# **DECLARATION**

The work reported in this thesis was carried out by me under the supervision of ­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Department of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ GC University, Faisalabad, Sahiwal Campus, Pakistan.

I hereby declare that the title of Project \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and the contents of thesis are the product of my own research and no part has been copied from any published source (except the references, standard mathematical or genetic models /equations /formulas /protocols etc.). I further declare that this work has not been submitted for award of any other degree /diploma. The University may take action if the information provided is found inaccurate at any stage.

**Student Name & Signature**

**Waqas Saleem**

**2015-GCUF-77121**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **CERTIFICATE**

This is to certify that **Waqas Saleem** Roll No. **2015-GCUF-77121,** has successfully completed the final project named as: **ENTREPRENEUR DIRECTORY** , at the Department of Computer Science, Government College University, Faisalabad, to fulfill the partial requirement of the degree of **BS(SE).**

**Internal Examiner**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**External Examiner**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Chairperson**

Department of Computer Science

Government College University Faisalabad

# **ACKNOWLEDGEMENT**

We thank the management of GCUF, our faculty and the staff of the Computer Sciences Department, for motivating us to think beyond books and imbibing in us the attitude and aptitude to implement what we have studied so far to make us aware with some of the real-world challenges. We would like to convey our heartfelt gratitude to our project guide “**Mr.Zaid** “ for inspiring us to take up this project. His valuable guidance and timely support, without which, we would have never been able to complete the project, cannot be forgotten.

Finally, we would like to express our gratitude for the faculty members of Computer Sciences Department of GCUF and thank them for their cooperation and timely assistance. Finally, we would like to thank our parents for their moral support.

Waqas Saleem

# **DEDICATION**

This project is dedicated to our loving parents, siblings and our beloved teachers who give us their full morale support to work hard day and night

# **Table of Contents**

DECLARATION.. i

CERTIFICATE.. ii

ACKNOWLEDGEMENT. iii

DEDICATION.. iv

Table of Contents. 5

1. Introduction*:* 11

1.1 Purpose: 11

Faster, smarter, better…wifi toy car goes where no app-controlled car has gone before. 11

Offline Entertaining device. 11

1.2 History: 11

1.3 Clear Statement of the problem: 12

1.4 Actors Catalogue: 12

1.6 Objective: 12

1.7 Tools and Technology: 12

Software Tools: 12

Android Studio. 12

1. Car chases kit 13

2.NodeMCU.. 14

3.DC Motor controller 15

4.Bread Board. 16

5. Batteries. 17

5. V380 camera. 18

2. STATE OF THE ART: 20

2.1 Related Work: 20

2.2 Limitations of System: 20

2.2.1 offline wifi Availability. 20

2.2.2 Must be a educated. 20

2.3 The Necessity: 20

2.3.1 Camera Confidentiality. 20

2.3.2 Accurate videos. 20

2.3.3 Better Communication. 20

3. METHODOLOGY & WORKPLAN.. 22

3.1 Proposed Application Architecture. 22

3.2 Existing Methodology: 22

3.2.1 Waterfall Model 22

3.2.2 Incremental Model 23

3.2.3 Agile Model 24

3.3 Adopted Methodology: 25

3.4 Roles & Responsibilities: 26

4. SYSTEM ANALYSIS & DESIGN.. 28

4.1 Requirement Analysis*:* 28

4.1.1 Functional Requirements. 29

Login as a User by camera to app: 29

4.1.2 Non-Functional Requirements: 29

Availability: 29

Maintainability: 29

Performance: 29

Security: 30

Usability: 30

Portability: 30

Supportability: 30

4.2 Use Case: 30

4.3 Activity Diagrams: 38

4.4 Sequence Diagrams: 40

Sequence diagram of User/Admin/Cashier Login: 41

Sequence diagram of Admin Dashboard. 42

Sequence diagram of User Dashboard: 43

Sequence diagram of Warden Dashboard: 44

Sequence diagram of Cashier Dashboard: 45

4.5 Deployment Diagram.. 46

4.6 Class Diagram.. 47

5. SYSTEM IMPLEMENTATION.. 49

5.1 Introduction. 49

5.2 Screenshots: 49

5.2.1 Android Screenshots: 49

6. SYSTEM TESTING.. 51

6.1 Introduction. 51

6.2 Testing Plan. 51

6.3 Unit Testing. 51

6.4 System Testing. 52

6.5 Integration Testing. 52

6.6 User Acceptance Testing. 52

6.7 Test Cases. 52

6.8 Testing Results. 53

7. CONCLUSION & FUTURE WORK.. 56

7.1 Conclusion. 56

8. Bibliography. 58

**List of Figures**

Figure 3.1 Waterfall Model 27

Figure 3.2 Incremental Model 27

Figure 3.3 Agile Model 28

Figure 3.3 Adopted Methodology. 30

Figure 4.1 Use Case 1. 35

Figure 4.1-2 Use Case 2 Login. 36

Figure 4.2 Use Case3 Registration. 37

Figure 4.3 User Functionalities. 38

Figure 4.4 Use Case Admin Functionalities. 39

Figure 4.5 Use Case logout 40

Figure 4.6 Warden. 41

Figure 4.7 User Activity. 42

Figure 4.8 Admin Activity. 43

Figure 4.9 Warden Activity. 43

Figure 4.10 Cashier Activity. 44

Figure 4.11 Sequence diagram of User Login. 45

Figure 4.12 Admin Dashboard. 46

Figure 4.13 Sequence diagram of User Dashboard. 47

Figure 4.14 Warden Dashboard. 48

Figure 4.15 Cashier Dashboard. 49

Figure 4.16 Deployment Diagram.. 50

Figure 4.17 Class Diagram.. 51

**List of Tables**

Table 3‑1 Comparison Table of three Models. 23

Table 14‑1: Testing Cases. 66

Table 14‑2 Testing Result 68

Chapter 1

**INTRODUCTION**

# **1. Introduction*:***

Wifi toy car controlling is designed and developed for Intelligence & Technology. It is widely applied in many fields, such as the electronic lover, college students, Internet of Things (IoT), mobile data collection, etc. Wifi toy car has a great temptation for the smart car solution.

The wifi toy car kit is including the car chassis, 4 pieces of 3V motors, NodeMCU WiFi board and Webcam. Especially, all the codes and hardware’s are open for all people.

WiFi toy car is designed and developed based on NodeMC to control the car to move on lefts, right, straghit and back.

## **1.1 Purpose:**

The wifi tpy car controlling app can

Ø Toy car is ready for any mission in search of new life forms

### **Faster, smarter, better…wifi toy car goes where no app-controlled car has gone before.**

Ø If our mission roam around you need to add this secret agent bot to your arsenal.

### **Offline Entertaining device.**

Ø User can view the location

Ø Keep a video record

Ø Keep a record of location of car

Ø Not only is this toy car operated with any android based mobiles phones, but a WiFi enabled connection makes it easy to connect and operate

## **1.2 History:**

Pakistan is a developing country and it also have to use modern techniques. Before of this toy car were controlled with remote system that cannot contained with any video features and wires system that have minimum distance covering. But now we use toy car as intelligence by using cameras and long distance of coverage using wifi connection to controlled the car without any wire system. So we can say that the toy cars can intelligence toy like find the person or delivered something one place to other place.

## **1.3 Clear Statement of the problem:**

Motorists particularly those belonged to others cities of the Punjab province complained that they had to suffer a lot during bank holidays because of manual ticketing system. Since banks are closed in the evening, there is no way to deposit fine amount during the night hours. Traffic officers routinely seize registration books or driving licenses after issuing challan tickets to the drivers.

According to research and findings there are many types and categories of traffic violations committed by the road users (driver). For every violation road user has to bear specific amount of fine. Officers may observe multiple violations for one road user.

## **1.4 Actors Catalogue:**

The system will need offline wifi connectivity for working as the application is an offline application and the data will be stored and retrieved from a camera. Following are the Actors Catalogue for the system.

## **1.6 Objective:**

The objectives are important to achieve the goal. The main objectives of this project are:

Ø Make a reliable, inexpensive app.

Ø Fun & entertaining

Ø Easy to do drive.

Ø Constant monitoring.

Ø Information received through camera.

Ø Admin can alter any table anyway they want.

## **1.7 Tools and Technology:**

Tools we use in this project are as follows:

### **Software Tools:**

### **Android Studio**

## **1. Car chases kit**

## **2.NodeMCU**

NodeMCU is an [eLua](http://www.eluaproject.net/) based firmware for the [ESP8266 WiFi SOC from Espressif](http://espressif.com/en/products/esp8266/). The NodeMCU *firmware* is a companion project to the popular [NodeMCU dev kits](https://github.com/nodemcu/nodemcu-devkit-v1.0), ready-made open source development boards with ESP8266-12E chips.

## **3.DC Motor controller**

Is designed for design and manufacturing engineers, maintenance technicians, or anyone requiring a basic understanding of DC motor controllers, functions, operations, and applications. All terms used are explained or defined throughout the courses, so participants are not required to have an extensive technical vocabulary to understand the

## **4.Bread Board**

Breadboards are one of the most fundamental pieces when learning how to build circuits. In this tutorial, you will learn a little bit about what breadboards are, why they are called breadboards, and how to use one. Once you are done you should have a basic understanding of how breadboards work and be able to build a basic circuit on a breadboard.

## 5**. Batteries**

Battery is normaly use for power. And we have required 12 volts so we use three cells of 3.7 volts rechargeable cells. through the dc motors we can converts into 5 volts.

## **5. V380 camera**

V380 IP Camera products can be used to achieve remote home monitoring and video playback. That is to say, it can be easily to come true the remote video monitoring and management. You can download the new generation of V380 smart home cloud camera application on our website for free. We will offer the most comprehensive and quick guide to help you set it up, which will become your most reliable home security housekeeper.

.

CHAPTER 2

STATE OF THE ART

# **2. STATE OF THE ART:**

In this chapter, we will discuss what is wifi toy car controlling app providing?

## **2.1 Related Work:**

With the globalization and advancements in technology, Challan system is also having improvements with time. We have recent research on this application and there is no more work done on this. The systems are available but those are not so well received and those are too in Pakistan.

In Pakistan there is no artificial intelligence techniques for developing wifi cars government is talking about doing something in this field but no solid step is taken in this direction which is a right direction to move on. So, we have introduced this app.

## **2.2 Limitations of System:**

Limitations of this systems are as follows:

### **2.2.1 offline wifi Availability**

To use this application, the not internet is very important. Without internet this system will work and run on system.

### **2.2.2 Must be a educated**

The user who know how to attract or deal with the app to control it.

## **2.3 The Necessity:**

There are some necessities for this android are as follows:

### **2.3.1 Camera Confidentiality**

Camera Confidentiality must be maintained the video recording. By this user have full control over car system.

### **2.3.2 Accurate videos**

Camera must show accurate records

### **2.3.3 Better Communication**

Meet now must provide the facility of better communication using high speed offline wifi signals connection.

CHAPTER 3

**METHODOLOGY & WORKPLAN**

# **3. METHODOLOGY & WORKPLAN**

In this chapter, we will discuss that what are the existing methodologies and which one we have chosen for implementation of this project in an effective way, also we will discuss advantages of adopted methodology.

## **3.1 Proposed Application Architecture**

The problems which are occurred in the existing applications are overcome in proposed application. In this application we are implemented all of the functionality by using any device with internet access for organization to handling the communication between the Admin and User.

## **3.2 Existing Methodology:**

There are the following existing methodologies

### **3.2.1 Waterfall Model**

In the software development process, the very first model that is published from other engineering processes that is cascaded from one phase to another is known as Waterfall model. This model is also known as linear sequential model. In the waterfall model, firstly all of the requirements gathered.

After gathering all of the requirements and analysis of all the requirements further move to next phase that is making the design of a project and then implementation, testing and maintenance phases. As all of the next phases of this model is dependent on previous phase. All the phases worked as a waterfall after one another.

It is a documentation driven model. In this model, feedback of the client is received at the end of the project and when the product is delivered to the user. Hence, mistake in the product requirement specification is discovered after delivered to the user. This is time and cost consuming methodology, that is why we have not used this methodology in developing this project.

The waterfall model is further described in the following diagram:

#### **Figure 3.1 Waterfall Model**

### **3.2.2 Incremental Model**

To overcome the drawbacks of the waterfall model, incremental model is used. In incremental model, the product is developed in increments and partitioned into smaller pieces. These smaller pieces, then built and delivered to client in increments. Quick response from clients. Each module is smaller than compare to whole module

#### **Figure 3.2 Incremental Model**

### **3.2.3 Agile Model**

Agile model is useful when there is no hope for changings in product after deployment. It is a particular approach to project management that is utilized in software development. This method assists teams in responding to the unpredictability of constructing software. It uses incremental, iterative work sequences that are commonly known as sprints. The ability to create and respond to change in order to succeed in an uncertain and turbulent environment.

|  |  |
| --- | --- |
|  |  |

#### **Figure 3.3 Agile Model**

Comparison Table of three Models

|  | **Model** | **Strengths** | | **Weaknesses** | |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Waterfall**  **Model** | | | 1). Minimizes planning overhead since it can be done up front.  2). Structure minimizes wasted effort, so it works well for technically weak or inexperienced staff. | | 1). Inflexible  2). Only the final phase produces a no documentation deliverable.  3). Backing up to address mistakes is difficult. | |
|  | **Incremental Model** | 1). Develop high-risk or major functions first.  2). Risk is spread across smaller increments instead of concentrating in one large development.  3). Lessons learned at the end of each incremental delivery can result in positive revisions for the next increment.  4). Customers get important functionality early and have an opportunity to respond to each build.  5). Each release delivers an operational product. · Initial product delivery is faster.  6). Reduces the risk of failure and changing the requirements. | | 1). Requires good planning and design.  2). Requires early definition of a complete and fully functional system to allow for the definition of increments.  3). The model does not allow for iterations within each increment. | |  |
|  | **Agile Model** | 1). Allows for changes to be made after the initial planning.  2). Easier to add features that will keep you up to date with the latest developments in your industry. | | 1). Inflexible  2). Project can become a series of code sprints in lack of proper management.  3). Initial project doesn’t have a definitive plan | |  |
|  |  |  |  |  |  |  |

Table 3‑1 Comparison Table of three Models

## **3.3 Adopted Methodology:**

Incremental model is used to develop this project, in which we divided our work in multiple modules. All these modules are further divided into more easily managed modules which made up the actual implementation of the requirements.

#### **Figure 3.3 Adopted Methodology**

## **3.4 Roles & Responsibilities:**

Project development team is consisting of two members. In order to accomplish a goal, documentation and development is equally distributed among them and each member work on parallel to avoid wastage of time.

CHAPTER 4

SYSTEM ANALYSIS & DESIGN

# **4. SYSTEM ANALYSIS & DESIGN**

In this chapter requirements analysis, feasibility study, planning, forecasting, modeling, scheduling and design of the project is discussed. For developing any project, the major problem is requirement gathering. Asking questions from clients is straightforward than collecting requirements.

We will also focus on functional and non-functional requirements. The procedure for gathering requirements has its own defined procedure according to the complexity of the application. To define project schedule and processing, different models and techniques also focused on this chapter.

A requirement can be defined as a condition or capability that must be processed by a product or an application. Techniques that can be used for collecting requirements are as follows:

Ø By questionnaire and survey

Ø By interview

Ø By observations

Ø Using software tools

Ø Using techniques for decision making

Ø Focus on facilitated groups and workshops

Ø Use of prototype

Ø The techniques we have used to collect requirements are observations and interviews.

## **4.1 Requirement Analysis***:*

Requirements analysis is the process of planning, forecasting and studying the overall former needs of the application requirements. Requirements analysis is further divided into two parts:

Ø Functional Requirements

Ø Non-Functional Requirements

### **4.1.1 Functional Requirements**

Functional requirements define the fundamental actions that system must perform. The functional requirements for the System are divided into two parts, one is software part and second is hardware part.

In software part there categories:

Ø User profile

On the offline portal the categories will;

Ø User profile

### **Login as a User by camera to app:**

The user will be install the app on mobile. when he opened it will be show an dialog box to enter our device to connect with wifi then camera will show the output on pc. V380 app have an local IP for camera.

### **4.1.2 Non-Functional Requirements:**

Functional requirements define the needs in terms of performance, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

### **Availability:**

Ø Application must be responsive and available at every time.

Ø Availability of high speed offline wifi connection is the major requirement of the application.

Ø Application must be work on time efficiently.

Ø Application connectivity time must be good.

### **Maintainability:**

Ø Making changes or upgradeability in the site will not be that much difficult. By having some knowledge of programming, some features of the application might be converted to a new version.

### **Performance:**

Ø This application must perform the action on time without any long delay.

Ø Over Application through an output in time.

Ø This Application perform efficiently.

### **Security:**

Ø This Application prevent from unauthorized modification of information.

### **Usability:**

Ø This application fulfills the requirement of local.

Ø All type of people must access of this application.

### **Portability:**

Ø This is android-based application that is why there is no problem in portability process.

### **Supportability**:

Ø This Application support and provide Information helpful for identifying and resolving issues when it fails to work correctly.

## **4.2 Use Case:**

An important part of the analysis phase is to drawing the diagrams of Use cases. They are used through the phase of analysis of a project to find and divide functionality of the application. Application is separated into actors and use cases. Actors play the role that are played by the application users.

Use cases define the application behavior when one of the actors sends any particular motivation. This type of behavior can be described by text. It describes the motivation nature that activates use case, the inputs and outputs to some other actors and the behavior of conversion of inputs to the outputs. Usually the use case describes everything that can go wrong during the detailed behavior and what will be helpful action taken by the application.

Some of the use cases are as follows:

##### **Use Case 1 Login/Sign up:**

**Actor:**

Ø Actor must be a user or an Admin.

#### **Figure 4.1 Use Case 1**

**Basic Path:**

Ø The Actor enters user information to camera app.

Ø The Actor clicks on SIGN UP button.

**Non-Functional Requirements:**

Ø Better response

Ø Robustness

Ø Reliable

##### **Use Case 2 Login:**

#### **Figure 4.1-2 Use Case 2 Login**

**Actor:**

Ø User may be Citizen, Cashier or may be Admin.

**Pre-Condition:**

Ø For the member, he/she must have registered already in application by the administrator.

Ø User must enter correct Name address and password for login.

**Post-Condition:**

Ø User can access to application’s main features.

**Basic Path:**

Ø Enter Name and password for login.

Ø The application verifies the correct format and valid Name and Password.

Ø If provided inputs are correct, the application displays the all other user’s content of the application and session of particular user started.

**Constraints:**

Ø If provided email address and password are incorrect or invalid, application redirect to main page.

**Non-Functional Requirements:**

Ø Short Response Time

Ø Better performance

Ø Availability

Ø Robustness

##### **Use Case 3 Registrations:**

#### **Figure 4.2 Use Case3 Registration**

**Actor**:

Ø User

**Pre-Condition:**

Ø The actor must add complete mandatory details.

**Post-Condition:**

Ø User Account should be created successfully.

Ø Member particular folder is created in project directory.

**Basic Path:**

Ø The Actor should be opened in to the application to perform this activity.

Ø The Actor enters user information.

Ø The Actor clicks on SIGN UP button.

Ø After verifying provided information; the application will generate the member account detail file.

**Alternative Path:**

Ø On wrong entries of mandatory fields, the application displays particular message for invalid entry or required fields.

**Non-Functional Requirements:**

Ø Better response

Ø Robustness

Ø Reliable

##### **Use Case 4 User Functionalities:**

#### **Figure 4.3 User Functionalities**

**Actor:**

Ø User

**Basic Activities:**

Ø Sign Up

Ø Log In

Ø View Challan with Number Plates

Ø View Challan with CNIC/License

Ø See updates and news

Ø Logout

##### **Use Case 5 Admin Functionalities:**

#### **Figure 4.4 Use Case Admin Functionalities**

**Actor:**

Ø Admin

**Basic Activities:**

Ø Log in

Ø See challan history

Ø Update challan form

Ø Update warden login form

Ø Send notification

Ø Logout

##### **Use Case 6 Cashier:**

#### **Figure 4.5 Use Case logout**

**Actor:**

Ø User

**Basic Functionality:**

Ø Login

Ø See challan history

Ø Update challan form after challan submission

Ø Logout

##### **Use Case 7 Warden:**

#### **Figure 4.6 Warden**

**Actors**

Ø Warden

**Basic Functionality**

Ø Login

Ø See user challan history

Ø Do new challan on a violation

Ø See regular violators

Ø See standard rates of violations

Ø Logout

## **4.3 Activity Diagrams:**

A diagram of the sequence of movements or actions of people or things involved in a complex system or activity.

##### **User Activity:**

#### 

#### **Figure 4.7 User Activity**

##### **Admin Activity:**

#### **Figure 4.8 Admin Activity**

##### **Warden Activity:**

#### **Figure 4.9 Warden Activity**

##### **Cashier Activity:**

#### **Figure 4.10 Cashier Activity**

## **4.4 Sequence Diagrams:**

Sequence diagram uses concept of a Message-Sequence-Chart. It shows interactions of objects in a sequence of time. It shows the classes and objects involved in the scenario and the message sequence between the objects which is desired to carry out the functionality of a given scenario.

Sequence diagrams are usually related with the understanding of use case in the logical View of the system which is under development. “Sequence diagrams are sometimes called event diagrams, timing diagrams, event scenarios”. A parallel vertical line on sequence diagram is called lifeline.

Different objects or processes that live simultaneously, and, on horizontal arrows, the messages exchanged, in the order in which processes occur. This allows some specification of some simple runtime scenarios in a graphical pattern.

### **Sequence diagram of User/Admin/Cashier Login:**

#### **Figure 4.11 Sequence diagram of User Login**

### **Sequence diagram of Admin Dashboard**

#### **Figure 4.12 Admin Dashboard**

### **Sequence diagram of User Dashboard:**

#### **Figure 4.13 Sequence diagram of User Dashboard**

### **Sequence diagram of Warden Dashboard:**

#### **Figure 4.14 Warden Dashboard**

### **Sequence diagram of Cashier Dashboard:**

#### **Figure 4.15 Cashier Dashboard**

## **4.5 Deployment Diagram**

#### **Figure 4.16 Deployment Diagram**

## **4.6 Class Diagram**

#### **Figure 4.17 Class Diagram**

CHAPTER 5

**SYSTEM IMPLEMENTATION**

# **5. SYSTEM IMPLEMENTATION**

In this chapter, we’ll focus on an implementation of our application. Where user can perform many activities on android application and Admin can control all others.

## **5.1 Introduction**

The most important goal of this phase is to develop the application. The work in this phase should be much more straightforward as a result of the work done in the planning and design phases. This phase involves changing design specifications into executable programs. When the design is there, developers can have an idea on looks of application. All that is needed by developers is to put them at one place to understand about the intended project.

## **5.2 Screenshots:**

### **5.2.1 Android Screenshots:**

**CHAPTER 6**

**SYSTEM TESTING**

# **6. SYSTEM TESTING**

In this chapter, we will discuss the testing phase of developed application **“TCB”** in different manner to know that how much efficient and effective application is.

## **6.1 Introduction**

A process of performing as application or program with the intention of finding errors and whether the application is fulfilling user needs. It can also be defined as the ability of a program in meeting the required or desired results. In many methodologies of software engineering, a separate phase is called phase of testing which is performed after the completion of the implementation. There is a benefit in using this approach that it is hard to see one's own mistakes, and a fresh eye can find observable errors much faster than the person who has read the material many times.

## **6.2 Testing Plan**

A process of performing as application or program with the intention of finding errors and whether the application is fulfilling user needs.

## **6.3 Unit Testing**

The software units in an application are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. The various controls are tested to ensure that each performs its action as required.

Commonly used method is White-Box Testing method. Every time a component of the program is changed, it can be run for testing that is the biggest and famous benefit of this testing phase. Issues that are arises during this phase, allowing to be resolved as quickly as possible. Unit testing is familiar by software developers. It allows them to test their application units before move them to testers for formal testing.

## **6.4 System Testing**

To test the complete application as a whole, system testing has been used. It is beneficial to check whether the application meets its requirements and fulfill Quality Standards.

## **6.5 Integration Testing**

Integration testing allows the software developers to integrate all of the components/ units of the application within a program and then test them in a group. Basically, this testing level is used to catch the defects in the user interface between the functions/ modules. It is useful to determine how logically and efficiently all the units/ components are running together.

Here the streaming module and encoding module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

## **6.6 User Acceptance Testing**

User acceptance of an application is the key factor for the success of any application. The application under consideration is tested for user acceptance by constantly keeping in touch with the application users at time of developing and making changes whenever required.

## **6.7 Test Cases**

| Test Cases | **Objectives** |
| --- | --- |
| 1 | To make sure that user can easily understand and can use the application |
| 2 | Make sure that user can easily login |
| 3 | Make sure request send successfully. |
| 4 | Make sure that all the requests receive successfully |
| 5 | Make sure search will be easily |
| 6 | Make sure all the user can see history easily |
| 7 | Make sure that notification can display successfully |
| 8 | Make sure that the application can fulfill the gap successfully |
| 9 | Make sure that the application run at cross-platforms successfully. |

Table 14‑1: Testing Cases

## **6.8 Testing Results**

| **CRITERIA** | **Test Status** | **REMARKS** |
| --- | --- | --- |
| All the graphical user interface options display successfully. | Test successful | None |
| Enter valid login user Name and password and then press login | Test successful | None |

Table 14‑2 Testing Result

**CHAPTER 7**

**CONCLUSION & FUTURE WORK**

# **7. CONCLUSION & FUTURE WORK**

In this chapter, we will discuss the results and discussions of this framework “**TBC**” with conclude remarks and will also discuss related future work of this application.

## **7.1 Conclusion**

We are very delighted working with this project and being able to develop a system to integrate with the local need of our country and its citizens. This documentation gives whole detail of a "Wifi Toy Car controlling" is to controlled with an application through this application firstly connect with wifi of our mobile device and the wifi modular of the car. And the camera also will be on for the view, then we can see where were now. Through app we will command the car to move on left, right, backward and forward anywhere we want or our target place

**CHAPTER 8**

**REFERENCE**

# **8. Bibliography**

1. Admin. (2018, May 1). *testing-methodology*. Retrieved from https://www.guru99.com: https://www.guru99.com/testing-methodology.html

2. Boryoswich, C. (2008, April 03). *developing-project-strategies-plans-052908*. Retrieved from https://it.toolbox.com: https://it.toolbox.com/blogs/craigborysowich/developing-project-strategies-plans-052908

3. Bradfared, L. (2002, September 30). *all-you-need-to-know-about-software-testing-methods-4019921*. Retrieved from https://www.thebalancecareers.com: https://www.thebalancecareers.com/all-you-need-to-know-about-software-testing-methods-4019921

4. Buttler, j. (2013, March 13). *main*. Retrieved from https://ctb.ku.edu: https://ctb.ku.edu/en/table-of-contents/structure/strategic-planning/develop-strategies/main

5. Ceta, N. (2013, September 18). *uml-diagram*. Retrieved from https://tallyfy.com: https://tallyfy.com/uml-diagram/

6. Comey, J. (2017, October 11). *uml-diagram-types-examples*. Retrieved from https://creately.com: https://creately.com/blog/diagrams/uml-diagram-types-examples/

7. Glechi, J. (2014, October 30). *What-are-the-different-software-testing-methodologies*. Retrieved from http://searchsoftwarequality.techtarget.com: http://searchsoftwarequality.techtarget.com/answer/What-are-the-different-software-testing-methodologies

8. GOP. (2017, November 02). *e-challan-in-lahore*. Retrieved from https://www.pakwheels.com: https://www.pakwheels.com/blog/e-challan-in-lahore/

9. Hobbs, A. (2018, June 17). *software-development-testing-methodologies*. Retrieved from https://www.softwaretestinghelp.com: https://www.softwaretestinghelp.com/software-development-testing-methodologies/

10. ICRS. (2003, March 11). *plan-develop-strategy-and-manage-projects*. Retrieved from https://icrs.info: https://icrs.info/cpd/competency-framework/plan-develop-strategy-and-manage-projects

11. ITP. (2009, February 02). *traffic-fine-payment*. Retrieved from http://islamabadpolice.gov.pk: http://islamabadpolice.gov.pk/ipwe/traffic-fine-payment.php

12. ITP. (2010, January 09). *violation-and-fine*. Retrieved from http://www.islamabadtrafficpolice.gov.pk: http://www.islamabadtrafficpolice.gov.pk/violation-and-fine.php

13. ITP. (2015, March 11). *islamabad-traffic-police-launches-e-challan-system-for-violators*. Retrieved from https://en.dailypakistan.com.pk: https://en.dailypakistan.com.pk/pakistan/islamabad-traffic-police-launches-e-challan-system-for-violators/

14. K., A. (2008, July 09). Retrieved from web.nchu.edu.tw: web.nchu.edu.tw/pweb/users/arborfish/lesson/10320.pdf

15. LTP. (2015, june 25). *47765-traffic-police-launch-e-challan-system*. Retrieved from https://www.thenews.com.pk: https://www.thenews.com.pk/print/47765-traffic-police-launch-e-challan-system

16. Release, P. (2018, January 08). *nbp-islamabad-traffic-police-sign-agreement-e-challan-payments*. Retrieved from https://propakistani.pk/2018/01/08/: https://propakistani.pk/2018/01/08/nbp-islamabad-traffic-police-sign-agreement-e-challan-payments/

17. Robbie, M. (2001, august 12). *agile-methodology-guide-agile-testing*. Retrieved from https://www.qasymphony.com: https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/

18. S., Z. (June, 13 2018). *agile-software-development*. Retrieved from https://searchsoftwarequality.techtarget.com: https://searchsoftwarequality.techtarget.com/definition/agile-software-development

19. Sacolick, I. (2005, April 04). *what-is-agile-methodology-modern-software-development-explained*. Retrieved from https://www.infoworld.com: https://www.infoworld.com/article/3237508/agile-development/what-is-agile-methodology-modern-software-development-explained.html

20. Saqib, T. (2018, March 18). *e-challan-system-islamabad*. Retrieved from https://www.techjuice.pk: https://www.techjuice.pk/e-challan-system-islamabad/

21. Short, k. (1999, june 08). *project-strategy*. Retrieved from http://ttthinc.com.au: http://ttthinc.com.au/services/advice/project-strategy/

22. sicon, s. (2009, June 20). *agile-101*. Retrieved from https://www.versionone.com: https://www.versionone.com/agile-101/

23. *software-testing-methods*. (2018, 01 21). Retrieved from http://softwaretestingfundamentals.com: http://softwaretestingfundamentals.com/software-testing-methods/

24. UNknown. (2015, June 23). *developing-strategic-and-action-plans*. Retrieved from https://ctb.ku.edu: https://ctb.ku.edu/en/developing-strategic-and-action-plans

25. Wahad, S. (2018, April 13). *digital-challan-app-launched*. Retrieved from https://www.pakistantoday.com.pk: https://www.pakistantoday.com.pk/2018/04/13/digital-challan-app-launched/

26. Z.Milosevicc, P. J. (2012). *Journal of Engineering and Technology Management.* new york: Fedreal Publishment.

27. Zrar, S. (2016, December 01). *automatic-traffic-challans-cctv-e-ticket-system-lahore-finally-approved*. Retrieved from https://propakistani.pk/2016/12/01/: https://propakistani.pk/2016/12/01/automatic-traffic-challans-cctv-e-ticket-system-lahore-finally-approved/