Suggestions

This report is intended for the developers, analysts, designers, coders, software engineers. So by the help of this document we can understand what we are going to do in future and what we will make. It shows the way to take the next step so we can say that it plays very important role in the software development. In this document, first we are going to collect the requirements, then there will come a planning phase that how we are going to develop the application and after that we will do analysis of this application and in the result of this analysis we will get some specific requirements so after that we will document those requirements. To some extent designing is also included in this document.

3.4. Examiners/Evaluators

The document will provide the FYP evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

3.4.1. Developers

The document will provide guidance to the developers to determine what the requirements are and how they should continue with the project.

3.4.2. Project Supervisor

This document will be used by the project supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly and completely.

3.4.3. Project Testers

Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. It will help in building up test cases for the testing process. This way testing becomes more methodically organized.

3.4.4. Up gradation Engineers

Up gradation engineers can review projects capabilities and more easily understand where their efforts should be targeted to improve or add more features to it. It sets the guidelines for future developments.

3.4.5. End Users

This document can be read by the end users if they wish to know what the project is about and what requirements have been fulfilled in this project.

3.5. Product Scope

This project is basically built for all cities in Pakistan which contain a data regarding traffic, in which we check out road safety measurements, updates, and particular best driving schools in major cities. This app can be used by people who have smart phone. It is very help full for those who are new to driving and for those who want to apply for driving tests.

.

3.6. Reference

We got the referred material for SRS template from following website:

- https://www.google.com.pk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwje44Tg8LrXAhXnIsAKHXWqAdsQFggpMAE&url=https%3A%2F%2Fweb.cs.dal.ca%2F~hawkey%2F3130%2Fsrs_template-ieee.doc&usg=AOvVaw3m6N2fEEPiLkM7-nflitlv

- Project Synopsis: An android document with complete information containing extended title, brief description of project, scope of work, academic objectives, applications, previous work, material resources required and special skills required has already submitted to the department.

3.7. Overall Description

The requirements analysis processes play a critical role for the success of any software engineering project. In requirements analysis a process is defined that determines the tasks identify the needs and conditions to design any innovative software product or to make modifications in any existing system. The process of requirement gathering and analysis revolve around all the stakeholders' conflicting requirements, and analyzes the documentation and validation of the system. All the gathered requirements should be measurable, feasible, testable, and related to the concerned needs of the developing system. In software- engineering, requirements analysis is described as a three-step process as:

- Requirements Elicitation
- Requirements Analysis
- Requirements Documentation

Requirements Elicitation

Elicit means to gather, or extract; and thus requirements elicitation means gathering requirements or discovering requirement types from stakeholders or from project documentation. It is an iterative process and consists of many activities including establishing objectives, understanding background, organizing knowledge, and collecting requirements.

Requirements Analysis

The aim of requirements analysis is to discover problems with the system requirements, especially incompleteness and inconsistencies. Detailed analysis usually takes place after the initial draft of the requirements document is produced. It is concerned with incomplete set of requirements, which has not been discussed by stakeholders or the requirements which are not discussed earlier can cause problem later.

Requirements Documentation

The third category is about documenting the requirements. Requirements are considered in various forms, including summary lists, natural language documents, visual documents, use cases, user stories, or process specifications. Software requirement specification document is classified in different ways according to the stakeholders' needs. In this report the following sections include the different categories of requirements specification document that are necessary for designing this system: the functional requirements, system constraints, and the system requirements etc.

3.7.1. Product Perspective

The Driving School is an android based system. It can be accessed using Internet and also offline on any android phones.

Functional Requirements

There are various functions of the application which are listed below:

- The app should allow user to know driver fast guide
- The app should allow the user to share the data on social networks
- The app should allow user to get the navigation
- The app should allow user to know the knowledge of mandatory, informatory and warning signs
- The app should allow user to learn the guidelines for children
- The app should allow user to view the menu
- The app should allow user to get instructions about LTV / HTV licenses.
- The app should allow user to stimulate the car in real time for different scenarios
- The app should allow user to send and get notifications about weather update.

3.7.2. Operating Environment

The operating environment required for this project is:

Hardware requirements:

- Smart Phone (Available for Android).

Software requirements:

- JAVA / Android Platform OR
- Android studio.
- Adobe Photoshop for Graphic designing

3.8. External Interface Requirement

Interface requirements are of many types such as; User Interface, Software Interface, Communications Interface and Hardware Interface. These requirements are described as:

3.8.1. User Interfaces

A graphical user interface allows easy access to all features of the system and will be available in all workflow. To make easy learning for user and reduce the complexity, navigation options in every screen will remain same. For exception handling we have used user friendly messages to catch the errors. The user interface should be

- Readable
- Easy to understandable
- Easy to operate

3.8.2. Hardware Interfaces

Driving school app will be implemented on mobile phone and it will require no additional hardware components. The required components of hardware would be:

- Touch screen
- Internet connection
- Wi-Fi connection

3.8.3. Software Interfaces

- Mobile device: Android 2.1 or later.
- Computer: Windows XP or later with C#.Net framework 4.0.

3.8.4. Communication Interfaces

The system would be connected to the internet, once user connected through browser, all the feature of the driving school app would be accessed.

3.9. Design Constraints

The system is built by using a standard android development tool:

- JAVA/ Android Platform OR
- Android studio

Standards Compliance

This system will follow the existing standards and regulations of Android app.

Hardware Limitations

This system does not support any other platform except android platforms.

3.10. Other non-functional Requirements

According to our project needs these are the following nonfunctional requirement of our system:

3.10.1. Security Requirements

As this application is designed for public benefits and it contains safety measures. Therefore, system requires no specific security constraints.

3.10.2. Scalability

This application would have ability to entertain numbers of query at a time because there are number of users in the specific place, and the projected load scenarios, the intention is for the system to be able to serve many queries per day.

3.10.3. Software Quality Attributes

In driving school app the first priority is given to design of graphical user interface. The system would be designed by considering the appealing design techniques to present the system in more attractive way, it is easy to navigate and having simplicity for the users. In case of any update and pop ups the user will provide with visual notifications.

3.10.4. Portability

Portability is not required in this project because it is developed only for android users**.**

3.10.5. Availability

This app would be available to users 24 hours a day, and 7 days a week, rather exception of being down for maintenance no more than one hour in a week.

3.10.6. Supportability

The source code developed for this system shall be maintained in configuration management tool.

3.10.7. General Constraints

Following are the constraints of our system

There is no maintainability of back up so availability will get affected.

There is no search box for the related items; user can only find the desired feature in main menu. This app has no multilingual support
Internet connection is required for online use.

This app requires the client to have a smart phone to run.

3.10.8. User Characteristics

The user of this system should have the
following characteristics User should be
familiar with Internet.
The user should be familiar of all the advanced terminologies related to android app.

The user must be educated enough to be able to use the app or user should know how to use it.

3.10.9. User Objectives

User will achieve the following objective by using this app:

- Awareness about traffic signs
- Famous driving schools in different cities
- Instructions for driving license
- Instruction for children for road safety
- Theory test preparation.

3.11. User Documentation

For the user documentation, a user manual will be provided with the system. It will include the details of the software's working. Help documents will also be a part of the system. The project report will also be available for the users which will highlight the system features, working and procedures.

Chapter 4

4. System Design Specification

4.1. Purpose

The purpose of this report is to fully document the high-level design framework for android app which is driving school (for driving licensing). The audience of this document will be the technical professionals developing the software.

4.2. Overview of document

The document is divided into sections and is already listed in the table of contents and figures list. However, here is a brief description of all the sections.

4.2.1. System Architecture Description

In this section, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by system Architecture diagram which shows an insider’s perspective of the system by describing the high level software components that perform the major functions to make the system operational.

4.2.2. Structure and relationships

This section ponders upon the interrelationships and dependencies among various components. It is mainly described by different diagrams which are further augmented by explanatory text. E.g. Class diagram and use case diagram.

4.2.3. User Interface Issues

This section presents the main principles of the product's user interface. Not touching about the technical details, the section is described by an overall diagram which is also augmented by explanatory text. Moreover, Activity diagrams, Sequence diagrams, and UI Design diagrams also elaborate the User Interface issues in a more intelligible manner.

Activity diagrams

Activity Diagrams follow a workflow-based approach to describe the overall functioning of the system. They are a very good means to see how various steps are involved in major tasks inside a system using a flow chart pattern without getting into the technical details.

Sequence diagrams.

Sequence diagrams show how different objects are involved in the completion of a functionality of the system. They have a unique format that allows the reader to see how many objects are used during their duration; for the completion of a system requirement.

UI Design

Some snapshots of graphical user interfaces are shown in this section that prototype the way a user shall be interacting with the system.

4.2.4. Detailed description of components

This section contains detailed description of all the major components of the system in a structured pattern (table), comprising of 10 x rows. The pattern (table) maintains symmetry in the document structure; and therefore it is followed for each of the components. Each part/row of the table is identified by a *label*, explaining the purpose of each point. The description of each point or the component being discussed, ponders upon the detailed account of it in the system.

4.2.5. Reusability and relationships to other products

This section focuses upon the Reusability aspects of the various components of the system. Since the project in hand is all new and doesn't carry out any enhancement work in the already existing system, so Reusability is just a recommended strategy to be employed while organizing various system components.

4.2.6. Design decisions and tradeoffs

This section highlights various design decisions and the ideas behind those. It enables the reader to understand the important crux of the design that is being used while excavating a bit more about the motivations behind those decisions.

4.3. Design Considerations

All design considerations were handled in this document.

4.3.1. Constraints

Design constraint

The system is built by using a standard android development tool:

- JAVA/ Android Platform OR
- Android studio

Standards Compliance

This system will follow the existing standards and regulations of Android app.

Hardware Limitations

This system does not support any other platform except android platforms.

4.3.2. General constraint

Following are the constraints of our system

- There is no maintainability of back up so availability will get affected.
- There is no search box for the related items; user can only find the desired feature in main menu.
- This app has no multilingual support
- Internet connection is required for online use.

This app requires the client to have a smart phone to run.

4.4. Design Methodology

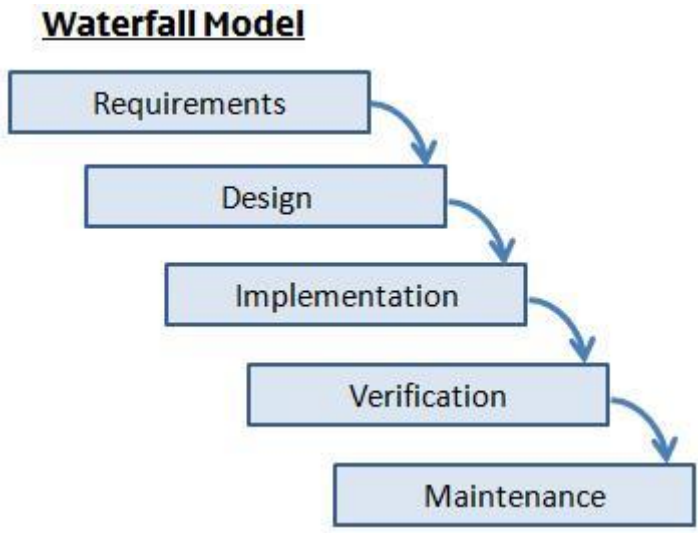


Figure 1: Design methodology

We have used Waterfall methodology which is a primarily top down approach. In this model we identify all distinct modules in the system and the interface between them. After that we gradually move down into each module to design the increasingly more detailed aspects.

4.4.1. Related Software and Hardware

This app is developed under the Android operating system using the following software and hardware components.

- Java JDK (java development kit)
- Android SDK (software development kit) tools
- Laptop and android device for development.

4.4.2. Android

Android's market share in the mobile operating system market is rapidly increasing. by the end of 2010 android is becoming the second famous worldwide operating system. We have decided to leverage the android platform to help the users in driving perspective. Android is basically designed by GOOGLE for mobile device which is consisting of an operating system, middleware, and key application. The platform of android software is also known as the android software stack. It is consists of the following different layered components:

❖ **Application Layer**

Application layer contains the core set of application which is delivered with the android operating system such as email, calendars, maps, SMS program, etc.

❖ **Application Framework Layer**

The App framework layer provides a set of application programming interface (APIS) used by the core application. Developers can also use this to develop their own application.

❖ **Libraries Layer**

This layer provides support to media and advanced development feature such as 2D & 3D graphics. It also provides a set of c/c++ libraries used by the various components of the android system.

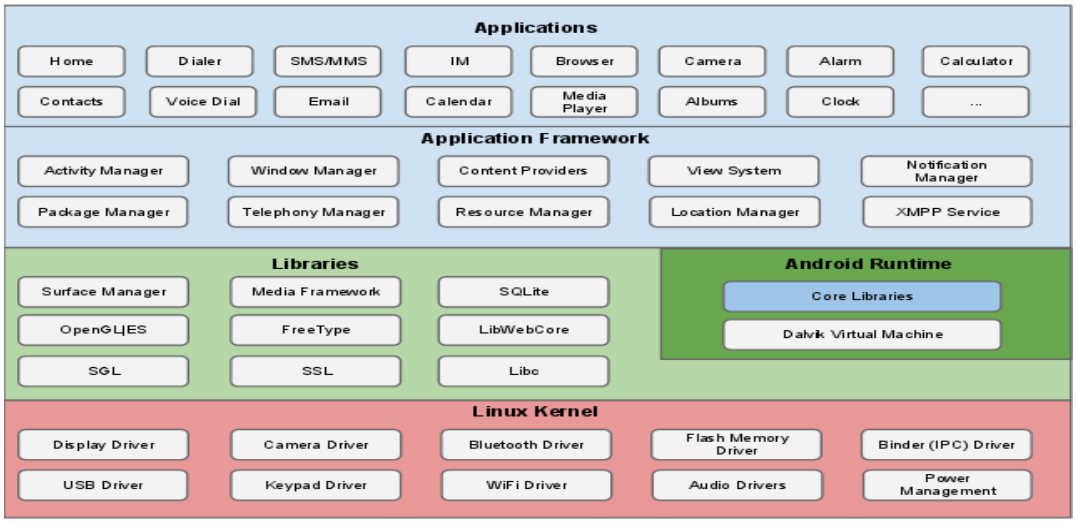
❖ **Android run time Layer**

It is related to java source files and includes the set of core libraries that provides most of the functionality available in the core java source files. In this layer instances of virtual machine for java based Application to run on are also created with Dalvik machine.

❖ **Linux kernel**

This is an abstraction layer between the hardware and the rest of the software stack. It provides core system such as security, memory management, networking, drivers, etc. for this app will mostly utilizing the application framework and libraries component from the android software stack to develop this android application. We will use some outside libraries such as data to store and retrieve information from database.

❖ **Android software stack, source**



4.5. System architecture description

The architecture provides the top level design view of a system and provides a basis for more detailed design work

4.5.1. Overview of module/components

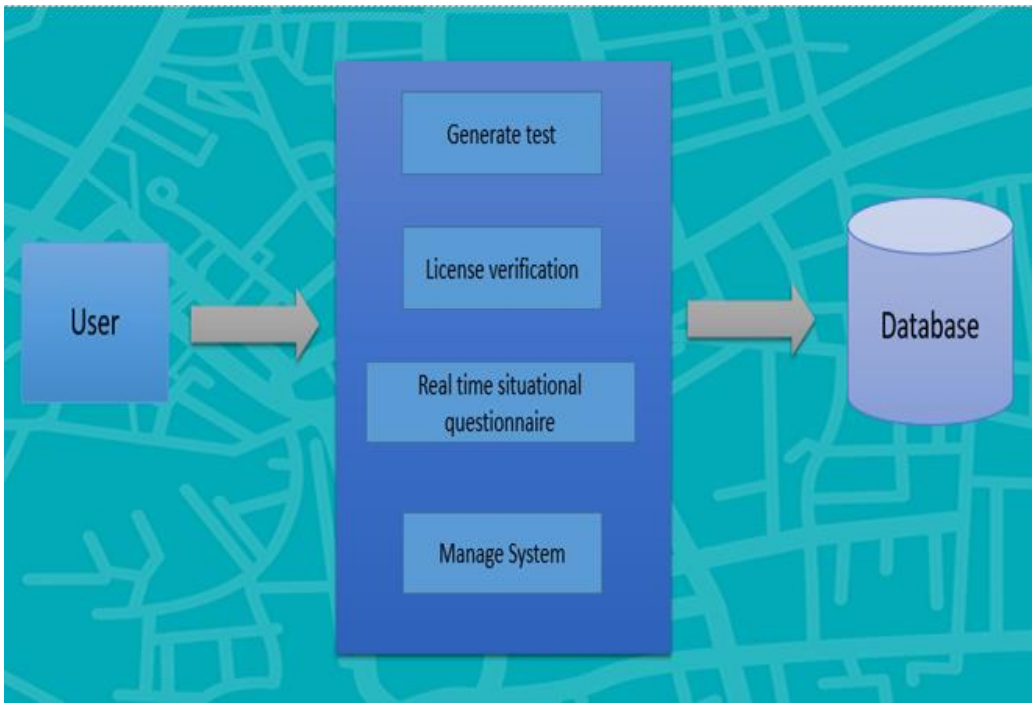


Figure 2: Overview of module

4.6. Structure and relationships

Focusing upon the internal structure of the system, this section ponders upon the interrelationships and dependencies among various components.

4.6.1. Use cases

Use cases describe the Actors of the system and their actions. It gives the overview about how the factors outside the system interact and what actions they perform on the system.

Actors	Use Cases
User of the app	1. User can register after installation 2. User can get on-hand guide while driving 3. User can get Instant help anytime regarding driving 4. User can apply for online theory test for practice of driving license test 5. User can have real Time situational simulations 6. User can analyze the result for theory test 7. User can also see the weather updates 8. User can verify his/her license through CNIC 9. User can get tutorials for driving

Figure 3: Use cases

4.7. Use case Diagram

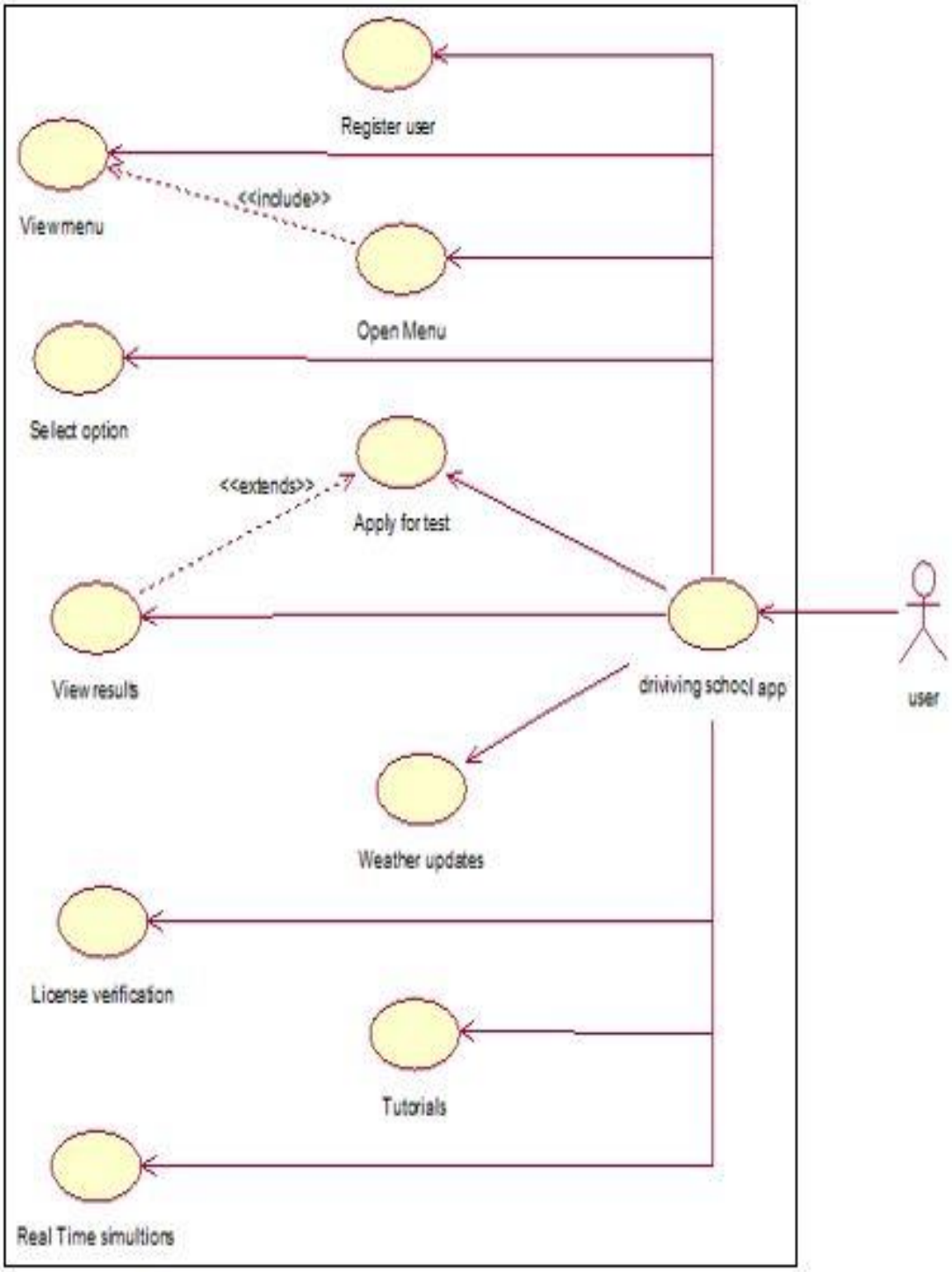


Figure 4: Use case diagram

4.8. Use case Description

4.8.1. Register User

Use Case ID:	1		
Use Case Name:	Register user		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user tries to sign up in to the system.		
Preconditions:	• User has to open the sign up page first.		
Post conditions:	• The System must record the membership information of the new member.		
Normal Flow (primary scenario):	• The member enters the membership details on the screen and clicks the sign up button. • The system checks for the availability of the username. • The system generates the membership ID. • The system records the membership information of the new member, in the database.		
Alternative Flows:	• The member enters the membership details on the screen and clicks the sign up button. • The system checks for the availability of the username. • The system displays an error report if the username is not available.		
Non-functional Requirements	• The system must perform an encoding technique such as hashing to save all passwords securely.		

Table 2: Use Case 1(Register User)

4.8.2. View Menu

Use Case ID:	2		
Use Case Name:	View Menu		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user tries to view the menu of the system		
Preconditions:	• User has to open the sign up page first.		
Post conditions:	• The System must describe the information regarding all the other features which are useful for users regarding road safety and driving.		
Normal Flow (primary scenario):	• The member enters the membership details on the screen and clicks the sign up button to view the menu.		
Alternative Flows:	• The system checks for the availability of the credentials to view the main menu. • The system displays an error report if the main menu is not available.		
Non-functional Requirements	• N/A		

Table 3: Use Case 2(View Menu)

4.8.3. Select Option

Use Case ID:	3		
Use Case Name:	Select Option		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	• A user can retrieve the information of any feature of this app as per his need by selecting option.		
Preconditions:	• User has to open the main menu by logging in to app.		
Post conditions:	• The System must reveal the information as per need of user.		
Normal Flow (primary scenario):	• The user enters the membership details on the screen and clicks the sign up button to view the menu. • The user enters into main menu where user can select the options.		
Alternative Flows:	• The member enters the membership details on the screen and clicks the sign up button. • The system checks for the availability of the main menu. • The system displays an error report if the main menu is not available. • The system will reset to the login page.		
Non-functional Requirements	• N/A		

Table 4: Use Case 3 (Select Option)

4.8.4. Apply for test

Use Case ID:	4		
Use Case Name:	Apply for test		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user tries to apply for driving license test		
Preconditions:	• User has to sign up first.		
Post conditions:	• The System must reveal a digital questionnaire to the user.		
Normal Flow (primary scenario):	• The user enters the main menu. • Then user applies for license driving test. • The system generates the questionnaire for the user. • The system records the user solution.		
Alternative Flows:	• The user enters the main menu. • Then user applies for license driving test. • The system checks for the availability of the test use case. • The system displays an error report if the test use case is not responding.		
Non-functional Requirements	N/A		

Table 5: Use Case *4 (Apply for test)*

4.8.5. View results

Use Case ID:	5		
Use Case Name:	View results		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user has done license test and now he want to view the results.		
Preconditions:	• User has to apply for the test first.		
Post conditions:	• The System must output the result for the given by the user.		
Normal Flow (primary scenario):	• The user enters the main menu. • The user enters the view results options. • The system checks for the availability of the result. • The system generates the result of applied test.		
Alternative Flows:	• The user enters the main menu. • The user enters the view results options.. • The system checks for the availability of the result use case. • The system displays an error report if the corresponding result use case is not available.		
Non-functional Requirements	• N/A		

Table 6: Use Case 5(View Results)

4.8.6. Weather updates

Use Case ID:	6		
Use Case Name:	Weather updates		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user tries to get the weather updates.		
Preconditions:	• User has to sign up first.		
Post conditions:	• The System must output the weather update for the user's desired location.		
Normal Flow (primary scenario):	• The user enters the main menu • The user selects the weather update option. • The user gives the desired location. • The system checks and updates the user with weather of his desired location.		
Alternative Flows:	• The user enters the main menu • The user selects the weather update option. • The user gives the desired location. • The system checks for the availability of the desired location. • The system displays an error report if the desired location is not available.		
Non-functional Requirements	N/A		

Table 7: Use Case 6(Weather update)

4.8.7. License verification

Use Case ID:	7		
Use Case Name:	License verification		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user tries to verify his license through his/her CNIC.		
Preconditions:	• User must be having his/her license. • User must be having his CNIC. • User must have registered the application.		
Post conditions:	• The System must show the verification of LTV_HTV license.		
Normal Flow (primary scenario):	• The user enters the main menu. • The user selects the license verification option. • The user provides his/her CNIC number. • The system shows the computerized driving license corresponding to given CNIC number		
Alternative Flows:	• The user enters the main menu. • The user selects the license verification option. • The user provides his/her CNIC number. • The system checks for the availability of the license. • The system displays an error report if the license is not available.		
Non-functional Requirements	N/A		

Table 8: Use Case 7(License Verification)

4.8.8. Tutorials

Use Case ID:	8		
Use Case Name:	Tutorials		
Actors:	User of the App		
Created By:	User	Last Updated By:	User
Date Created:	19/12/2017	Date Last Updated:	19/12/2017
Description:	A user will be able to see the video tutorials of different driving situations.		
Preconditions:	• User has to sign in first.		
Post conditions:	• The System must view the tutorials to the user.		
Normal Flow (primary scenario):	• The user enters the main menu. • The user selects the tutorials option. • The list of videos of different situation will be available to the user. • The system plays the selective video to the user.		
Alternative Flows:	• The user enters the main menu. • The user selects the tutorials option. • The list of videos of different situation will be available to the user. • The system displays an error report if the video is not playing.		
Non-functional Requirements	N/A		

Table 9 : Use Case 8 (Tutorials)

4.9. Class Diagram with description

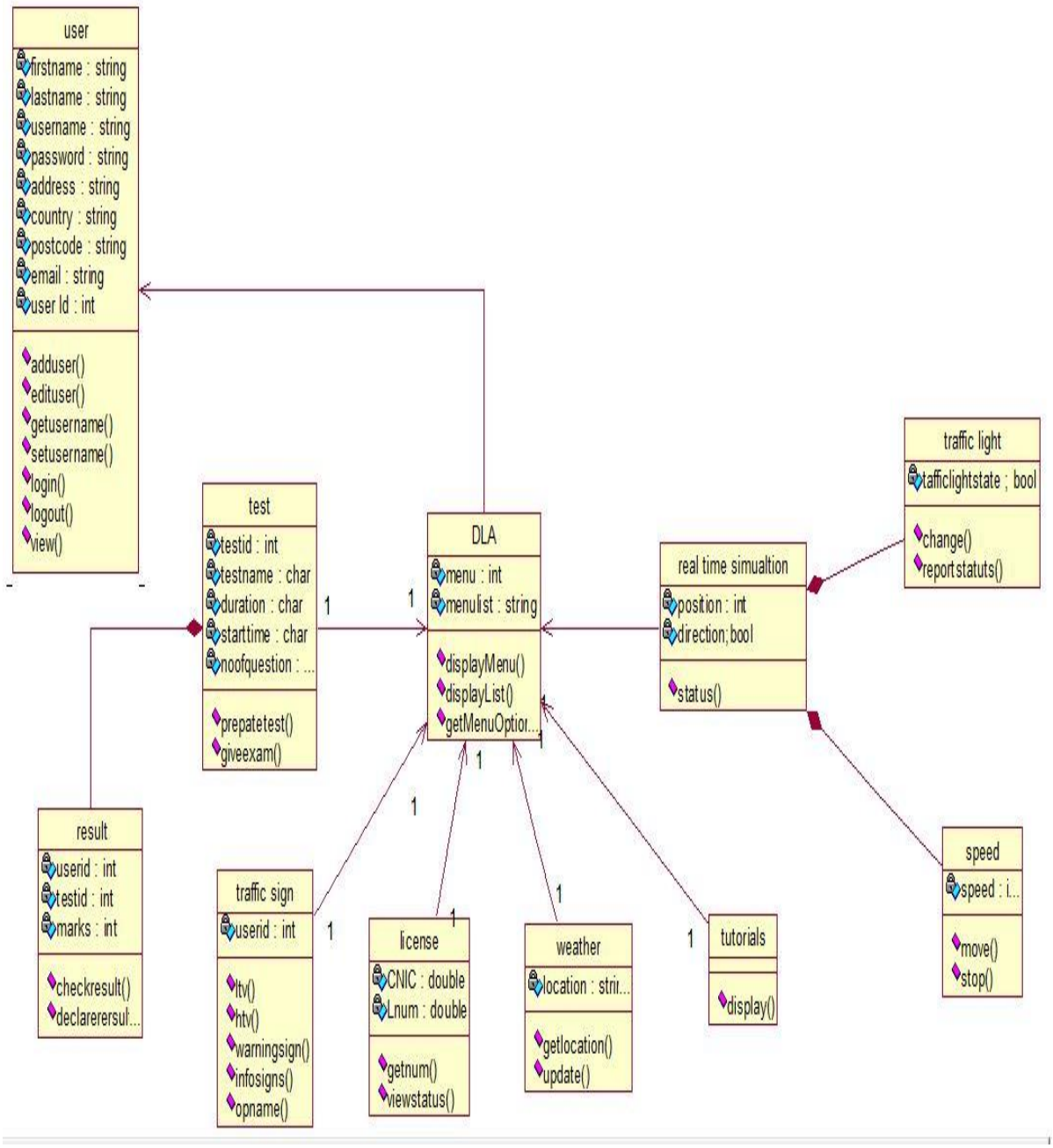


Figure 5: Class diagram

Description

Class name	Description
User	User class contains all the information related to user management. Generalization with other classes of user categorization and the functions that performs all the user management functions.
Menu	It is the main class which will be acting as a gateway to all the other classes
Children guide	This class handles the main guidance of the new drivers.
LTV_HTV	This class handles the type of license.
Administrator	The class handles all the functions that an administrator can perform and maintains the rights of an administrator.
Warning signs	This class informs the user about the warning signs.
Sign In	This class handles the login system for LRAS.
Sign up	This class helps the user to make a new user account.

4.10. User interface issues

4.10.1. User Interfaces

A graphical user interface allows easy access to all features of the system and will be available in all workflow. To make easy learning for user and reduce the complexity, navigation options in every screen will remain same. For exception handling we have used user friendly messages to catch the errors. The user interface should be

- Readable
- Easy to understandable
- Easy to operate

4.10.2. User Characteristics

The user of this system should have the following characteristics:

- User should be familiar with Internet.
- The user should be familiar of all the advanced terminologies related to android app.
- The user must be educated enough to be able to use the app or user should know how to use it.

4.11. Activity diagram

4.11.1. System Activity diagram

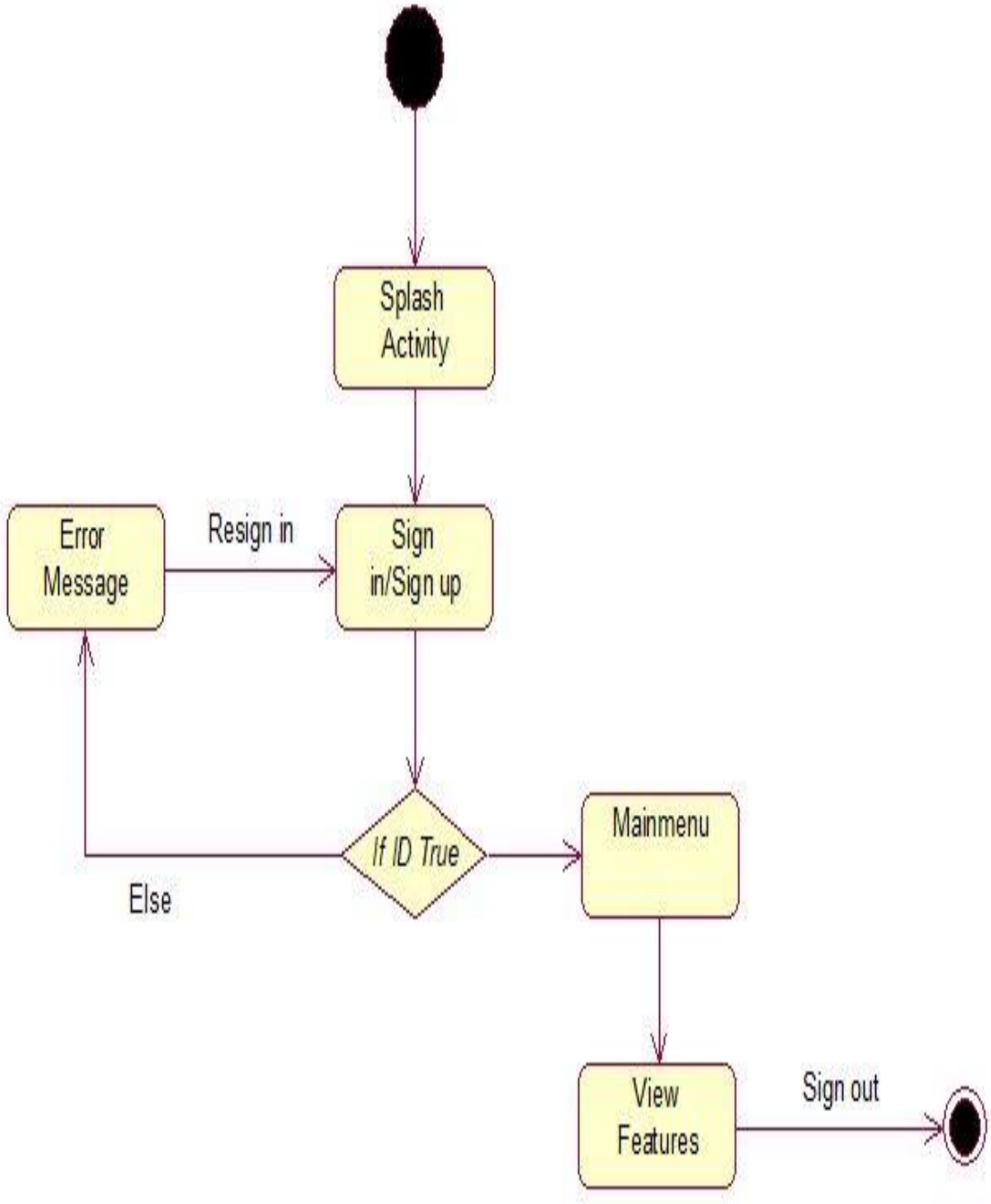


Figure 6: System activity diagram

4.11.2. Activity diagram for registration

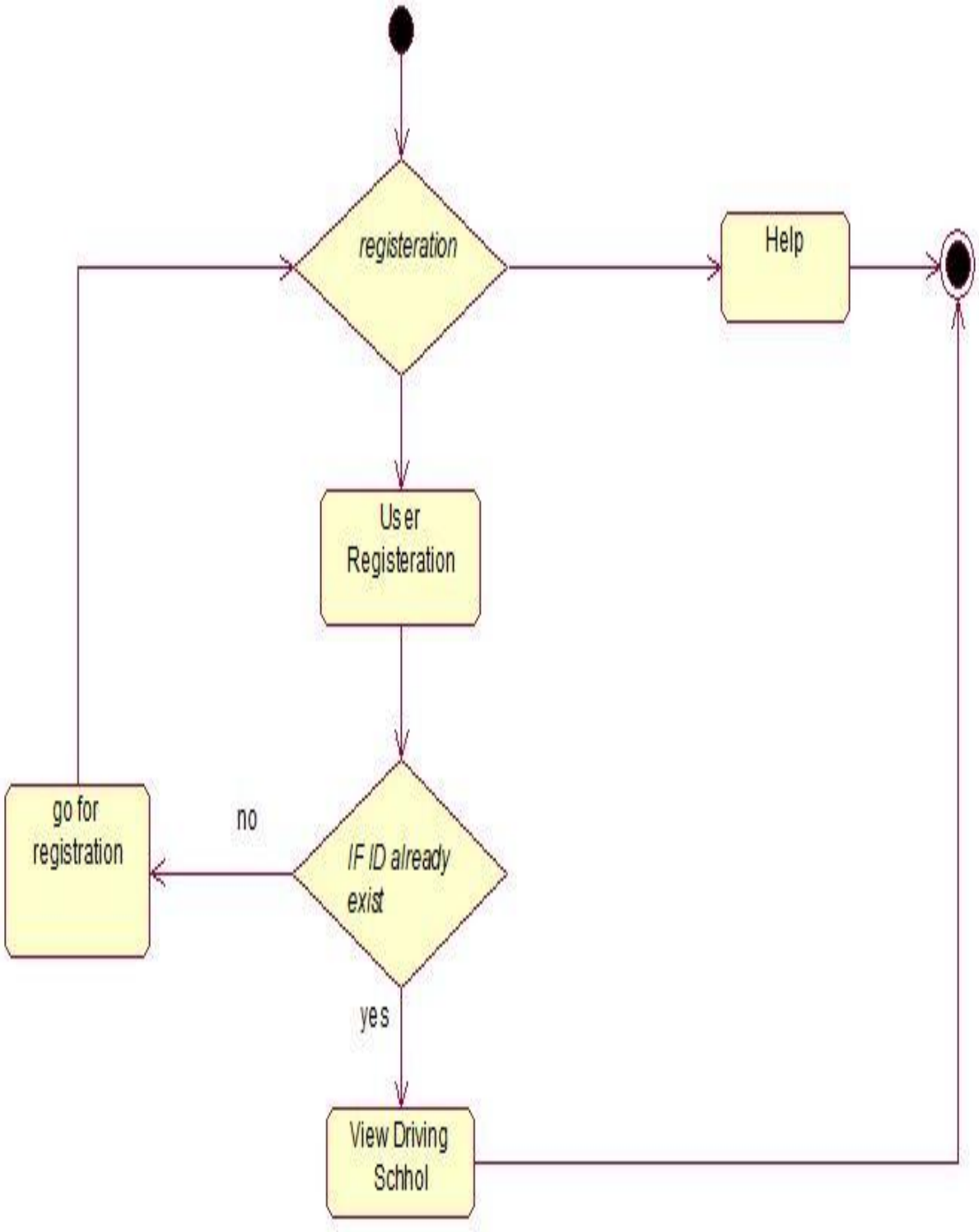


Figure 7: Activity diagram for user registration

4.11.3. Activity diagram for test activity

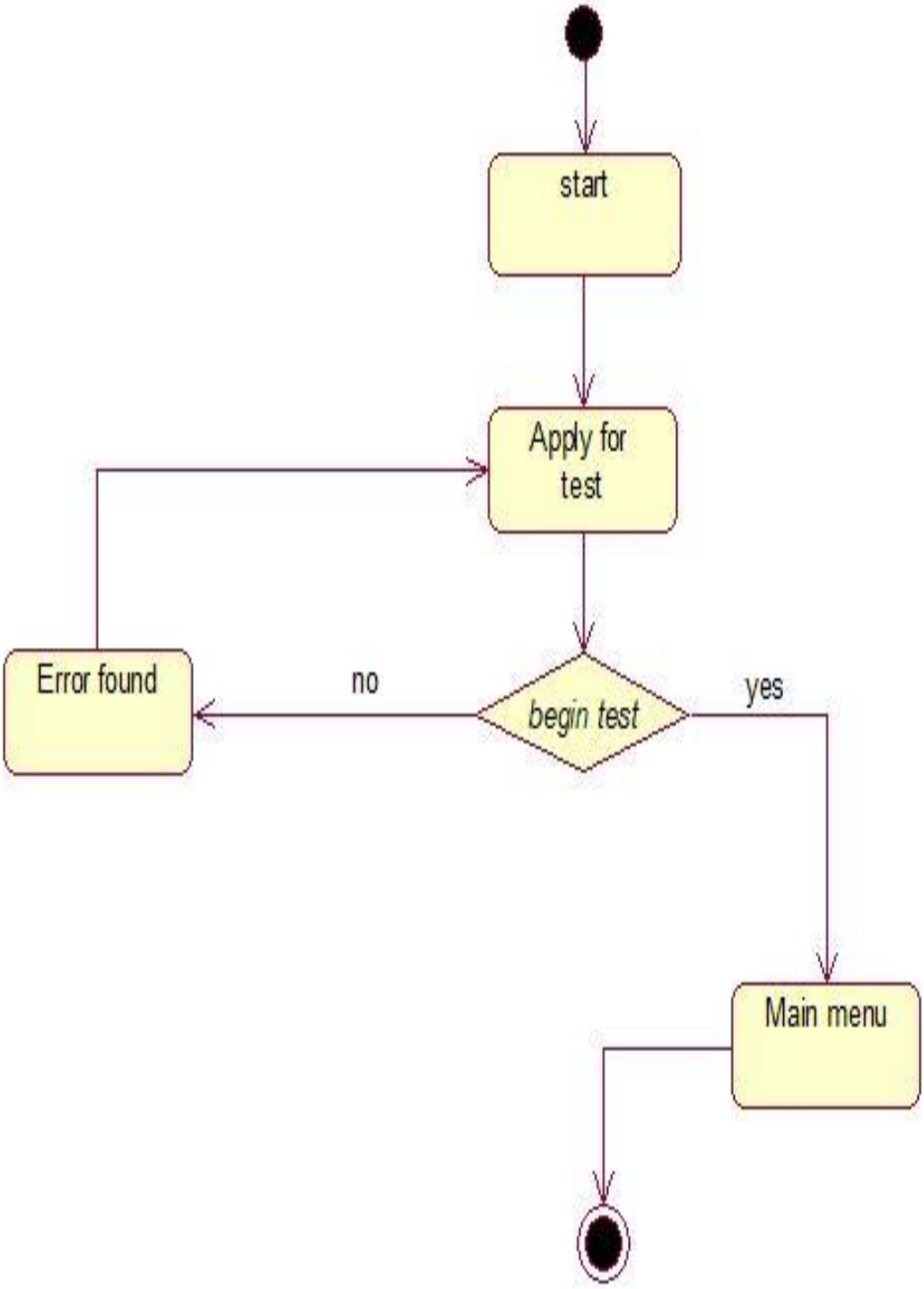


Figure 8: Activity diagram for test activity

4.11.4. Activity diagram for video link

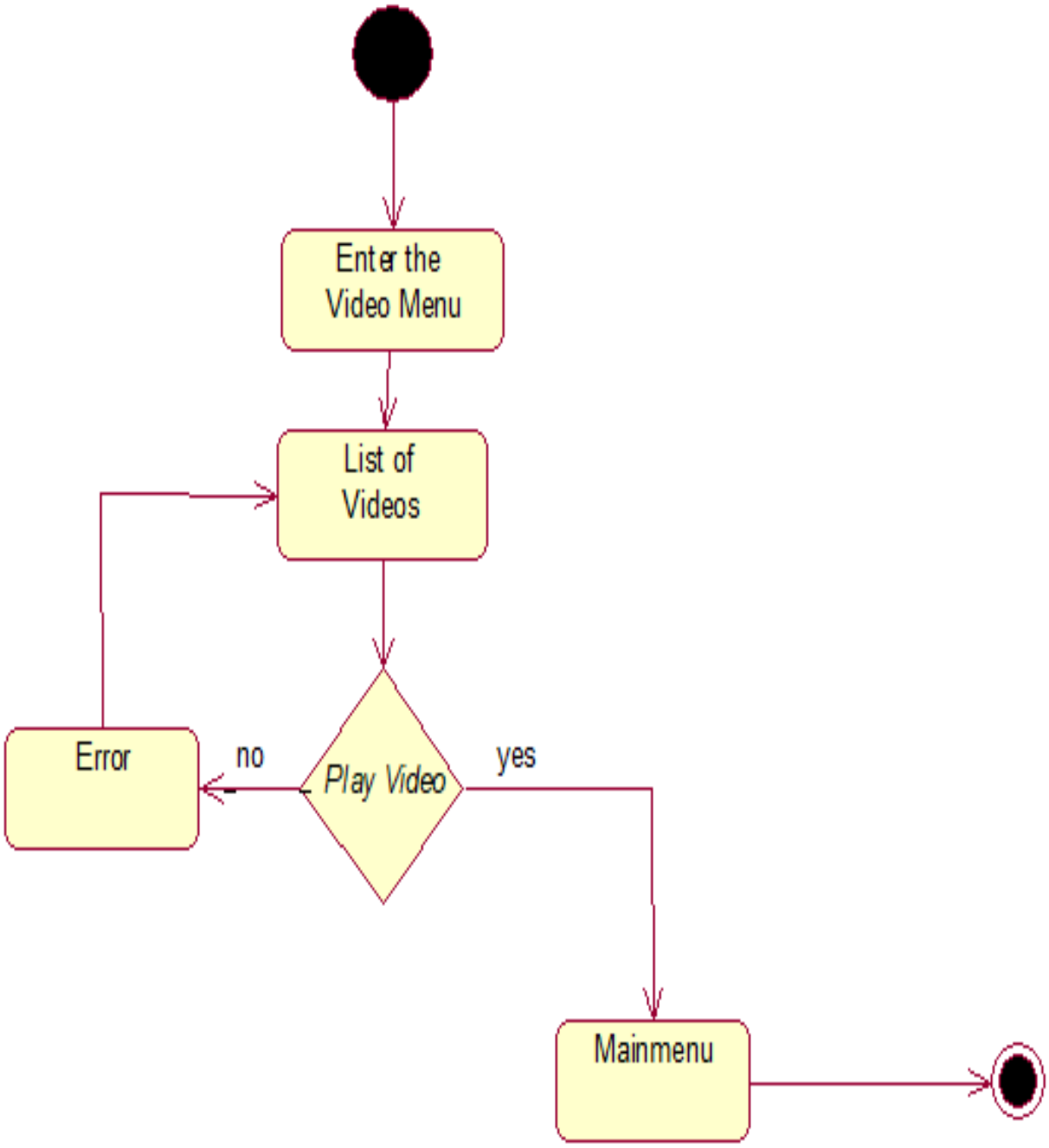


Figure 9: Activity diagram for video link

4.12. Sequence diagram

System Sequence Diagram

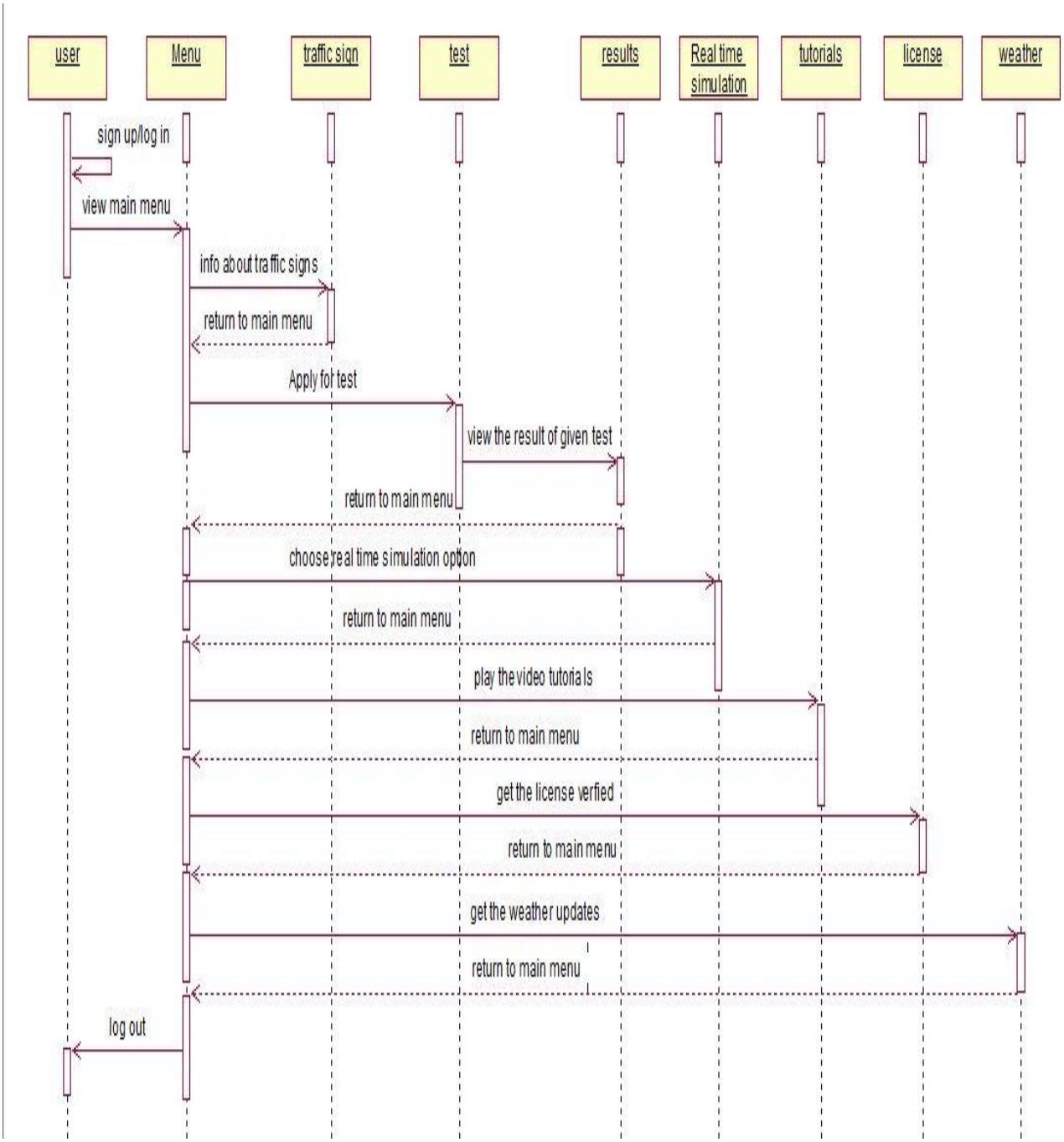


Figure 10: System sequence diagram

Sequence diagram for real time simulation

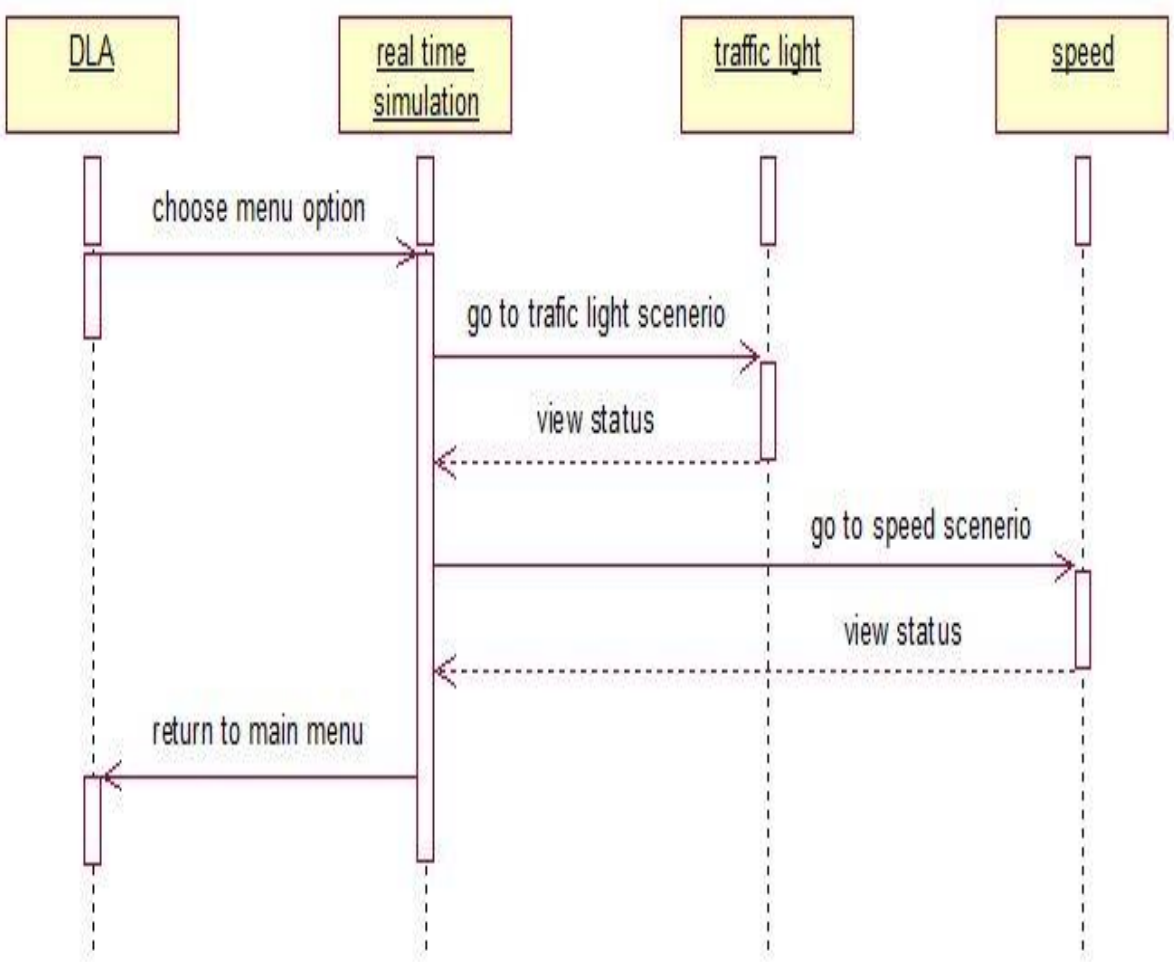


Figure 11: Sequence diagram for real time simulation

4.13. UI Design

The system under development shall sport an intuitive and easy to use UI that will have an extremely shallow learning curve and require minimum training to be operated at maximum efficiency. In this way, users of the android application won't feel reluctant to use the application.

This Driving School app provides a user friendly GUI. Any person having basic knowledge of using mobile phones can use this app easily. All feature of an application are accessible from the main menu.

Mobile application interface

Following are the sketches of a possible UI implementation for the Android Application.

Start up screen/ Main Menu:

This will be the first screen that the user sees on his android device. The user shall be presented with a welcome message.

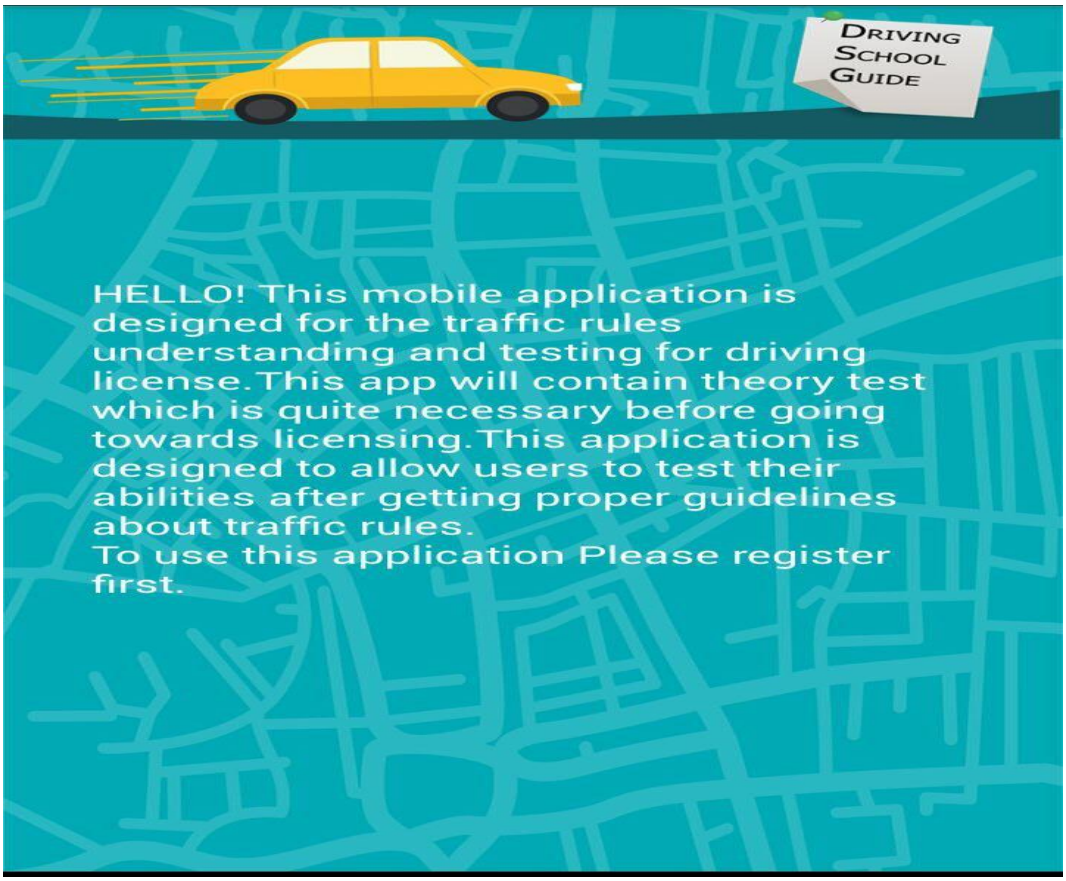


Figure 12: Splash activity interface

The option to test driving abilities or to view the some other updates he/her has to lodged, or to view other details about what the app does, and how to use the application. As visible, the menu is simple and self-explanatory.

Figure 13: Main menu interface

4.14. Detail description of components

User Management

Identification	User Management
Type	Component
Purpose	• To manage a set of Users (registered users, Instant help user). It will manage the registration of Users, their User Credential Database and validation of User Credentials in case of Login activity. • ALL of this functionality is hidden from the user. Complete details of managing the users are the job of the User Management class.
Function	• Represents User management options to the user • Signup(button) • Login(button) • Logout(button)
Subordinates	User Database will be used to hold the user records.
Dependencies	All User management functions depends on it.
Interfaces	• login(user) logs in the user • logout(user) logs out the user • verify(user) verify user credentials from the user database • signup(user)add user to the user database • modify(user)modify user record in the user database
Resources	User Database
Processing	• login() logs in the user • logout() logs out the user • verify() verify user credentials from the user database • signup()add user to the user database • modify()modify user record in the user database
Data	Graphics for interface support. Database repository for maintaining the user record.

Menu Component

Identification	**Menu**
Type	Component
Purpose	• User interface to access different features of the Application • Separate menu class instances for each menu screen
Function	• Display a window of buttons representing different menu choices. • For each menu choice, the menu class will call the function display (button) to display button onscreen.
Subordinates	An array will hold different buttons, one for each option of the menu.
Dependencies	Calls instances of the button class.
Interfaces	On Screen Touch() [calls the button's callback function]
Resources	Nil
Processing	display(button) display the button onscreen
Data	Array of button classes

Button Class

Identification	**Button**
Type	Class
Purpose	• Class to allow user to choose various Menu options • Graphic user interface representation
Function	• Represents various options available to the user • ButtonCallback (button) – a Callback function, used when the button is clicked in order to call the requisite function related to that button.
Subordinates	A graphical picture will be used for the button's graphic.
Dependencies	Depends on the menu class
Interfaces	• button->text() [text overlay on the button] • button->test() [graphic for the button] • button->screenX(), button->screenY() [screen coordinates for the button]
Resources	Nil
Processing	ButtonCallback(button() Callback functions for the button.
Data	A graphical picture to hold the visual graphic of the button.

Graphics

Identification	Graphics
Type	Class
Purpose	• Hold all of the loaded graphical images • Draw real time situations to the graphics display
Function	• Load online test. • Tutorials are loaded stored in the application. • Show emergency contact button to get emergency numbers.
Subordinates	N/A
Dependencies	All components are dependent on this class for their drawing to the screen.
Interfaces	• Next page • Back page • Registration
Resources	Nil
Processing	• Registration () to get registered as a new user. • Onlinetest (name, location) load the online test. • Viewresult (name) load the results. • Updateweather(location) show the current location weather. • Emergency () show the emergency contacts.
Data	Nil

User Access Rights

Identification	User Access Rights
Type	Class
Purpose	• To classify the users into categories on the basis of privileges assigned to each user category. • The privileges contain user rights of interacting the app.
Function	• Categorize(User) • Rights (categorized_user)
Subordinates	User Database will be used to hold the user records.
Dependencies	All User management functions depend on it. • Login() • Onlinetest() • Updates() • ManageAccounts()
Interfaces	None
Resources	User Database
Processing	• Categorize (User) is used to categorize the user into one of the defined categories. • Rights (categorized_user) are used to extend the requisite rights to the user, based on his category.
Data	Database repository for maintaining the user record, based in their individual Access Rights.

Report Generation

Identification	**Report Generation**
Type	Class
Purpose	• To search the records of her/his previous results. • To generate the report/results based on the given test.
Function	• Gets invoked by the result of the information mapping. • search (test_record)
Subordinates	None
Dependencies	Dependent on the output of the search query.
Interfaces	Search (test record, criteria)
Resources	Complaints Database
Processing	Search (test_record, criteria)
Data	Services and user repository.

4.15. Reuse and relationship to other products

To date there had been no such mobile application in Pakistan which allowed users to efficiently get the knowledge of driving and traffic rules.

We have taken “driving stimulators” available in google play store and Apple store (related applications in other countries) as a reference for design and added the functionalities according to our need.

The end user interface is designed using the layout editor which comes with ADT (Android Development Tools). By using the layout editor, designing the interface becomes very easy and the development time is saved as we have to code from scratch.

4.16. DESIGN DECISIONS AND TRADEOFFS

We have kept the user interface simple and friendly so even a user with only basic knowledge of android applications can use it effectively.

We had to eliminate the need for the details of the user to be input who need instant help. The user can enable this feature in any case of emergency to retrieve the information of any emergency contact number easily regarding to road emergencies.

A user without getting registered can get only the emergency feature. For complete use of application he/she has to log to get fully access. New drivers are available with user guides for driving.

Since this application is a new development effort in its category in Pakistan, our focus was to develop an application that is simple enough for an average mobile user to use and get the knowledge about driving and traffic rules.

Chapter 5

5. System Implementation

5.1. Android Development Environment

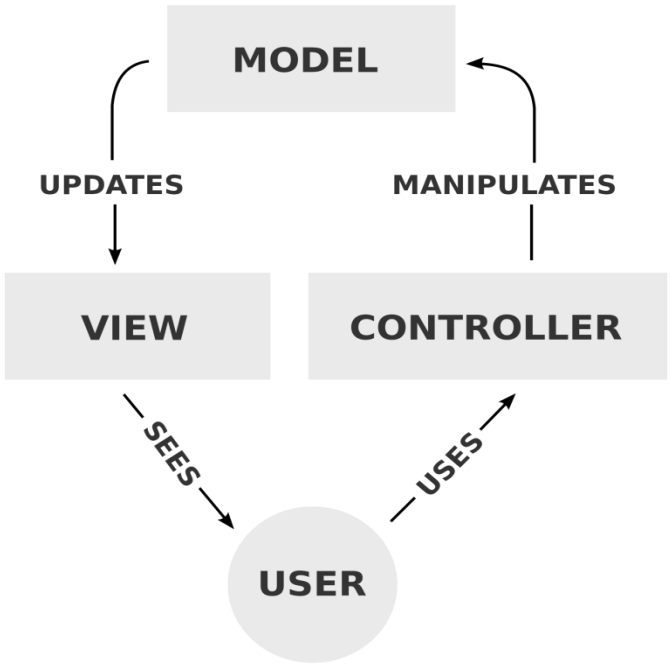
The coding of this android application is done using the java programming language which is an object oriented programming language. We have used high level language to develop our application using a fairly modular approach. Like other software development environment the software development pattern in the android environment is similar to the Model-View-Controller.

5.2. Model View controller pattern

For the software development we use model view controller pattern that separates the modeling of the business entities (domain), user interface (presentation) and the logic of the three separate independent layers. We can test each individual layer independently, if we want to divide up the work among the several developers.

The data of the application domain is represented using this model and it responds to request for information about its state and requests to change its state.

The information is displayed with views on the screen and is comprised of the entire user interface element. The controllers are responsible to respond to user inputs, actions and they instruct the model or the view to change their states properties accordingly.



5.3. Code

Code is referred as a system of rules which are used to convert information such as letter, sound, image, or into another sometimes shortened or secret, form or representation for communication through a channel or storage in a medium.

In the chapter it is mentioned that this app is designed under the Android operating system software using the java development kit (JDK) and the android software development kit (SDK) tools. The main tool is android studio.

5.4. Graphical User Interface (GUI)

This Driving School app provides a user friendly GUI. Any person having basic knowledge of using mobile phones can use this app easily. All feature of an application are accessible from the main menu.

Chapter 6

6. System Testing and Evaluation

6.1. Introduction

Software configuration management (SCM) is a set of tracing and control activities and it starts with the software beginning and terminated with the taken out of software for operation. Software testing is an umbrella activity that is applied throughout the software process. Because change can occur at any time, SCM activities are developed to support the following factors:

- Control change
- Identify change
- Ensure that change is properly being implemented
- Report change to others who may have an interest

The most critical element of the software quality assurance is software testing. It represents the ultimate review of specification, design and coding. The software which is being tested should be reliable and sustainable. As testing is the process of finding errors being made during the development of software. Errors do occur in the software developed by human because humans are imperfect testing demonstrates that the software functions according to the software requirements. It involves executing implementations of the software with test data and examining the outputs and operational behavior top check that is performing as required.

A series of test case that are intended to demolish the software were created and the following objectives were defined the process of testing.

- Correct errors those are uncovered
- Testing in a process of executing program with intent of finding an error.

6.2. Test Items

The individual element to be tested. One test Object can have many Test Items. Based on the requirements of LICENSE DRIVING ANDROID APPLICATION, design description, modules of android application, and non-functional scenario will be tested. The requirements defined in Software Requirements Specification and the design entities as explained in Software Design Document will be tested.

Features to Be Tested

Following features are to be tested:

- Ability to provide instructions for LTV, HTV licenses.
- Awareness about traffic signs
- Having guidelines for Safe & Responsible Driving.
- Playing Video tutorials.
- Ability to verify License through CNIC.
- Provides sample theory test and questionnaire for learners.
- Ability to provide weather updates.
- Real time Situational simulation
    - Awareness of traffic lights.
    - Over Speeding.

6.3. Testing Process

Approach

An independent group of tester performs the functionality of development before it is shipped to the customer as a common practice. This practice often results in the testing phase being used as project buffer to compensate for project delays thereby compromising the time devoted to testing. With the start of project, the software testing process is start and it continues till the end of the project development cycle. Testing process includes the following factors:

- Exercising of computer program with predetermined inputs
- Comparing the actual results against the expected results
- Cannot absolutely prove absence of defects

Extreme programming and the agile software development movement are the emerging software discipline; adhere to a “test-driven software development” model. First we have to write for tests, often with pair programming in the extreme programming methodology. It may happen that these tests fail initially; as they are expected to. As code is written then it passes gradually with increments the larger portions of the tests suites. There is a need to update the test suits, because new failure conditions and corner cases are discovered. These conditions are then integrated with any regression tests that are developed Unit tests are maintained along with the rest of the software

source code and generally integrated into the build process, for the interactive test which is being relegated to a partially manual build acceptance process.

6.4. Testing levels

Our test plan for this project will test the following items of the project.

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing
- Regression Testing
- White Box Testing
- Black Box Testing

Main Constraints

- Limited resources
- Time constraints
- Expensive testing software

Unit Testing

Basically in computer science technology, unit testing is a software design and development method where the programmer verifies that individual module of source code is working properly. It is done on small units; unit is the smallest testable part of an application it may belong to a base/super class, abstract class or derived / child class. Ideally, each test case is independent from the other.

It is taken as a type of white-box testing but sometimes treated as black-box. Usually the developers perform this testing and unit tests are written in code in the same language as the module. It is basically done during development when an individual module is completed.

Integration Testing

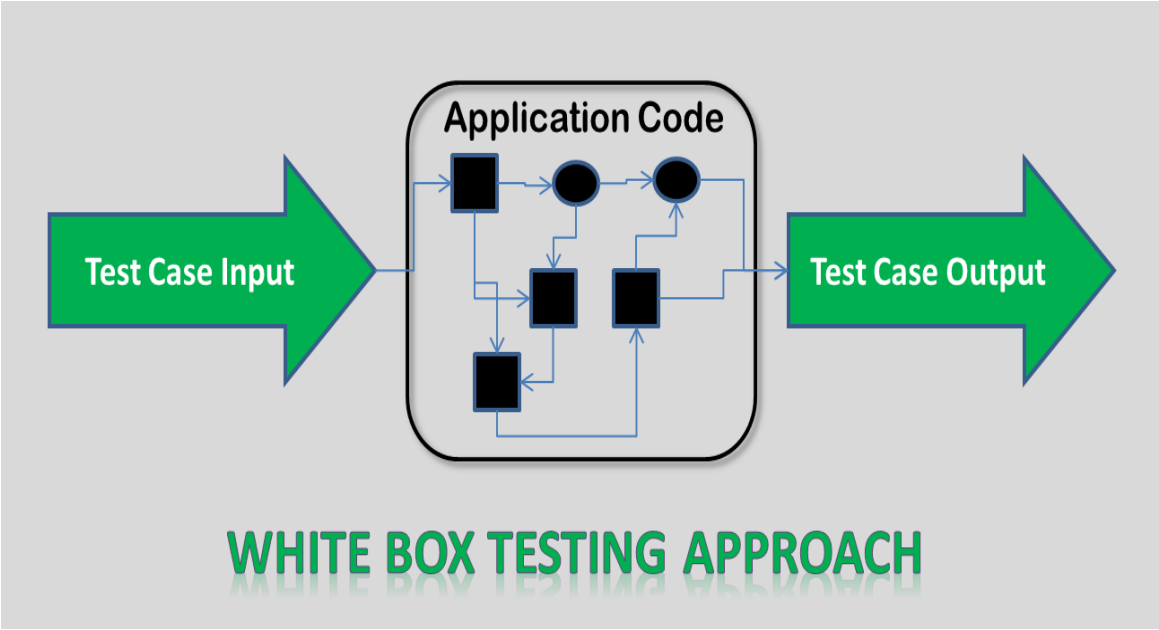
Integration testing provides interfaces between components and it is the first step after unit testing. It is done for the components because components may work alone but fail when put together. It is possible that defect may exist in one module but manifest in another. Integration testing exposes defects in the interface interaction between the integrated components. Progressively larger groups of tested software components corresponding to elements of architectural design are integrated and tested until the software works as a system.

System Testing

It involves the testing of the complete system and refers to a type of black-box testing. It is done for the system to be accepted for operational use. On the entire system level testing is done to ensure that all the desired requirements and specifications are fulfilled.

White box Testing

White box testing (also known as clear box testing, open box testing, glass box testing, transparent box testing, code-based testing or structural testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

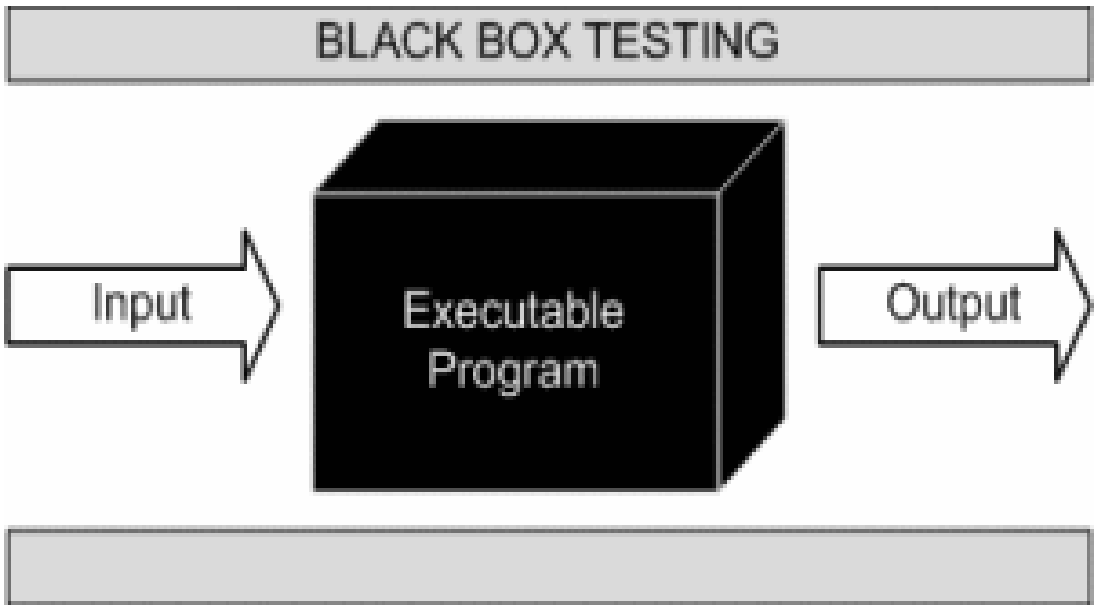


Black Box testing

Black box testing, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors



User Acceptance Testing

User acceptance testing is the last milestone in testing phase. It is ultimate customer test & sign-off. Sometimes user acceptance testing synonymous with beta testing and customers are satisfied that software meets their requirements at the completion of this phase. The results are based on “acceptance criteria”. Acceptance criteria is consists of the conditions the software must meet for customer to accept the system. Ideally it is defined before contract is signed between the customer and the developer.

Acceptance testing

This process was carried out for the users in normal work environment, to test and making sure the software works correctly.

- **Alpha test**

  The complete software is tested under the supervision of the developer at the developer's site and this test was done for the complete version.

- **Beta test**

  The complete software is tested by supervisor at his own site without developer being present and this was also done for the complete version of the system.

Major objective for acceptance testing

The main objectives of the acceptance test were as follows:

- To verify that the complete system meets the agreed upon criteria
- To identify that is there are any discrepancies, and to solve them
- To establish the readiness of the system for cut-over to live operations.

Regression Testing

It is a testing phase of re-running the tests after fixes or changes are made to software or the environment i.e. Quality Assurance finds defect, developer fixes, and Quality Assurance runs regression test to verify.

The tests given below were focused tests during the above mentioned testing levels

a) Input testing
b) Output testing
c) Code testing

a. Input testing

To insure the best performance of the software an input is given to the proposed system to check that wrong output on incorrect input.

b. Output testing

Output testing process is carried out to test the outputs generated by the software and to specify that either the software is providing the required results or not.

c. Code testing

For the testing of code this process is carried out. It is done to specify that either the coding is error free or not and check that the regarding software satisfy the

performance, provides a quick flow of data and code design matches the required functionality.

All the components were focused individually for test at initial levels of testing. Each function of the system was treated as unit. Next component were integrated to form a complete software package and applied integration testing. Integration testing addressed the issue associated with the verification and program construction. On the whole system testing was carried out for the verification of all elements mesh properly and the overall desired system performance is achieved at final levels.

6.5. Item Pass /Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

Preconditions are met

- Inputs are carried out as specified.
- The result works as what specified in output => Pass
- The system doesn't work or not the same as output specification => Fail.

6.6. Environmental needs

Hardware

- ANDROID PHONE

Software

- Android Studio
- XML Language
- JAVA for coding

6.7. Testing Tasks

- Develop test cases.
- Execute tests based on the developed test cases for the app.
- Report defects from the executed test cases if any.
- Provide complete test report.
- Incorporate or manage changes later in the stage of the project development.

6.8. Test Deliverables

User registration

Test Case Number	01
Test Case Name	User registration
Created by	Developer and Tester
Description	A user tries to sign up in to the system.
Testing Technique	Black Box Testing
Preconditions	User has to open the sign up page first.
Test Steps	• The user enters the membership details on the screen and clicks the sign up button. • The system checks for the availability of the username. • The system generates the membership ID. • The system records the membership information of the new member, in the database.
Test Data	• The User provides his personal information for registration.
Expected output	User got registered in the app.
Actual output	User got registered in the app.
Status	Test case passed successfully.

Table 10: Test Case 1(User registration)

Figure 14: Test case 1(User registration)

View main menu

Test Case Number	02
Test Case Name	View Main menu
Created by	Developer and Tester
Description	A user tries to view the main menu of the system.
Testing Technique	Black Box Testing
Preconditions	User has to open the sign up page first.
Test Steps	• The member enters the membership details on the screen and clicks the sign up button to view the menu. • The system checks for the availability of the credentials to view the main menu. • The system displays an error report if the main menu is not available.
Test Data	• N/A
Expected output	Application show the main menu after login.
Actual output	Application show the main menu after login.
Status	Test case passed successfully.

Table 11: Test Case 2 (View Main menu)

Figure 15: Test case 2(View main menu)

Apply for test

Test Case Number	03
Test Case Name	Apply for test
Created by	Developer and Tester
Description	A user has to apply for driving license test.
Testing Technique	Black Box Testing
Preconditions	User has to login first
Test Steps	• The user enters the main menu. • Then user applies for license driving test. • The system generates the digital questionnaire to the user. • The system records the user's solution.
Test Data	.
Expected output	User will apply for driving license test.
Actual output	User will apply for driving license test.
Status	Test case passed successfully.

Table 12: Test Case 3(Apply for test)

View Results

Test Case Number	04
Test Case Name	View results
Created by	Developer and Tester
Description	A user has complete his/her license test and now he wants to view the result.
Testing Technique	Black Box Testing
Preconditions	The user must have applied for test first.
Test Steps	• The user enters the main menu. • The user enters the view results options. • The system checks for the availability of the result. • The system generates the result of applied test.
Test Data	.The user must have applied for test and the checks for his/her results.
Expected output	The system output the result for the given test by the user.
Actual output	The system output the result for the given test by the user.
Status	Test case passed successfully.

Table 13: Test Case 4(View Result)

Weather update

Test Case Number	05
Test Case Name	Weather Update
Created by	Developer and Tester
Description	A user has to get weather update.
Testing Technique	Black Box Testing
Preconditions	The user has to login first.
Test Steps	• The user enters the main menu • The user selects the weather update option. • The user gives the desired location. • The system checks for the availability of the desired location. • The system displays an error report if the desired location is not available.
Test Data	.The user must have provided the desired location.
Expected output	The system output the weather update for user's desired location.
Actual output	The system output the weather update for user's desired location.
Status	Test case passed successfully.

Table 14: Test Case 5(Weather Update)

Figure 16: Test case 5(Weather update)

License verification

Test Case Number	06
Test Case Name	License verification
Created by	Developer and Tester
Description	A user has to verify his license through his/her CNIC.
Testing Technique	Black Box Testing
Preconditions	• User must be having his/her license. • User must be having his CNIC. • User must have registered the application.
Test Steps	• The user enters the main menu. • The user selects the license verification option. • The user provides his/her CNIC number. • The system checks for the availability of the license. • The system displays an error report if the license is not available.
Test Data	.The user must have provided the CNIC & license number.
Expected output	The system output the verification of LTV_HTV license.
Actual output	The system reports an error (unverified license).
Status	Test case passed successfully.

Table 15: Test Case 6 (License Verification)

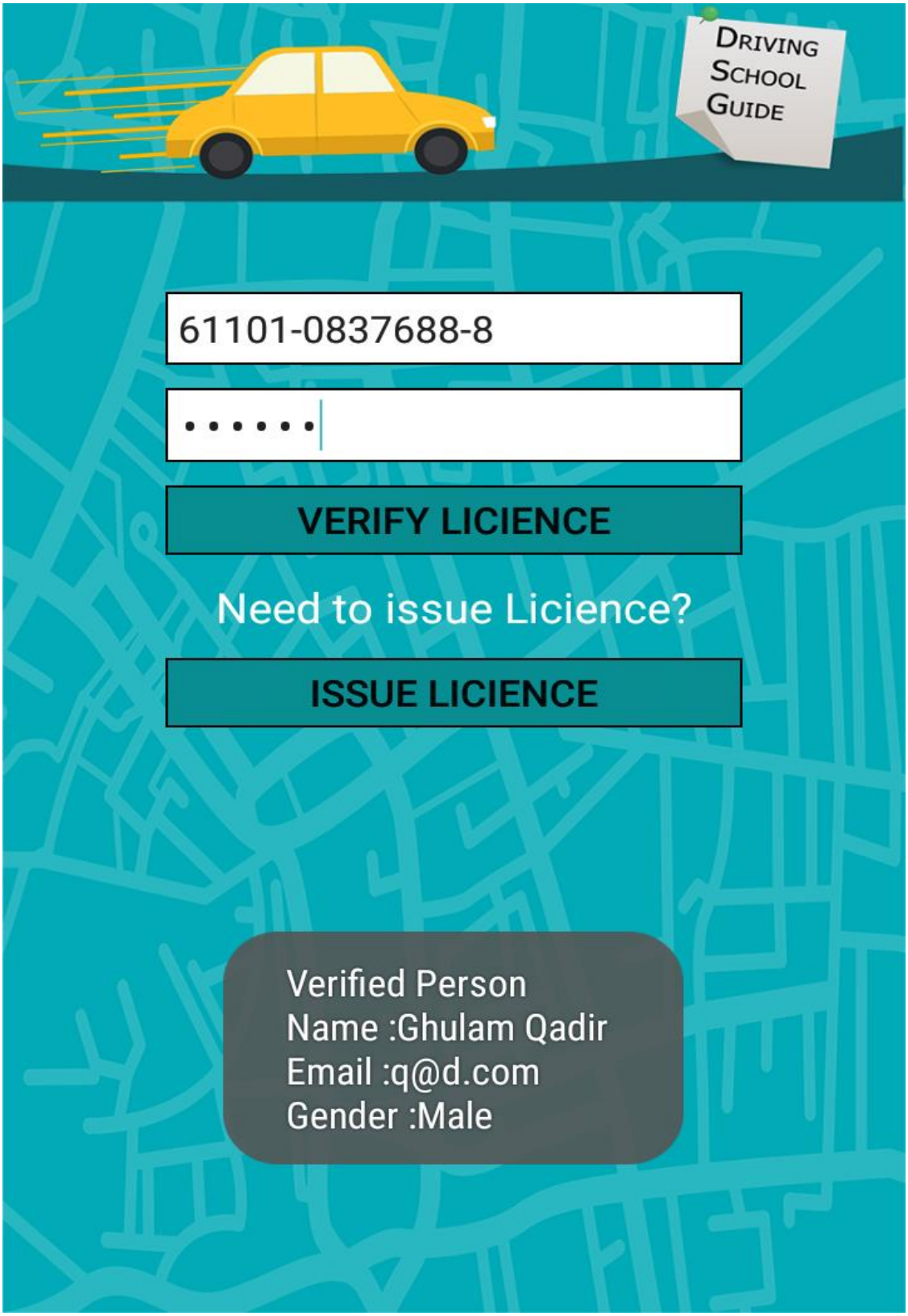


Figure 17: Test case 6a (Verified license)

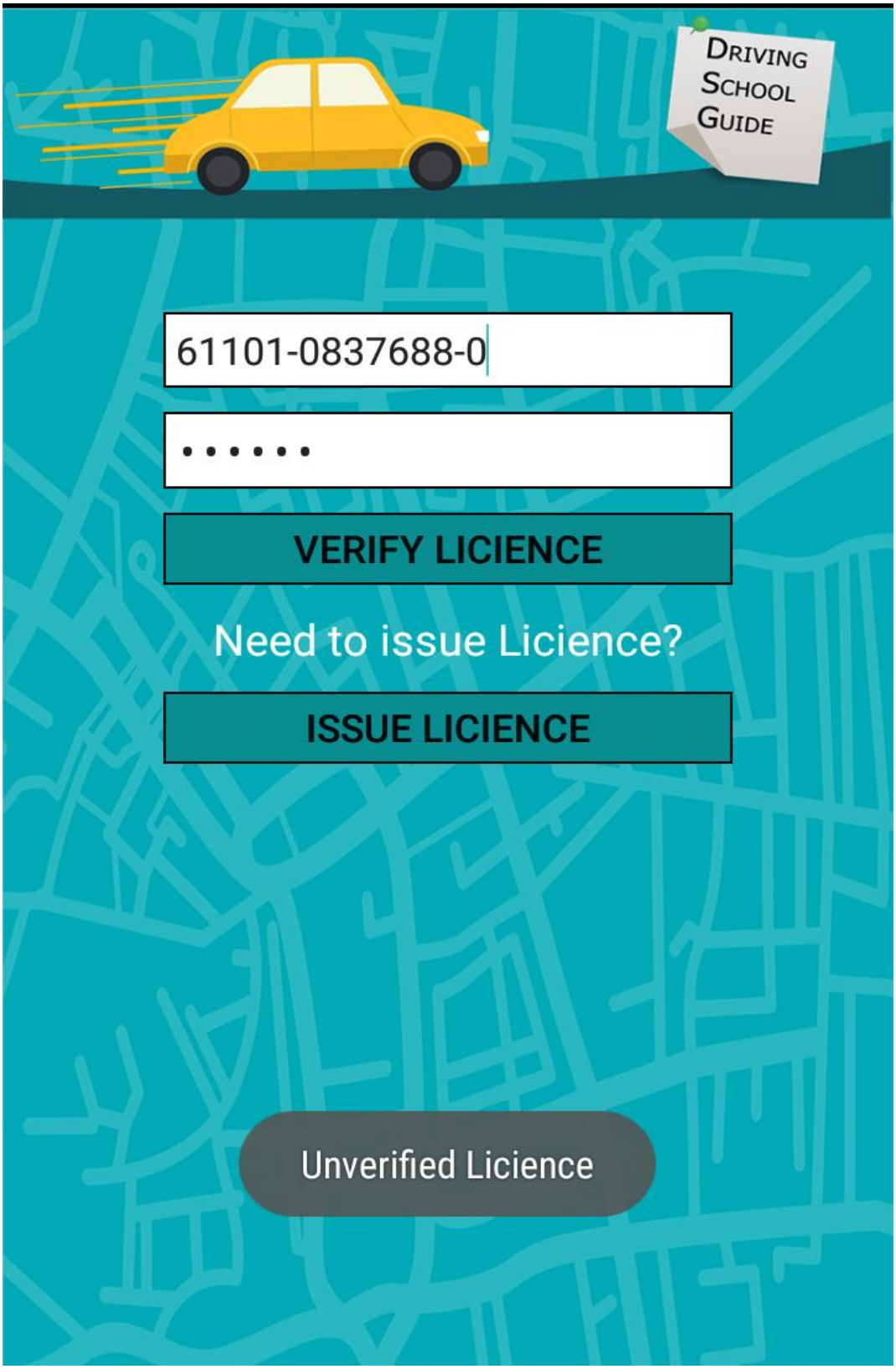


Figure 18: Test case 6b (Unverified license)

Tutorials

Test Case Number	07
Test Case Name	Tutorials
Created by	Developer and Tester
Description	A user will be able to see the video tutorials of different driving situations.
Testing Technique	Black Box Testing
Preconditions	• User has to sign in first.
Test Steps	• The user enters the main menu. • The user selects the tutorials option. • The list of videos of different situation will be available to the user. • The system plays the selective video to the user. • The system displays an error report if video is not found.
Test Data	.N/A
Expected output	The system view the tutorials to the user.
Actual output	The system view the tutorials to the user.
Status	Test case passed successfully.

Table 16: Test Case 7(Tutorials)

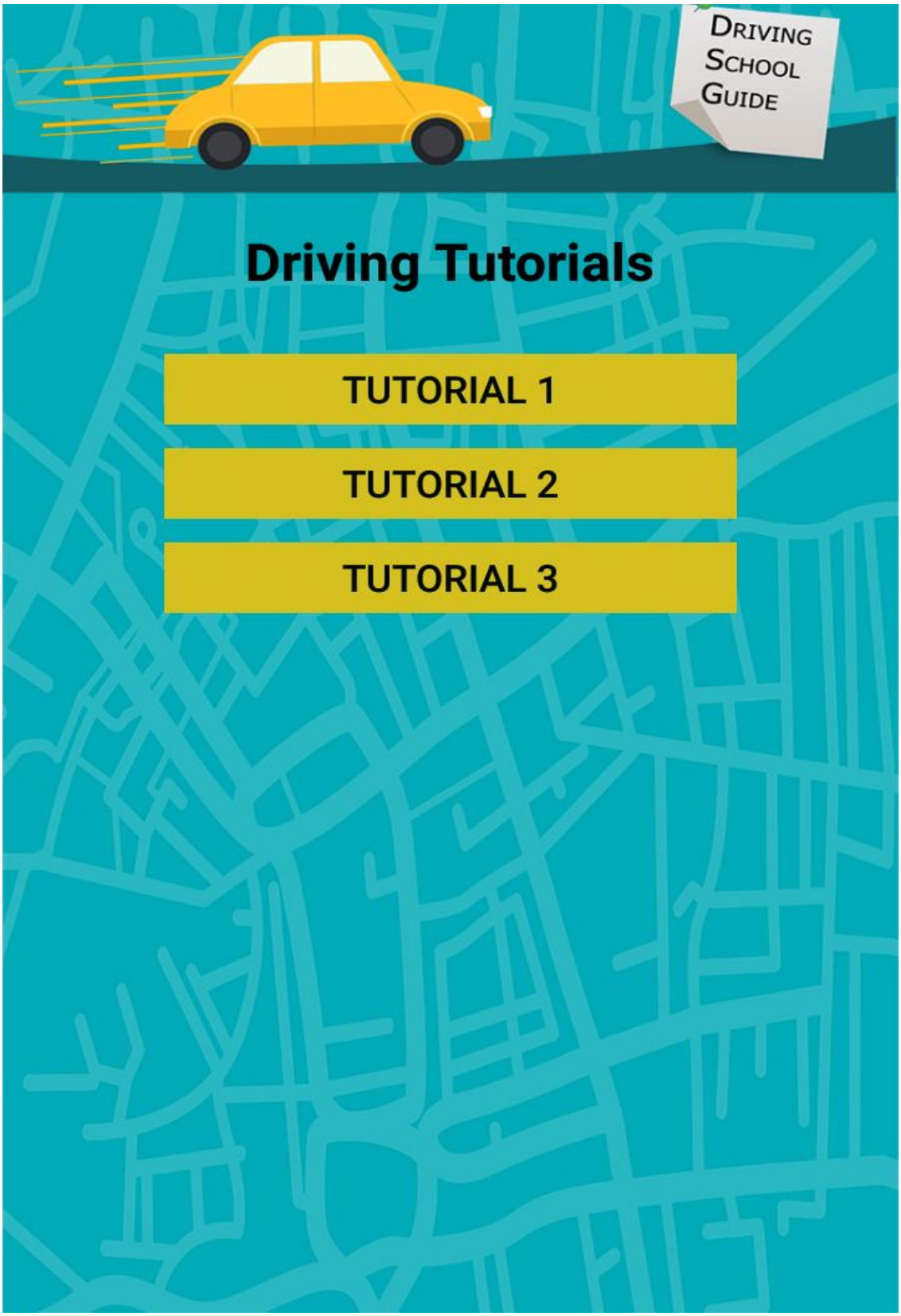
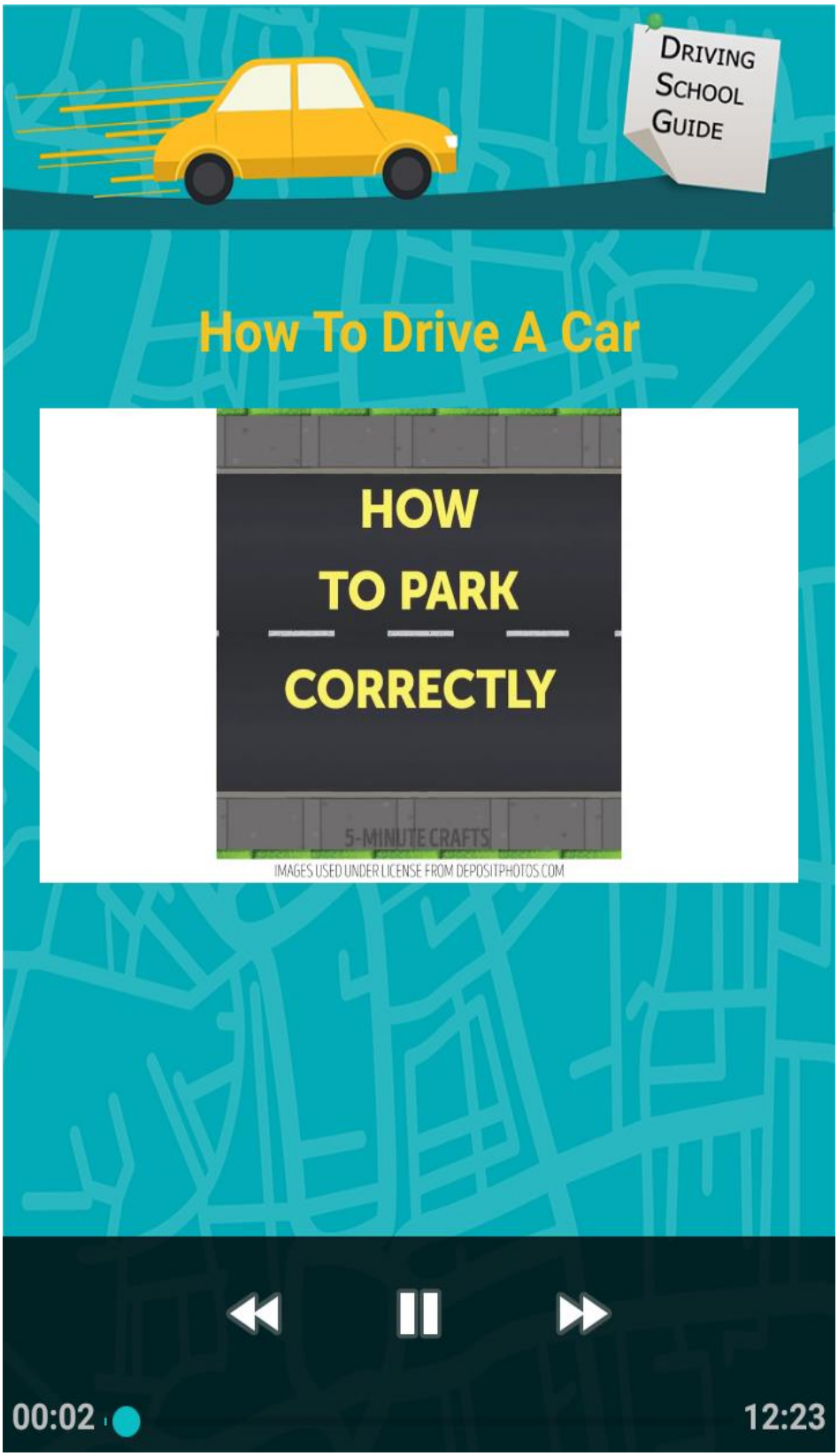
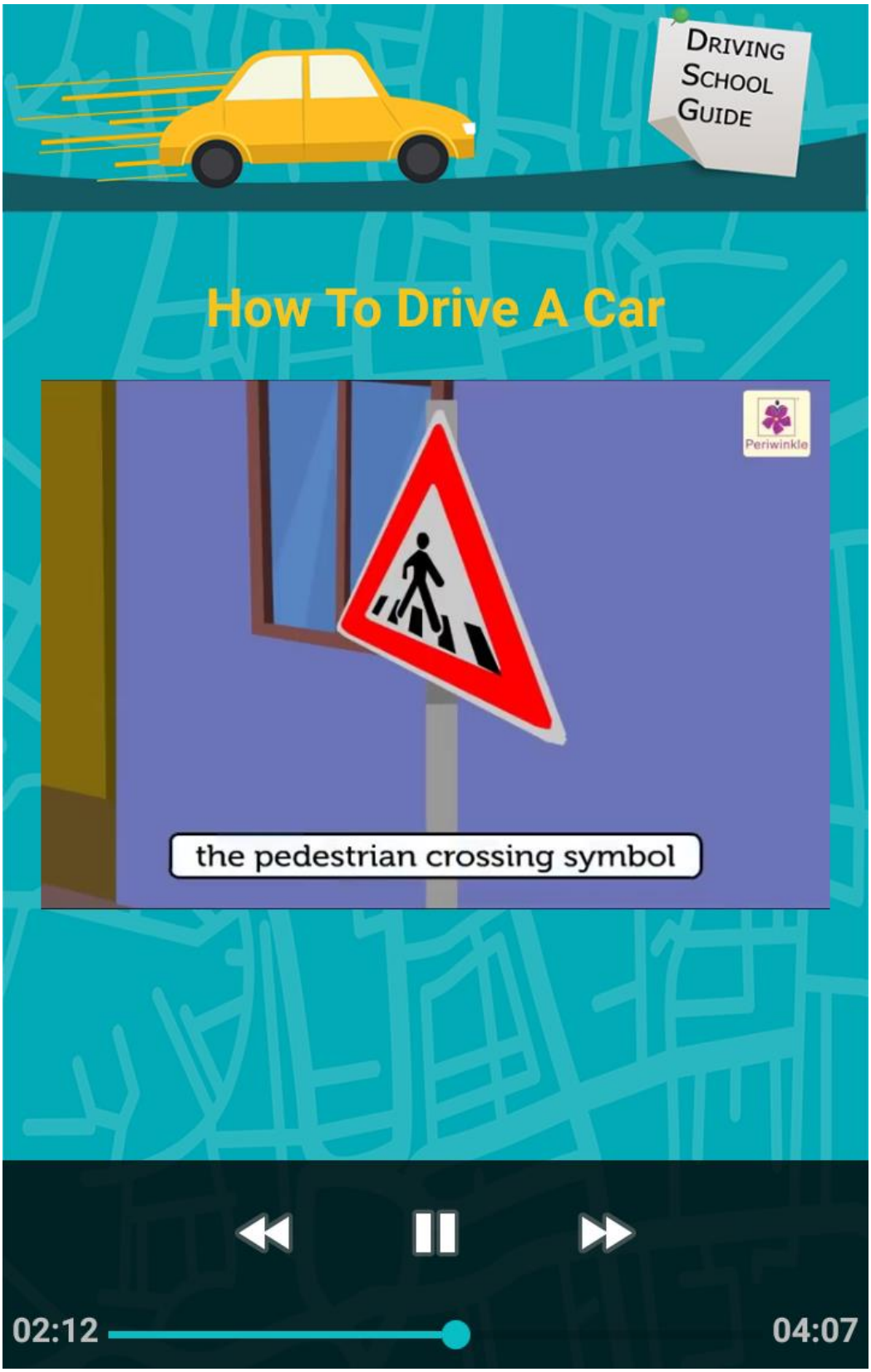


Figure 19: Test case 7(Tutorials)


DRIVING SCHOOL GUIDE
How To Drive A Car
HOW TO PARK CORRECTLY
5-MINUTE CRAFTS
IMAGES USED UNDER LICENSE FROM DEPOSITPHOTOS.COM
00:02
12:23

How To Drive A Car


DRIVING
SCHOOL
GUIDE
How To Drive A Car
Periwinkle
the pedestrian crossing symbol
02:12
04:07

Real time Simulation

Test Case Number	08
Test Case Name	Real Time Simulation
Created by	Developer and Tester
Description	A user will be able to check the speed of simulation.
Testing Technique	Black Box Testing
Preconditions	• User has to sign in first.
Test Steps	• The user enters the main menu. • The user selects the Simulation option. • The system shows a car simulation to the user. • The system displays an error report if video is not found.
Test Data	.N/A
Expected output	The system will warn the user about over speed. (If speed goes above 60km/h)
Actual output	The system warns the user about over speed. (speed goes above 60km/h)
Status	Test case passed successfully.

Table 17: Test Case 8 (Real time Simulation)

Figure 20: Test case 8a (Traffic lights)

Figure 21: Test case 8b (Over speeding)

Chapter 7

7. Conclusion and Future Work

7.1. Conclusion

For a system development a developer should keep in mind the main factors of robustness, reliability and error free system while in the process of development of software. After the learning of four year theory, the development of this application was really steeping into practical workplace. This application is developed as driving school guide over Android studio.

Our project covers the basic functionalities of driving school. It provides awareness about traffic signs, list of famous driving schools in different cities, instructions for driving license, instruction for children about road safety, theory test preparation and platform for taking a theory test. During the project development, we used to integrate current innovative technologies for the solutions regarding our project problems. Our project named **Driving School Guide** was completed successfully.

Our focus was on the error free product while developing this app. We ensured to make this app error free in less time with great care to get efficiency. Our purpose of developing this product was to give a platform to the users to enhance their knowledge about road safety and for the theory test preparation for licensing.

While designing and in the whole development cycle of this app we had get much practical knowledge about designing, coding for android application, usage of templates, and management of databases. We had learnt about the different testing techniques, and practically test all the features of the product individually. We had tried to make this product secure although there is no security issue. We are much satisfied with the design of this product and it can be used by many users on real life bases when it is available in play store with some modifications if required.

7.2. Future Work

Our project can be enhanced to advance level for traffic police. There is a scope for further development to take proper test and issuance of license for driving from different institutions.

Different feature regarding traffic and driving can be added more in future. In short an online system for licensing can be given to users with simulation, first for learning and then to check the driving

abilities in simulation. Future work involves the addition of more categories which will provide further ease in learning about driving and testing. We are working on another module of this app to add a feature of road maps to check the road conditions for the assistance of drivers to follow the track with less traffic. For this there we will use cameras to capture the road conditions in different areas and those pictures would be shown according to the requested area of the user. There will be another feature providing with simulation activities for the learners to follow the road signs while driving, in which a user would drive in simulation and will test himself for the rules regarding driving.

8. Appendix A: Glossary

Android	An operating system designed for mobile devices (i.e. cell phones, tablet computers) by Google, Inc.
App	Application
API	Application Programming Interface
Black box Testing	Testing emphasizes on the external behavior of the software entity.
CNIC	Computerized National Identity Card
MCS	Military College of Signals
GUI	Graphical User Interface
MVC	Model View Controller
GPS	Global Positioning System
FYP	Final Year Project
JAVA	Just another Vulnerability Announcement
LTV	Light traveling vehicles.
HTV	Heavy traveling vehicles
NUST	National University of Science and Technology
N/A	Not Applicable
OS	Operating System
SRS	Software Requirements Specification
White Box Testing	Testing emphasizes on the internal behavior of the software entity
pdf	portable document format
24/7	All the time(24 hours 7 days)

9. Bibliography

- https://www.google.com.pk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwje44Tg8LrXAhXnIsAKHXWqAdsQFggpMAE&url=https%3A%2F%2Fweb.cs.dal.ca%2F~hawkey%2F3130%2Fsrs_template-ieee.doc&usg=AOvVaw3m6N2fEEPiLkM7-nflitlv
- The Unified Modeling Language Reference Manual. James Rumbaugh, Ivar Jacobson, AND Grady Booch. 1998. P.81. **ISBN:** 0-201-30998-X.
- The Unified Modeling Language Reference Manual. James Rumbaugh, Ivar Jacobson, AND Grady Booch. 1998. P.87. **ISBN**: 0-201-30998-X.