

SMART LICENSE PLATE

A Project Report

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

V USHA (158T1A05A9)

V TEJA BALAKRISHNA (158T1A05A8)

SK SAMARIN BANU (158T1A0594)

D CHANDRIKA (158T1A05B8)

Under the esteemed guidance of

Mrs. CH PADMINI B. Tech., M. Tech., (PhD)

Assistant Professor



Department of Computer Science and Engineering

DHANEKULA INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE-New Delhi, Affiliated to JNTU, Kakinada)

Programmes Accredited by NBA : B. Tech in CE, MEC, ECE & EEE

NAAC Accredited & An ISO 9001 – 2015 Certified Institution

GANGURU, A.P. (INDIA) - 521139

APRIL 2019

DHANEKULA INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE-New Delhi, Affiliated to JNTU, Kakinada)

Programmes Accredited by NBA : B. Tech in CE, MEC, ECE & EEE

NAAC Accredited & An ISO 9001 – 2015 Certified Institution

GANGURU, A.P. (INDIA) - 521139



CERTIFICATE

This is to certify that the project titled “**SMART LICENSE PLATE**” is a bonafide work carried out by **V USHA (158T1A05A9)**, **V TEJA BALAKRISHNA (158T1A05A8)**, **SK SAMARIN BANU (158T1A0594)**, **D CHANDRIKA (158T1A05B8)** are students of B. Tech (CSE) of DhaneKula Institute of Engineering and Technology, affiliated to JNTU, Kakinada, AP(India) during the academic year 2018-19, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Signature of the Guide

Mrs. CH PADMINI
Assistant Professor

Signature of the HOD

Dr. S SURESH
Professor & HOD

External Examiner

DECLARATION

We hereby declare that the project entitled “**SMART LICENSE PLATE**” submitted for the B. Tech (CSE) degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

V USHA (158T1A05A9)

V TEJA BALAKRISHNA (158T1A05A8)

SK SAMARIN BANU (158T1A0594)

D CHANDRIKA (158T1A05B8)

ACKNOWLEDGMENT

We would like to take this opportunity to express our deep gratitude to the members who assisted us directly and indirectly for the completion of this project work.

We would like to thank **CH PADMINI, Assistant Professor**, the project guide for her esteemed guidance and support, especially the valuable ideas and thoughts provided during this project work. She is expertise in the area of **Data Mining, Sentiment Analysis & Opinion Mining** this helps us in solving the problems encountered during the project work.

We would like to express our immense pleasure in expressing immeasurable sense of gratitude to, **Dr. SURESH SUNDARADASU, Professor & Head of the Department in Computer Science and Engineering** for his valuable suggestions in the completion of the project.

We whole heartedly acknowledge **Dr. RAVI KADIYALA, Principal, and K SANDEEP, Assistant Professor, Project Coordinator** for giving opportunity to make this project a successful one.

We also extend our thanks to all the faculty members of **Computer Science and Engineering** department for their valuable contributions in this project.

We would like to extend our warm appreciation to all our friends for sharing their knowledge and valuable contributions in this project.

Finally, we express our deep sense of gratitude to our parents for their continuous support throughout our academic carrier and their encouragement in completion of this project successfully.

V USHA (158T1A05A9)

V TEJA BALAKRISHNA (158T1A05A8)

SK SAMARIN BANU (158T1A0594)

D CHANDRIKA (158T1A05B8)

ABSTRACT

This work manages the making of a real-time smart license plate (vehicle number plate), where all the fundamental subtle elements of a vehicle (i.e., Driving License, Registration Certificate, Insurance Policy and vehicle's Pollution Under Control Certificate) are coordinated and stored by the user without much effort. Quick Response Code (QR-Code), an exceptionally powerful and quick decipherable innovation is utilized for Storing and examining different subtle elements of a vehicle. We create a website through which user can generate their unique QR-code for their vehicle or can update the details of the vehicles. In perspective of the security dangers of uncovering individual subtle elements of clients, more noteworthy consideration has been given for securing the information by utilizing QR-Code and securing access to those points of interest.

DHANEKULA INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Computer Science & Engineering

VISION – MISSION - PEOs

Vision/Mission/PEOs

Institute Vision	Pioneering Professional Education through Quality
Institute Mission	<p>Providing Quality Education through state-of-art infrastructure, laboratories and committed staff.</p> <p>Moulding Students as proficient, competent, and socially responsible engineering personnel with ingenious intellect.</p> <p>Involving faculty members and students in research and development works for betterment of society.</p>
Department Vision	To empower students of Computer Science and Engineering Department to be technologically adept, innovative, global citizens possessing human values.
Department Mission	<p>To Encourage students to become self-motivated and problem-solving individuals.</p> <p>To prepare students for professional career with academic excellence and leadership skills.</p> <p>To Empower the rural youth with computer education.</p> <p>To Create Centre's of excellence in Computer Science and Engineering.</p>
Program Educational Objectives (PEOs)	<p>Graduates of Computer Science & Engineering will:</p> <p>PEO1: Excel in Professional career through knowledge in mathematics and engineering principles.</p> <p>PEO2: Able to pursue higher education and research.</p> <p>PEO3: Communicate effectively, recognize, and incorporate societal needs in their professional endeavors.</p> <p>PEO4: Adapt to technological advancements by continuous learning.</p>

DHANEKULA INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Computer Science & Engineering

PO's

1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental Considerations
4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PSO's

PSO1: Have expertise in algorithms, networking, web applications and software engineering for efficient design of computer-based systems of varying complexity. **PSO2:** Qualify in national international level competitive examinations for successful higher studies and employment.

PO's & PSO's Mappings

Project Vs PO's Mapping:

Project Title	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
SMART LICENSE PLATE	2	2	2	2	3	2	1	3	3	2	3	1

Project Vs PSO's Mapping:

Project Title	PSO1	PSO2
SMART LICENSE PLATE	3	1

3 – Strong Correlation, 2- medium Correlation, 1- Low Correlation

Signature of the Guide

Mrs. CH PADMINI

Assistant Professor

LIST OF FIGURES

S. No.	Name of the Figure	Page No.
1	Flowchart	15
2	ER Diagram	17
3	Class Diagram	18
4	Use case Diagram	19
5	Sequence Diagram	21
6	Collaboration Diagram	23
7	State chart Diagram	25
8	Activity Diagram	27
9	Component Diagram	29
10	Deployment Diagram	30

TABLE OF CONTENTS

Title Page	i
Declaration	ii
Certificate	iii
Acknowledgement	iv
Abstract	v
Vision-Mission-PEO's	vi
PO's & PSO's Description	vii
PO's & PSO's Mappings	viii
List of Figures	ix
	Page No.
1. INTRODUCTION	1 - 2
1.1. Problem Definition	1
1.2. Project Overview	1
1.3. Hardware Specifications	2
1.4. Software Specifications	2
2. LITERATURE SURVEY	3 - 5
2.1. Existing System	3
2.2. Proposed System	4
2.3. Feasibility Study	4
2.3.1. Technical Feasibility	4
2.3.2. Operational Feasibility	5
2.3.3. Economic Feasibility	5
3. SYSTEM ANALYSIS	6 - 11
3.1. Study of the System	6
3.2. Number of the Modules	6

3.3.	Module Description	6
3.4.	Process Flow	7
3.5.	SDLC Methodologies	9
4.	SYSTEM DESIGN	12 - 30
4.1.	DFD	12
4.2.	UML	16
4.3.	ER	17
5.	TECHNOLOGY DESCRIPTION	31 - 41
5.1.	HTML	31
5.2.	JAVASCRIPT	32
5.3.	JAVA	33
5.4.	JDBC	35
5.5.	Database	37
5.6.	Servlets	39
5.7.	Eclipse IDE	41
6.	TESTING & DEBUGGING TECHNIQUES	42 - 46
6.1.	Testing Methodologies	42
6.2.	Levels of Testing	42
6.3.	Types of Testing	42
6.4.	STLC	44
6.5.	Test cases	45
7.	OUTPUTS	47 - 54
8.	CONCLUSION	55
9.	REFERENCES	56 - 57

INTRODUCTION

1. INTRODUCTION

1.1 Problem Definition:

Most of the Urban people spend average of 2.5 hours a day commuting by vehicles. Means 38 hours in a year. In Metropolitan areas double the time. When a traffic police stops them on the road to verify the credentials of their vehicle the commuter loses more of his/her precious time due to manual checking.

1.2 Project Overview:

This work manages the making of a real-time smart license plate (vehicle number plate), where all the fundamental subtle elements of a vehicle (i.e., Driving License, Registration Certificate, Insurance Policy and vehicle's Pollution Under Control Certificate) are coordinated and stored by the user without much effort. Quick Response Code (QR-Code), an exceptionally powerful and quick decipherable innovation is utilized for Storing and examining different subtle elements of a vehicle. We create a website through which user can generate their unique QR-code for their vehicle or can update the details of the vehicles. In perspective of the security dangers of uncovering individual subtle elements of clients, more noteworthy consideration has been given for securing the information by utilizing QR-Code and securing access to those points of interest.

It is a known fact and has been established by various studies that, most of the urban people spend at an average 2.5 hours a day commuting by vehicle. On larger metropolitan cities the commuting time is almost twice than the original time required to reach the destination because of unavoidable traffic congestions, disproportionate number of vehicles a road can accommodate and due to rush in peak hours. It is estimated that, an average commuter spends 38 hours a year on his/her vehicle commuting. In this scenario, when the traffic police stop them on the road to verify the credentials of their vehicle the commuter loses more of his precious time.

1.3 Hardware Specifications:

Processor	:	Core2Duo Processor
Hard Disk	:	160GB
RAM	:	2GB or more

1.4 Software Specifications:

Operating System	:	Windows 7/8/10
User Interface	:	HTML, CSS
Client-side Scripting	:	JavaScript
Programming Language	:	Java
Web Applications	:	Hibernate, Servlets, JSP
IDE/Workbench	:	Eclipse
Database	:	Oracle 10g XE
Server Deployment	:	Tomcat 8.0

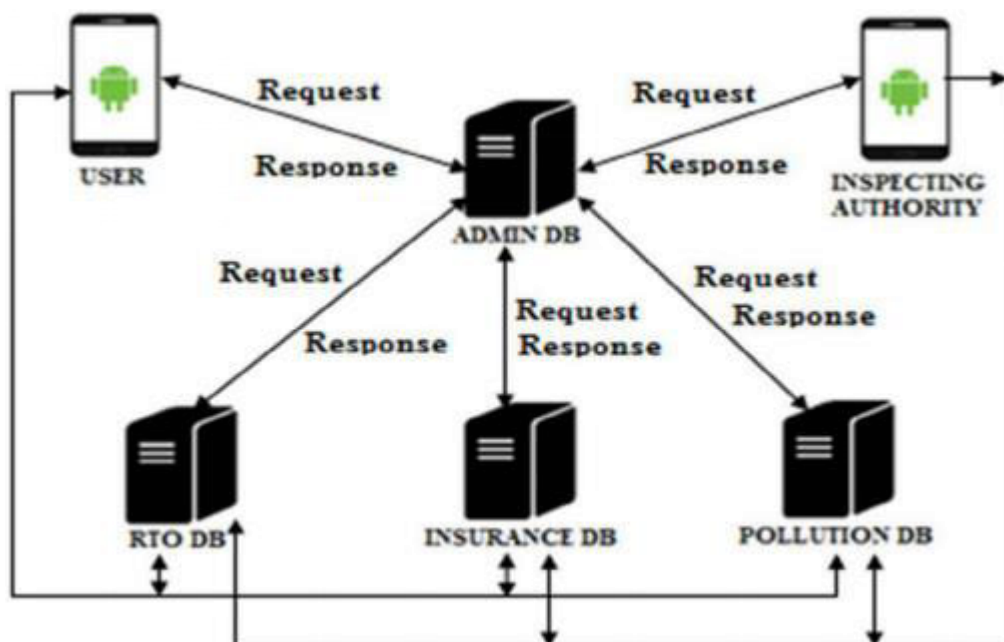
All over the world, as per the Motor Vehicles Act of the respective countries, it is mandatory that the users, of the vehicles running on the road should carry relevant documents pertaining to it. Those documents include a valid Registration Certificate Book (RC-Book), vehicle's Insurance Policy and the vehicle's Pollution Under Control (PUC) certificate. All these documents are checked by the inspecting authority of the traffic police department. Since the whole process is done manually, it consumes a lot of time and there is possibility for malfeasance due to a lack of transparency in the existing system.

LITERATURE SURVEY


2. LITERATURE SURVEY

2.1 Existing System:

- An android application for the user (User Application).
- Three servers corresponding to the three departments providing relevant documents of the vehicle (Department Servers)
- A centralized database administrator (Admin Server).
- An android application for the inspecting authority (Authority Application).



RESULT:

Checking Authority NAME Shivani <div>VEHICLE QR</div> <div>SETTINGS</div>	Name : Shivani Email : asha26@gmail.com Phone: 9926526425 	Checking Authority VEHICLE BLOCKED Name: Rahul Reg.No: KL-03-1991 E-mail: rahul@gmail.com Mobile: 9946854825 <div> INSURANCE</div> <div> RC BOOK</div> <div> POLLUTION</div>	Checking Authority RTO DETAILS Chassis No: MD626B Seating Capacity: 2 Reg.No: KL-03-1991 Dealer Name: AVG Motors Dealer Address: Pandalam Owner Name: Rahul Owner Address: Nandanam PTA License No: 1/42584/2006
--	--	---	--

Disadvantages:

- It is an android application and the data is only viewed by the registered user and registered inspecting officer.
- Though it is portable the SVAM-ID is not transferrable.
- Not all documents required to create a qr-code.

2.2 Proposed System:

- We design a real-time smart license plate (vehicle number plate), where all the fundamental subtle elements of a vehicle (i.e., Driving License, Registration Certificate, Insurance Policy and vehicle's Pollution Under Control Certificate) are coordinated and stored by the user without much effort.
- Quick Response Code (QR-Code), an exceptionally powerful and quick decipherable innovation is utilized for Storing and examining different subtle elements of a vehicle.
- We create a website through which user can generate their unique QR-code for their vehicle or can update the details of the vehicles.

2.3 Feasibility Study:

2.3.1 Technical Feasibility:

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, now, not too many detailed designs of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. Several issues have to be considered while doing a technical analysis.

i) Understand the different technologies involved in the proposed system:

Before commencing the project, we must be very clear about what are the technologies that are to be required for the development of the new system.

ii) Find out whether the organization currently possesses the required technologies:

- Is the required technology available with the organization?
- If so, the capacity enough? For instance --“Will the current printer be able to handle the new reports and forms required for the new system?”

2.3.2 Operational Feasibility:

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

- Is there enough support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
- Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
- Had the user been involved in the planning and development of the project?
- Early involvement reduces the chances of resistance to the system and in
- General and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was operational feasible.

2.3.3 Economic Feasibility:

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decisions making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 Study of the System:

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows:

Administrator Interface Design.

User Interface.

Security Authentication.

3.2 NUMBER OF MODULES:

The system after careful analysis has been identified to be presented with the following modules:

The Modules involved are

- 1.Authentication and Security Module
- 2.Users Module
- 3.Admin Module

3.3 MODULES DESCRIPTION:

1. Authentication and Security Module

The user details should be verified against the details in the user tables and if it is valid user, they should be entered into the system. Once entered, based on the user type access to the different modules to be enabled / disabled.

2. User Module

We develop another website (User website) which is used by user to view and update the personal details of the user.

- when i click on login button another webpage opens to view or update the user personal details

- when i click on personal details button another webpage opens to update the user personal details
- when clicked on this button data is updated and redirected to homepage

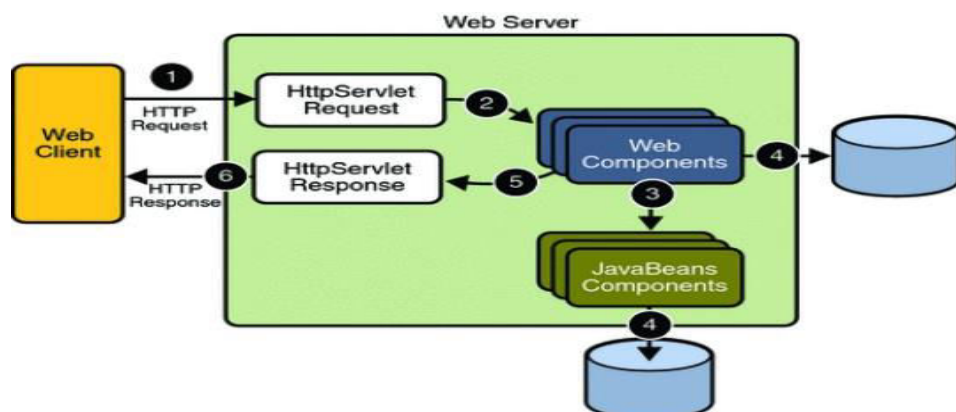
3. Admin Module

We develop a website (Authority website) to generate a QR-code where, this qr-code is integrated with the details of the vehicle like driving license, registration certificate, insurance, pollution under control certificate

- when i click on new account button another web page opens to upload the user details
- when i click on personal details button another web page opens to upload the user personal details
- when clicked on save button it redirects to home page
- when i click on vehicle details button another web page opens to upload the user vehicle details
- upload image are hyperlinks to upload corresponding images from the file system
- when clicked on generate button a QR code generated and it asks to print the code
- when clicked on print button a QR code is printed
- when i click on existing account button another web page opens to upload the ONLY VEHICAL details
- when i click on login button another web page opens to update the user vehicle details
- when clicked on save button it must open another web page to generate a QR-code

3.4 Process Flow:

ARCHITECTURE DIAGRAM:



1. THE PRESENTATION LAYER

Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

2. THE BUSINESS RULES LAYER

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process

3. THE DATA ACCESS LAYER

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

4. THE DATABASE LAYER

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc. In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

3.5 SDLC METHODOLOGIES:

This document plays a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by **Barry Boehm** in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing several users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
 2. Defining the requirements of the second prototype.
 3. Planning an designing the second prototype.
 4. Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.

- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

The following diagram shows how a spiral model act like

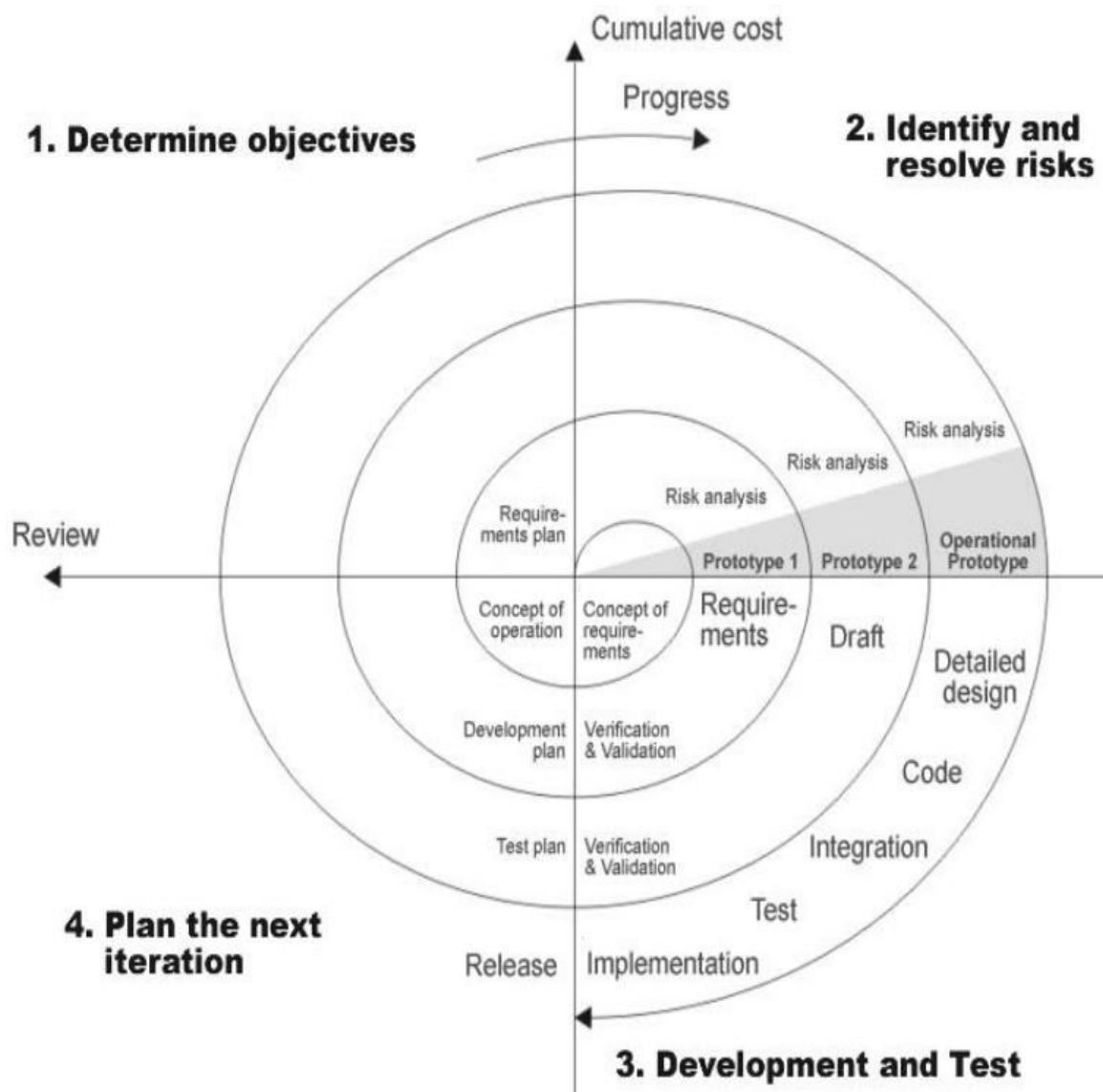


Fig: Spiral Model

ADVANTAGES

- Estimates (i.e. budget, schedule etc) become more realistic as work progresses, because important issues discovered earlier.
- It is more able to cope with the changes that are software development generally entails.
- Software engineers can get their hands in and start working on the core of a project earlier.

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 DFD:

Data Flow Diagram

A graphical tool used to describe and analyse the movement of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart. DFD's are the model of the proposed system. They clearly should show the requirements on which the new system should be built.

The Basic Notation used to create a DFD's are as follows:

1. Dataflow: Data move in a specific direction from an origin to a destination.



2. Process: People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

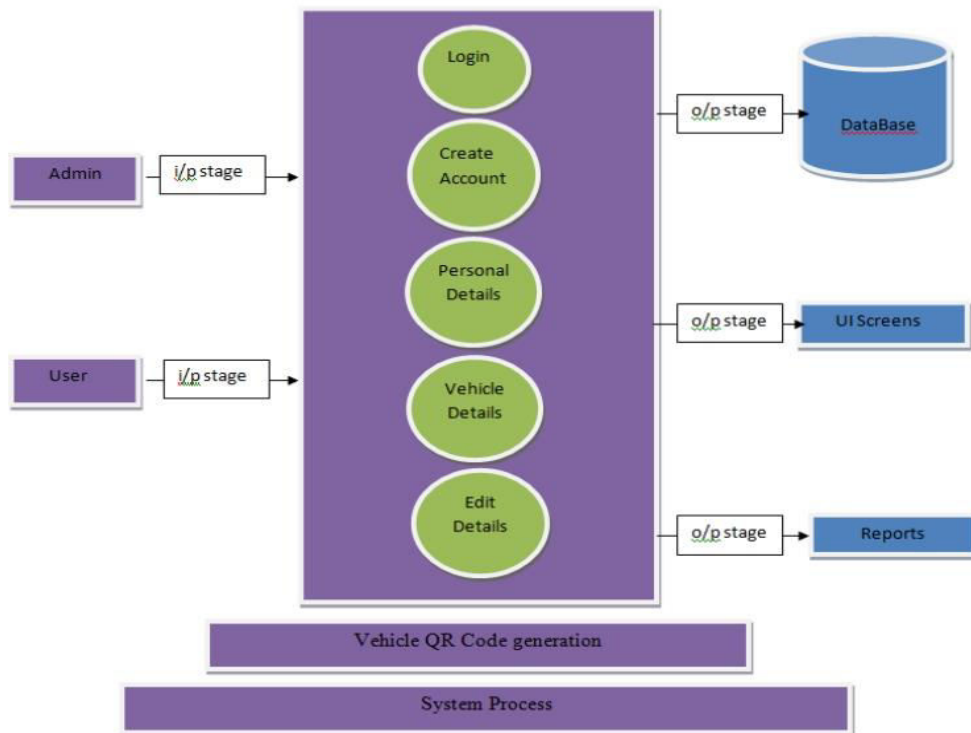


3. Source: External sources or destination of data, which may be People, programs, organizations or other entities.

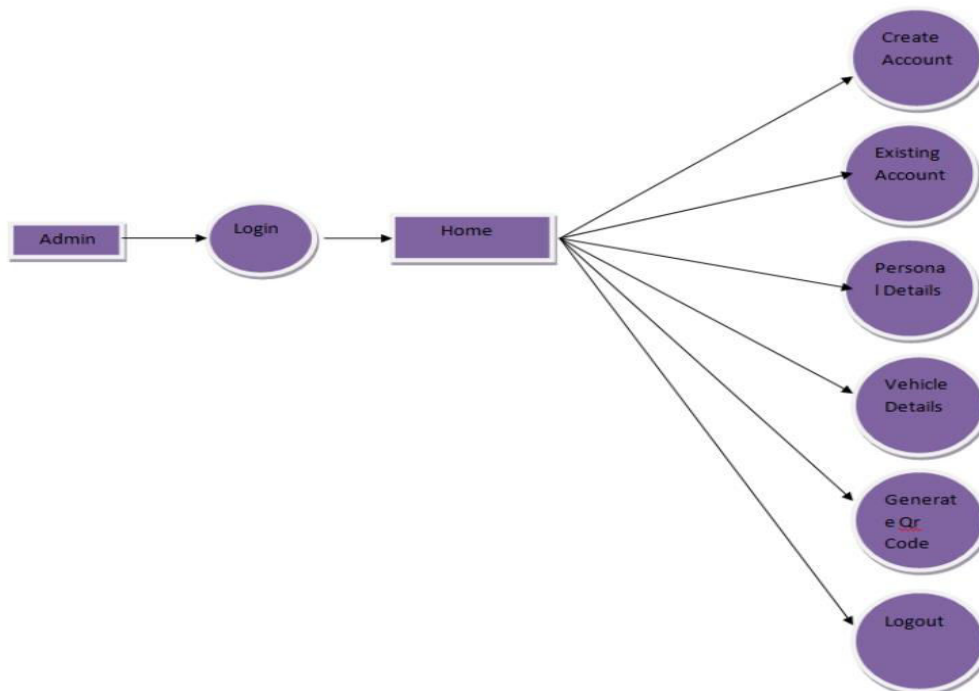


4. Data Store: Here data are stored or referenced by a process in the System.

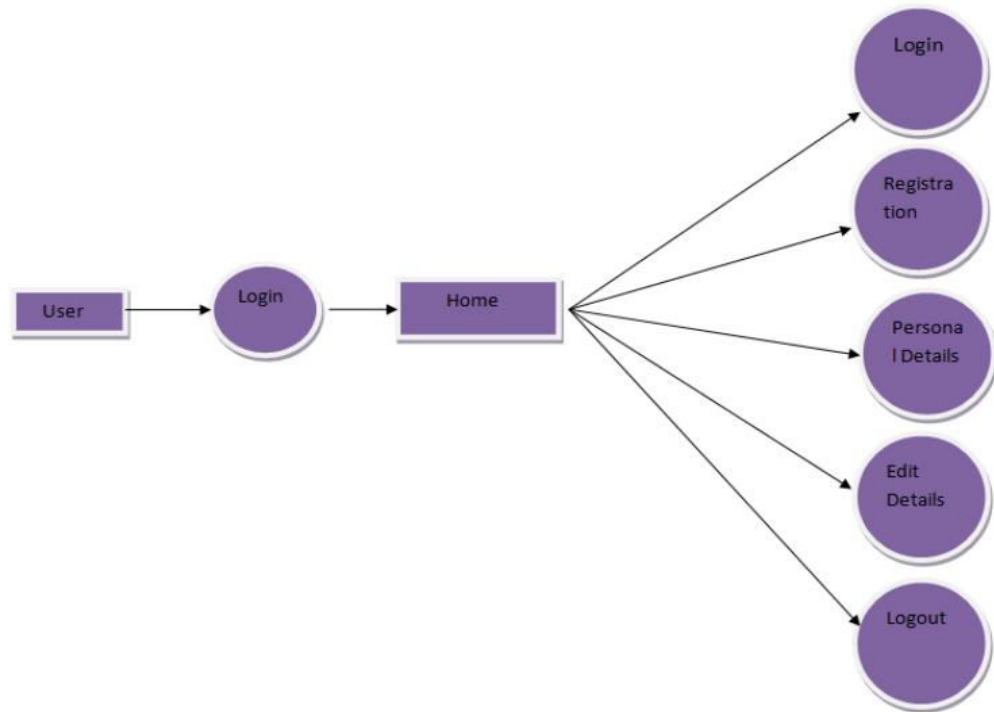
Context Level Data Flow Diagram



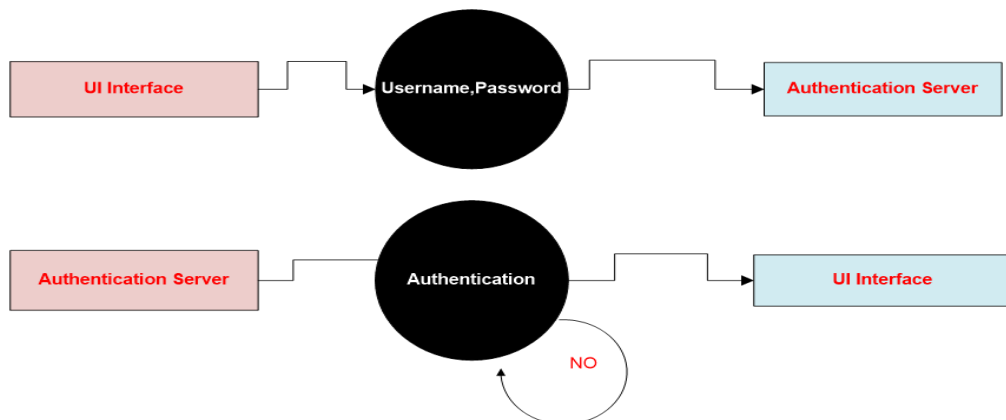
Admin Level 1 Data Flow Diagram



User Level 1 Data Flow Diagram



Authentication DFD



FLOW CHART

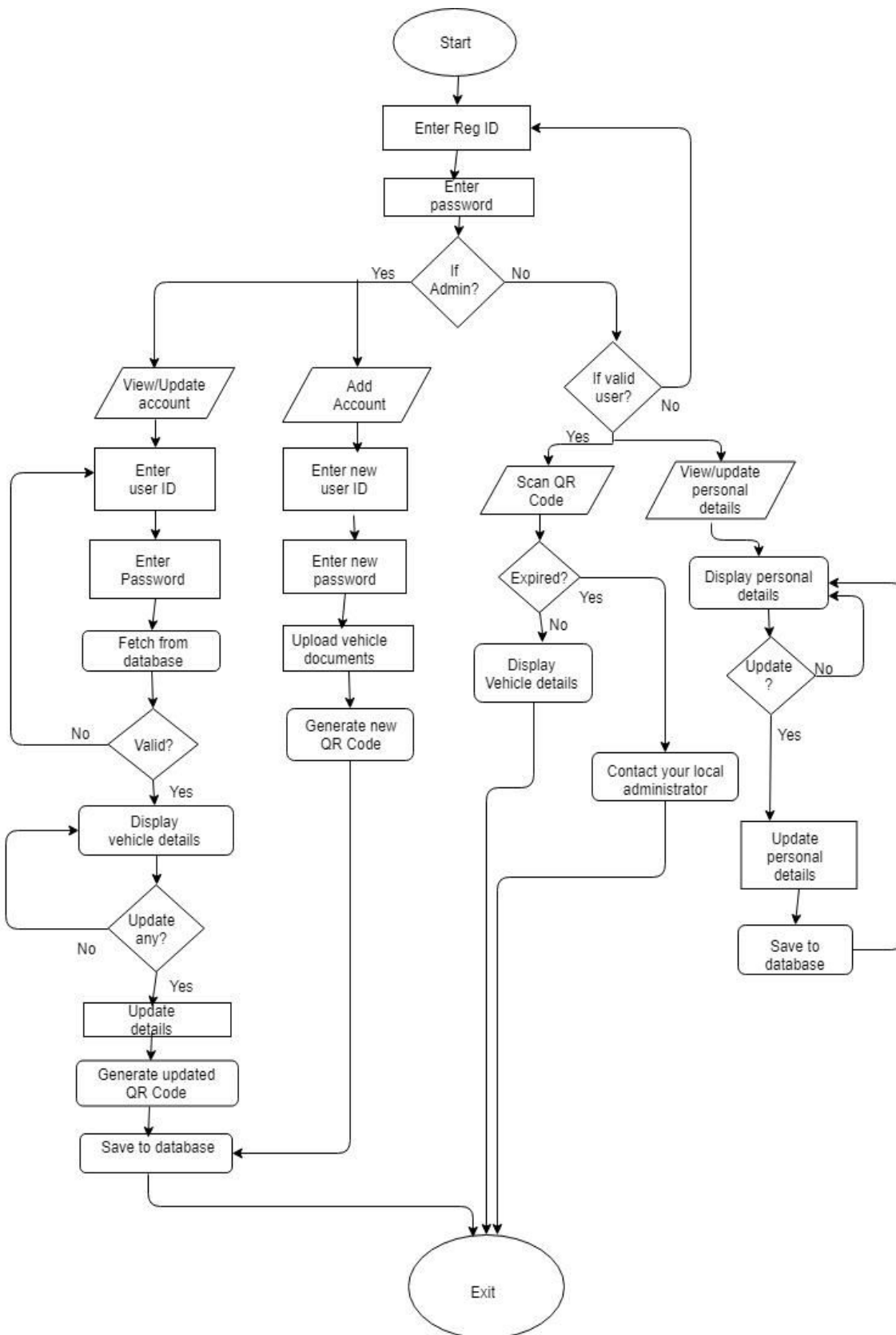


Fig-1: Flowchart

4.2 UML DIAGRAMS

UNIFIED MODELING LANGUAGE DIAGRAMS

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

USER MODEL VIEW

This view represents the system from the user perspective.

The analysis representation describes a usage scenario from the end-user perspective.

STRUCTURAL MODEL VIEW

In this model the data and functionality are arrived from inside the system.

This model view models the static structures.

BEHAVIORAL MODEL VIEW

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

IMPLEMENTATION MODEL VIEW

In this the structural and behavioural as parts of the system are represented as they are to be built.

ENVIRONMENTAL MODEL VIEW

- In this structural and behavioural aspects of the environment in which the system is to be implemented are represented.
- UML is specifically constructed through two different domains they are:
- UML Analysis modeling, which focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioural modeling, implementation modeling and environmental model views.
- Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the

functionality of the system. Use cases focus on the behaviour of the system from external point of view.

- Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer ...etc., or another system like central database.

4.3 E-R Diagram

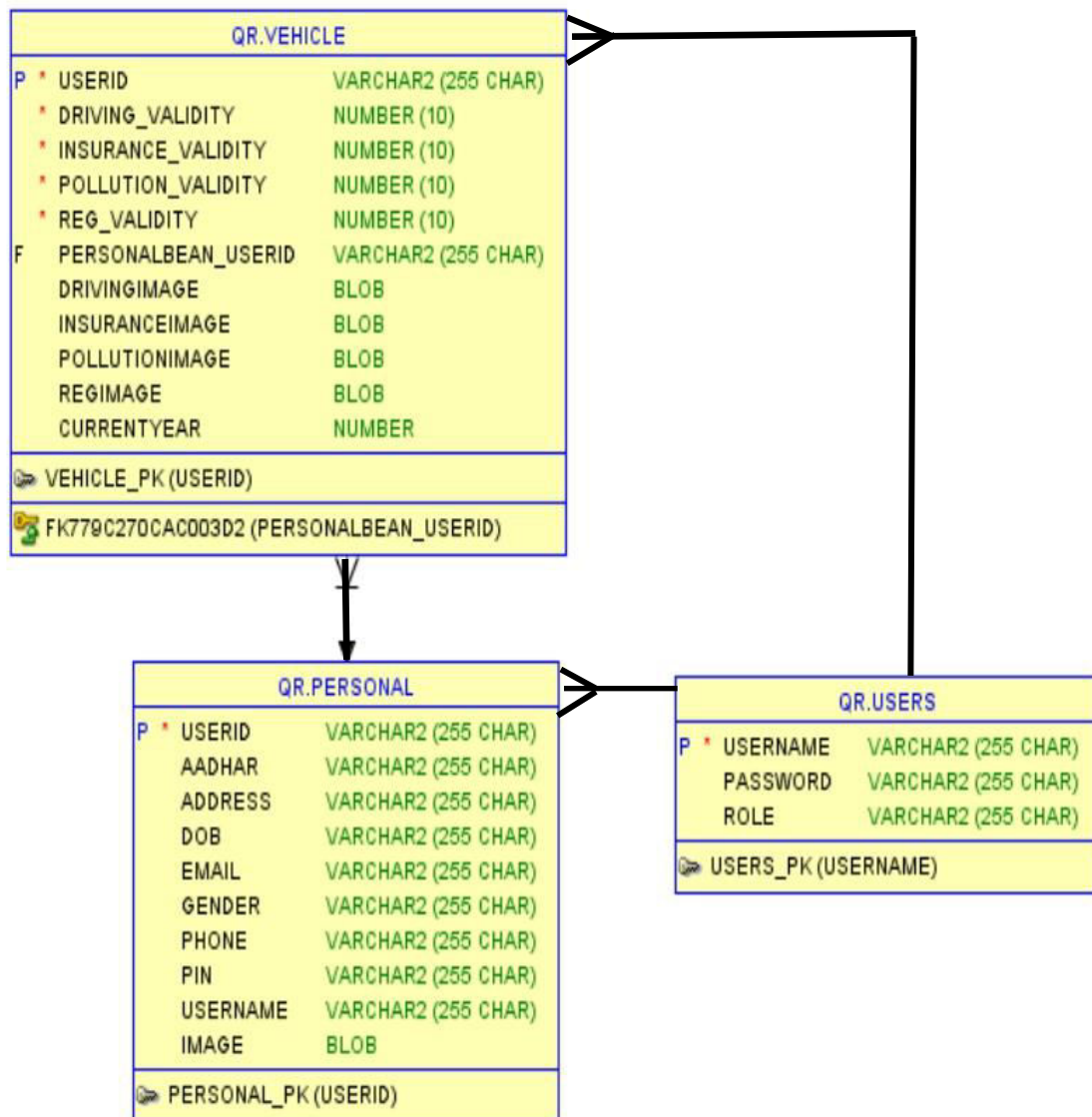


Fig-2: ER Diagram

Class Diagram

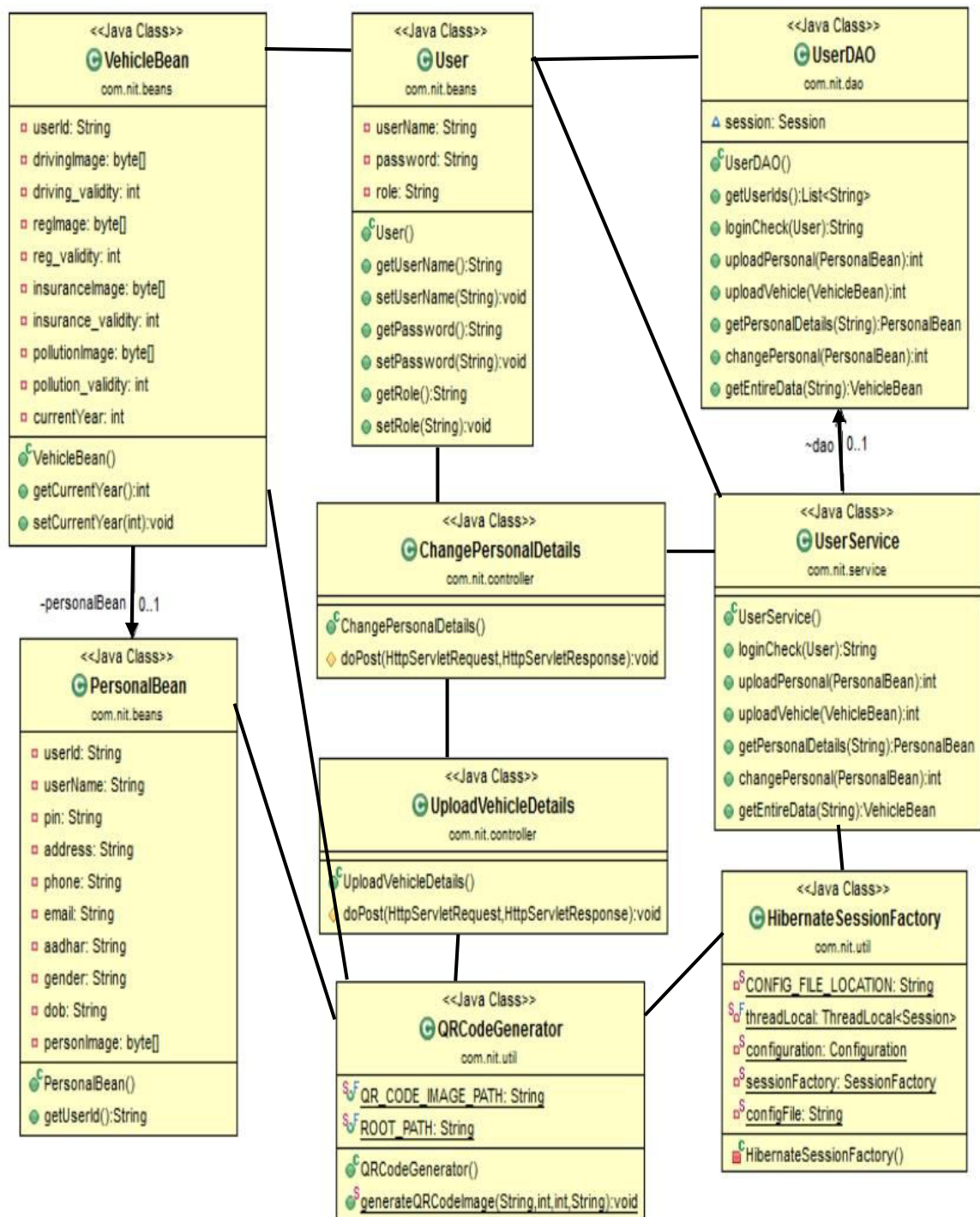


Fig-3: Class Diagram

Use case Diagram

Admin use case diagram:

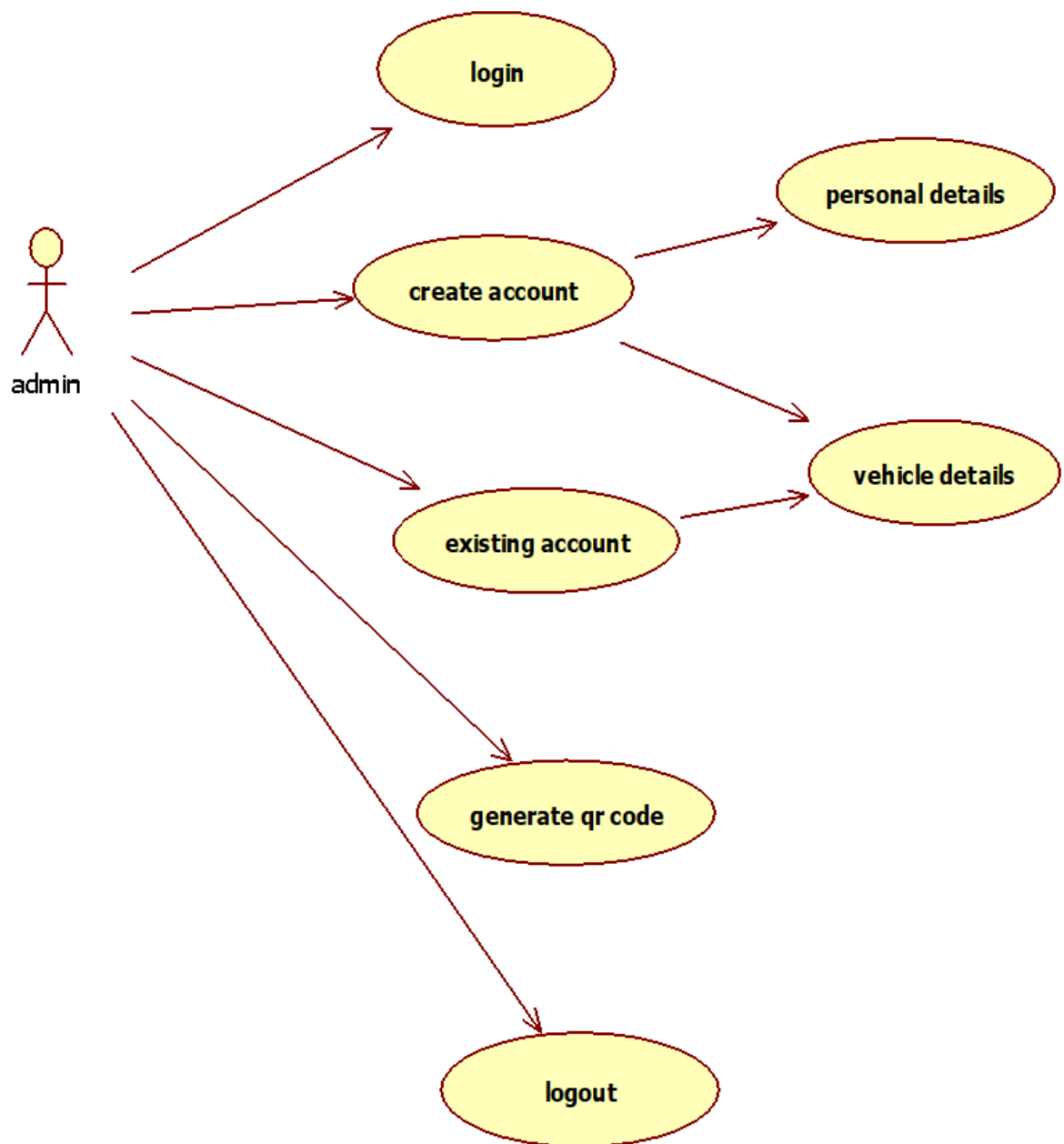


Fig-4: Use case Diagram

User use case diagram:

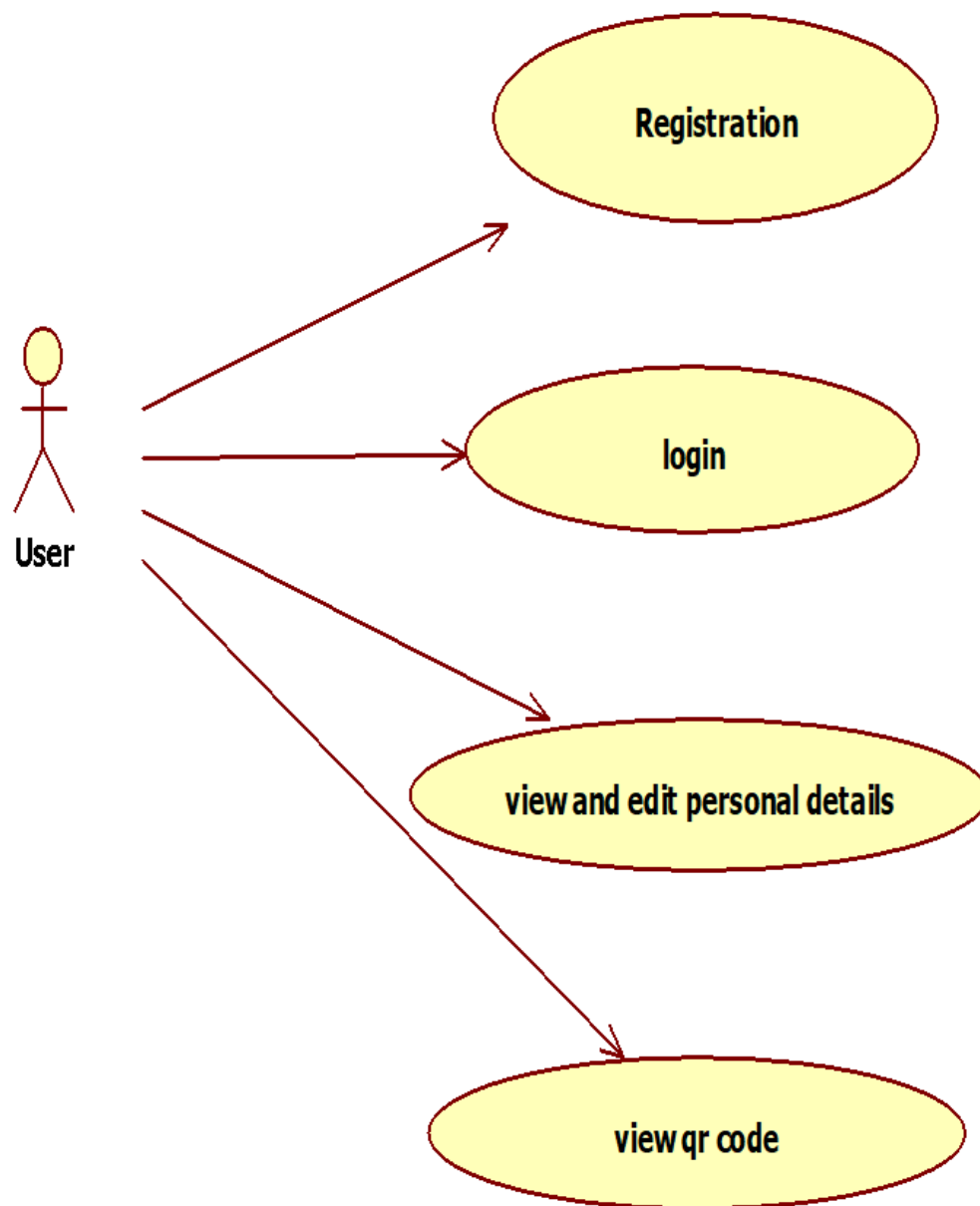


Fig-4: Use case Diagram

Sequence Diagrams

Admin Sequence:

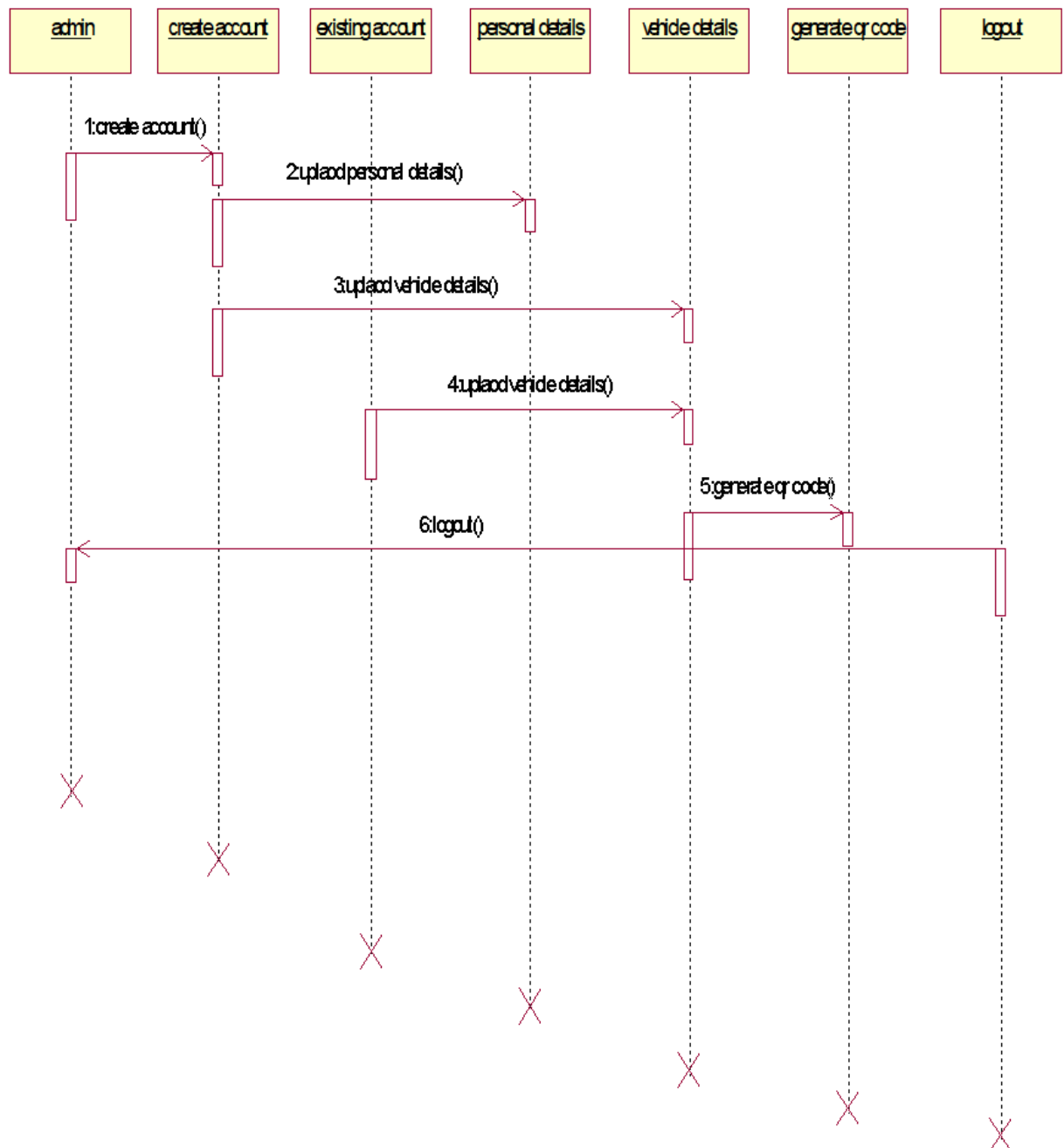


Fig-5: Sequence Diagram

User Sequence:

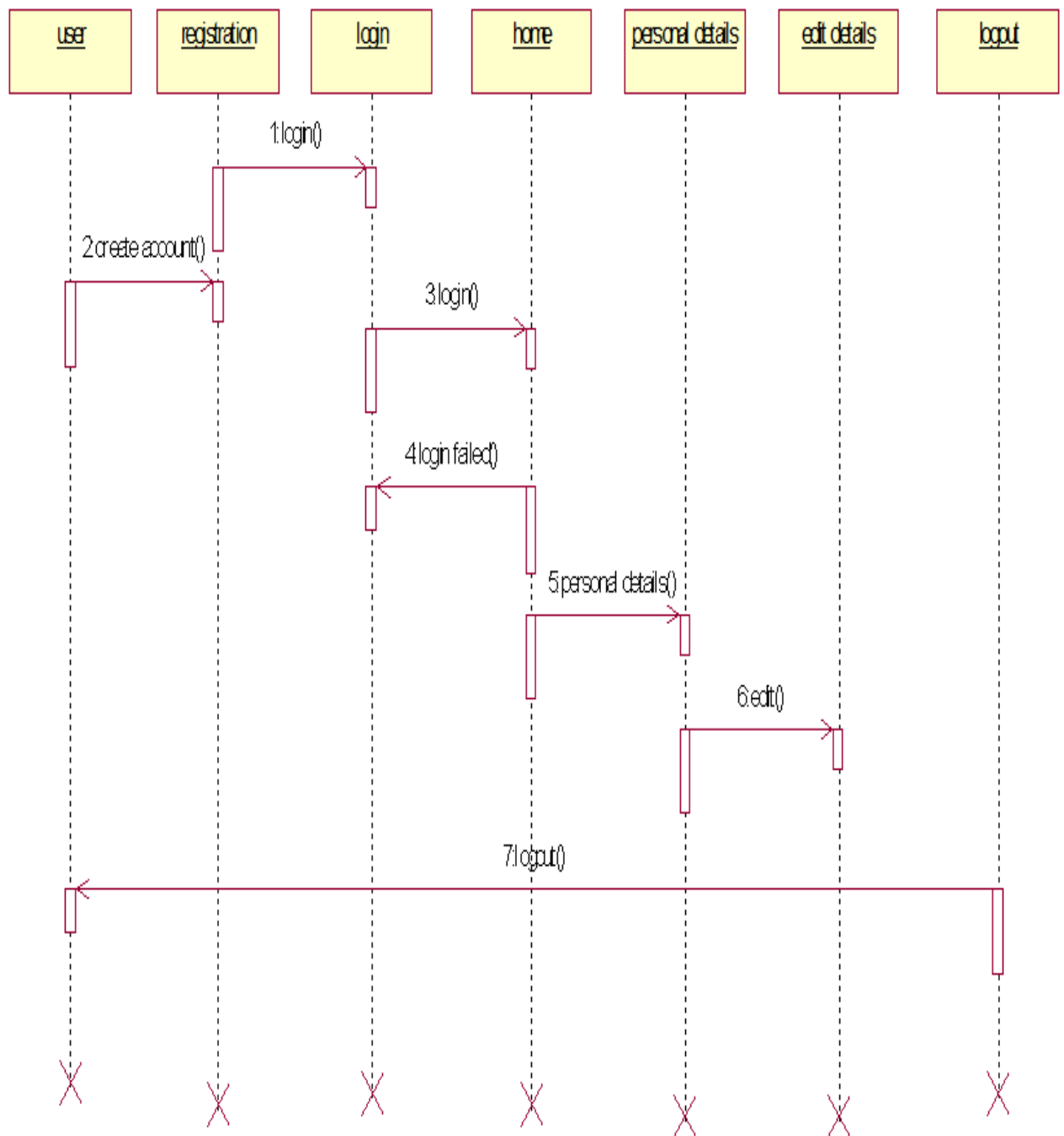


Fig-5: Sequence Diagram

Collaboration Diagram

Admin collaboration:

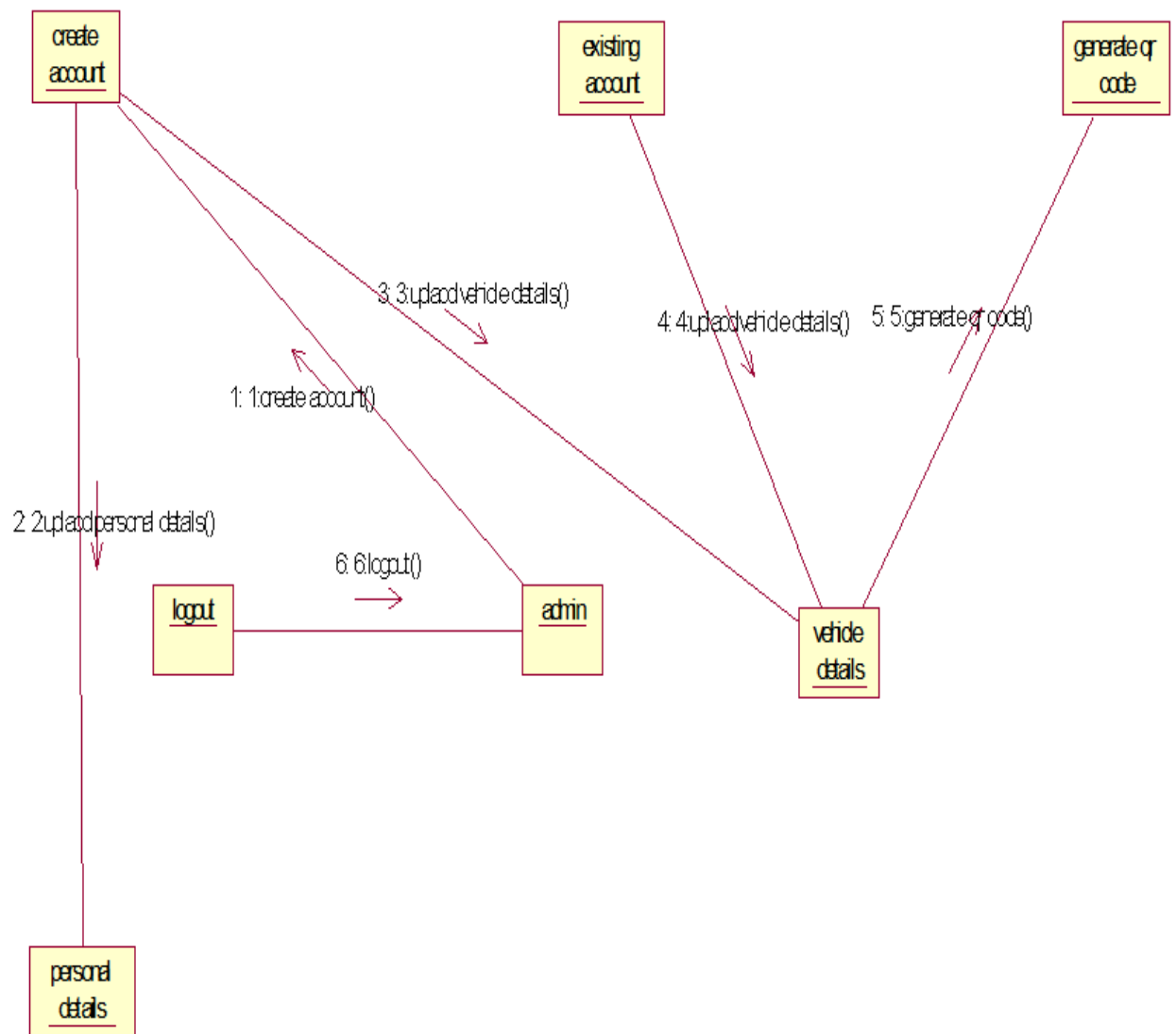


Fig-6: Collaboration Diagram

User collaboration:

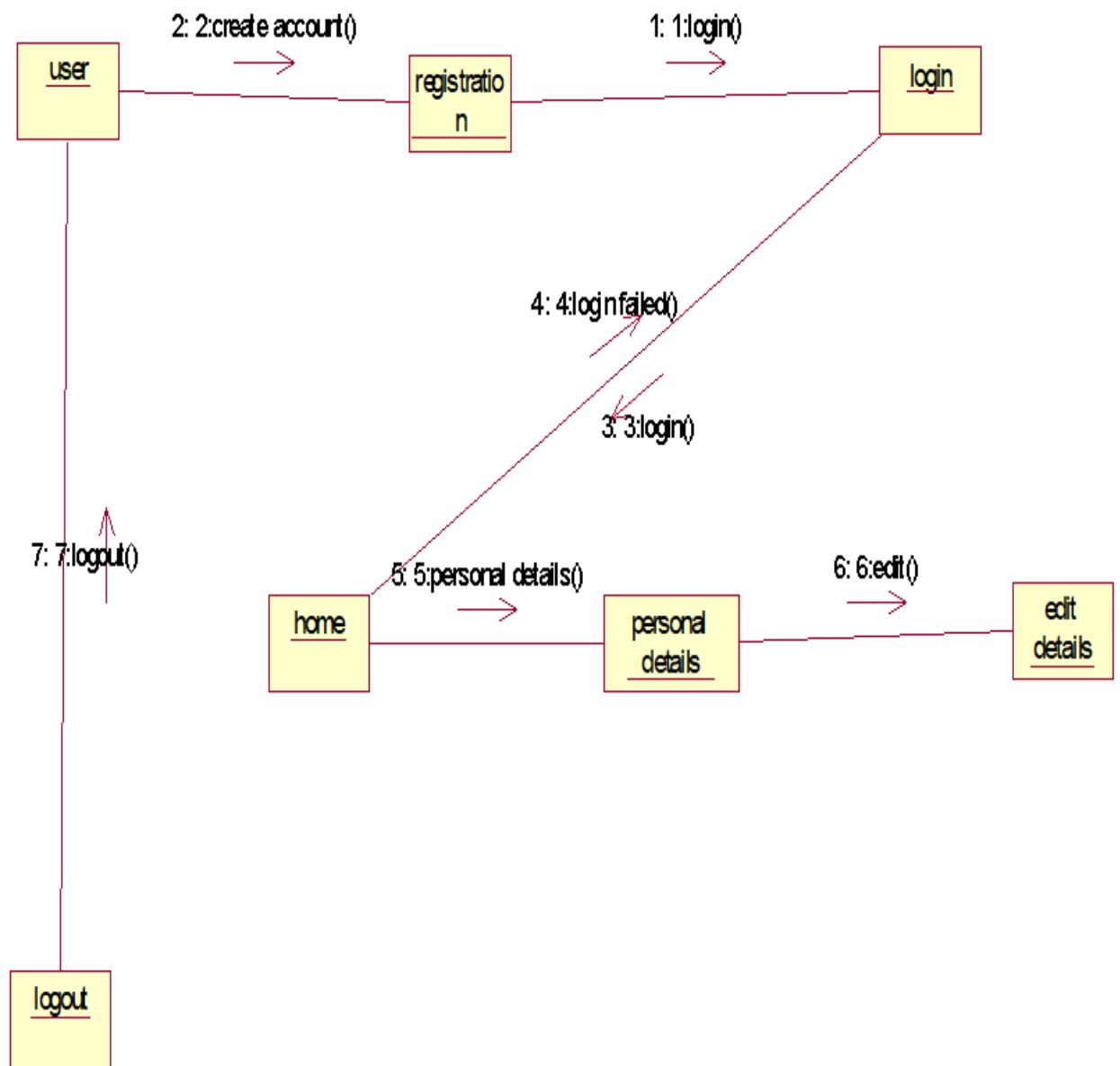


Fig-6: Collaboration Diagram

State Chart Diagram

Admin state chart:

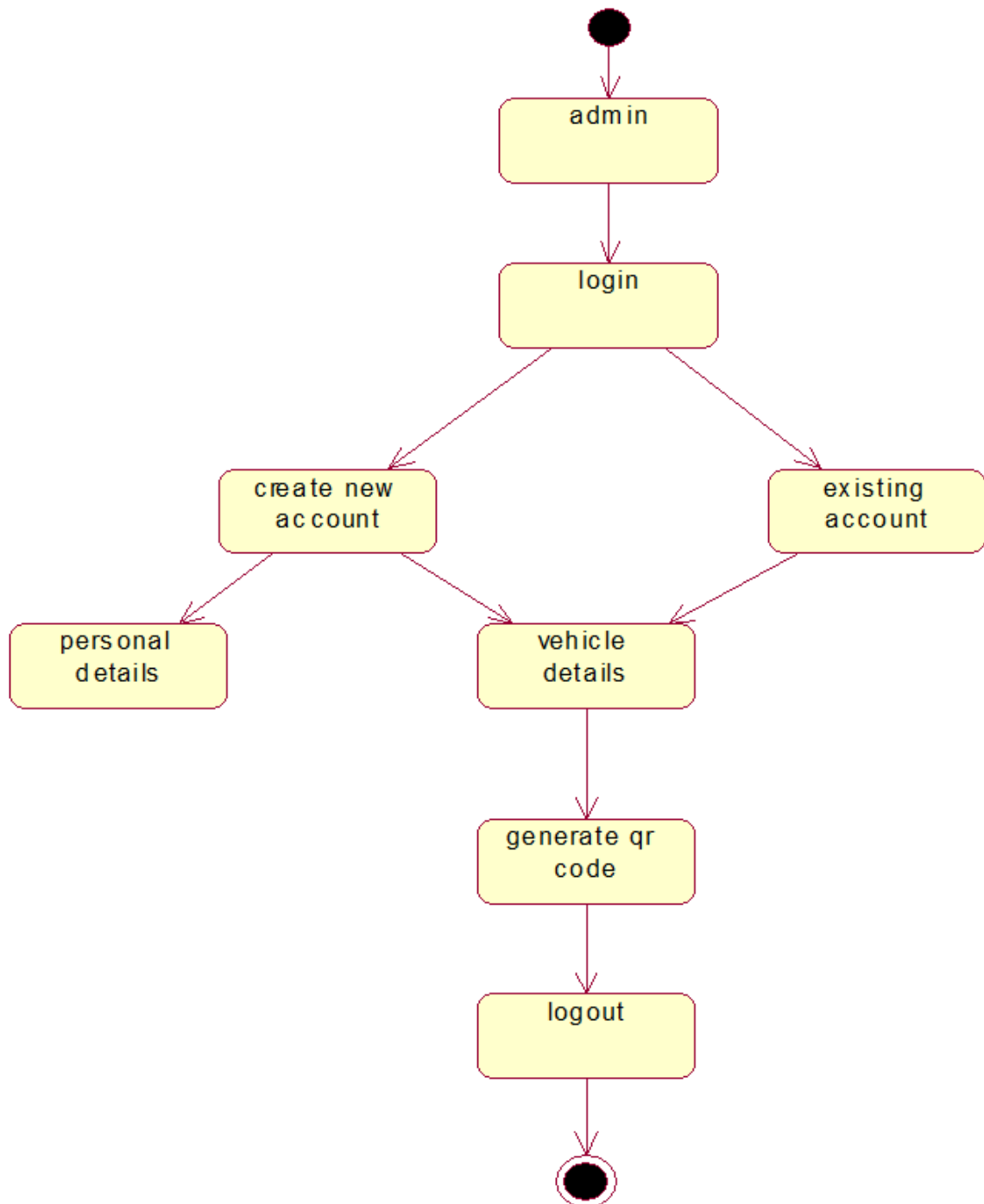


Fig-7: State Chart Diagram

User State chart:

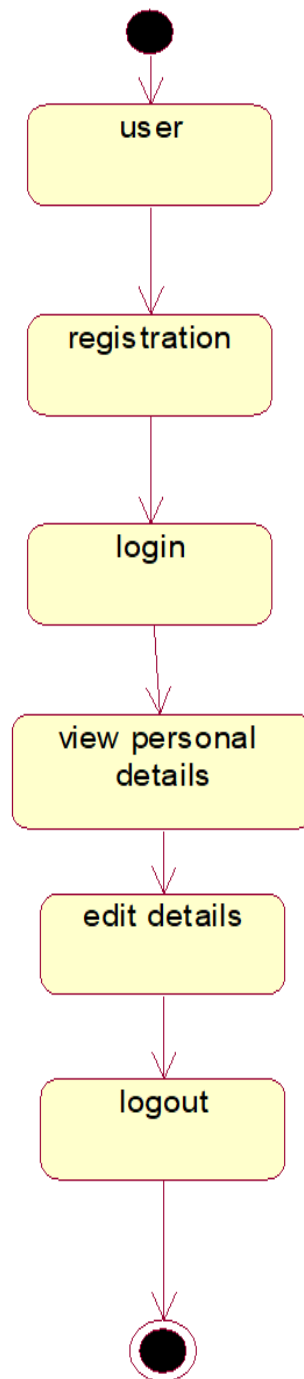


Fig-7: State Chart Diagram

Activity Diagram

Admin activity:

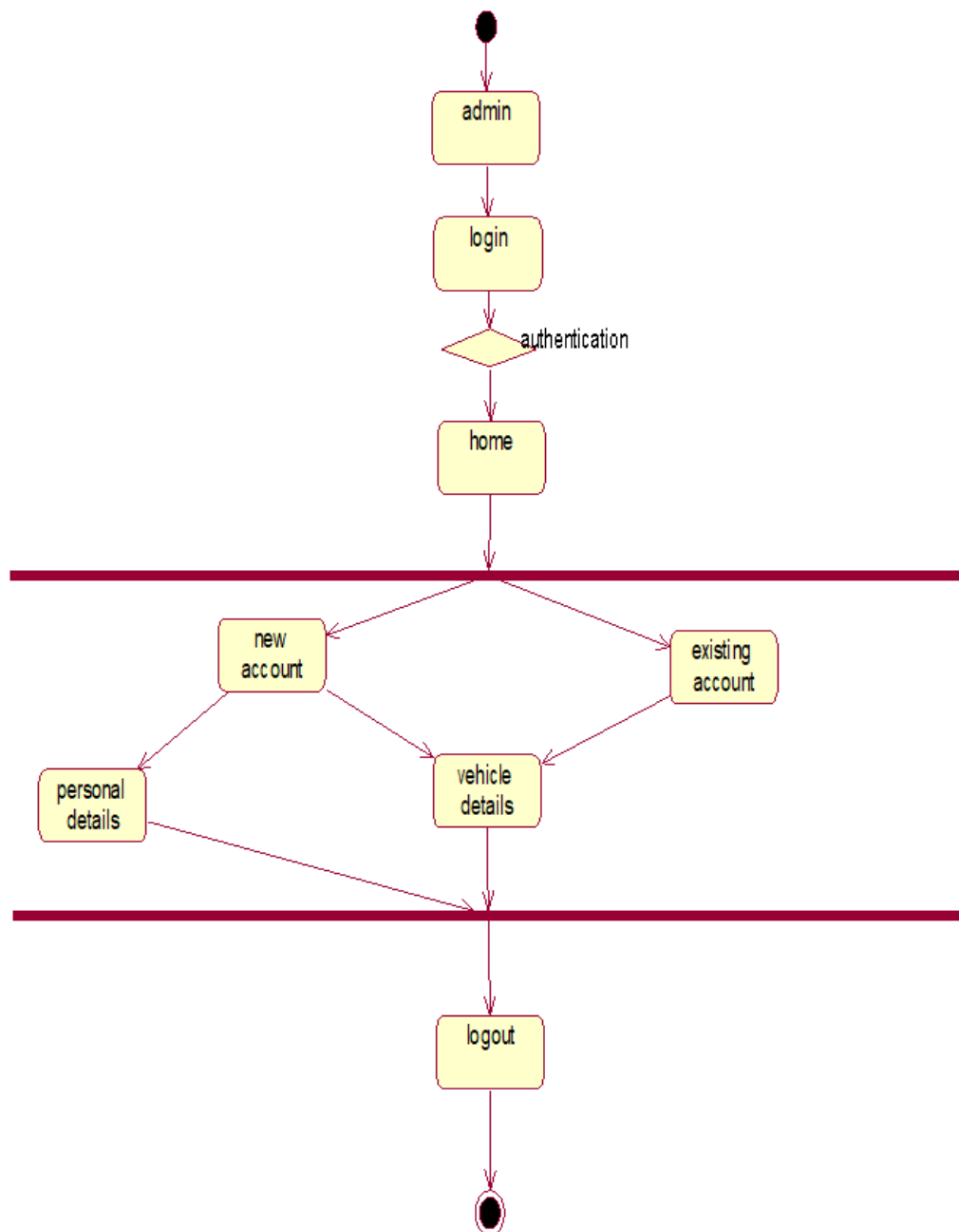


Fig-8: Activity Diagram

User activity:

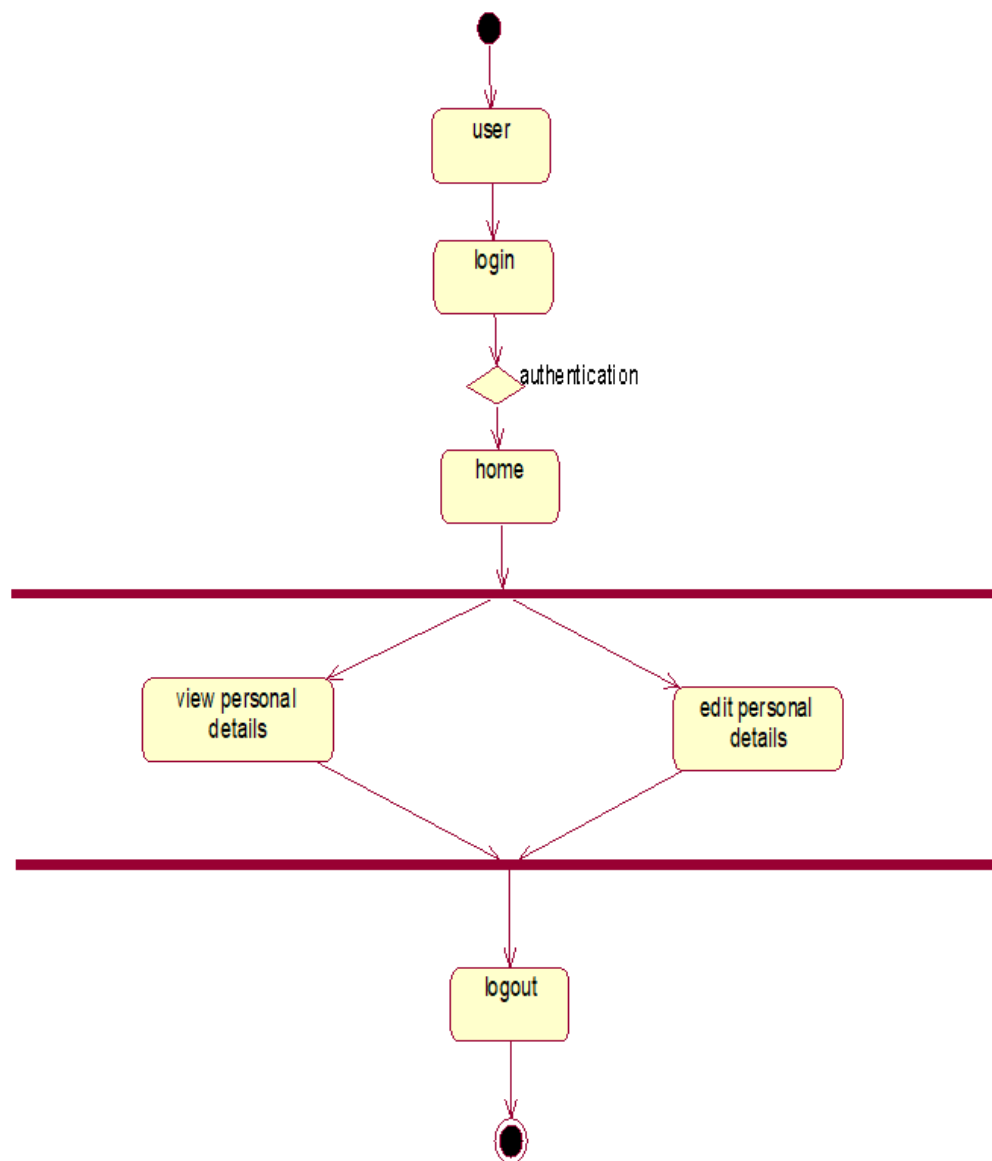


Fig-8: Activity Diagram

Component Diagram

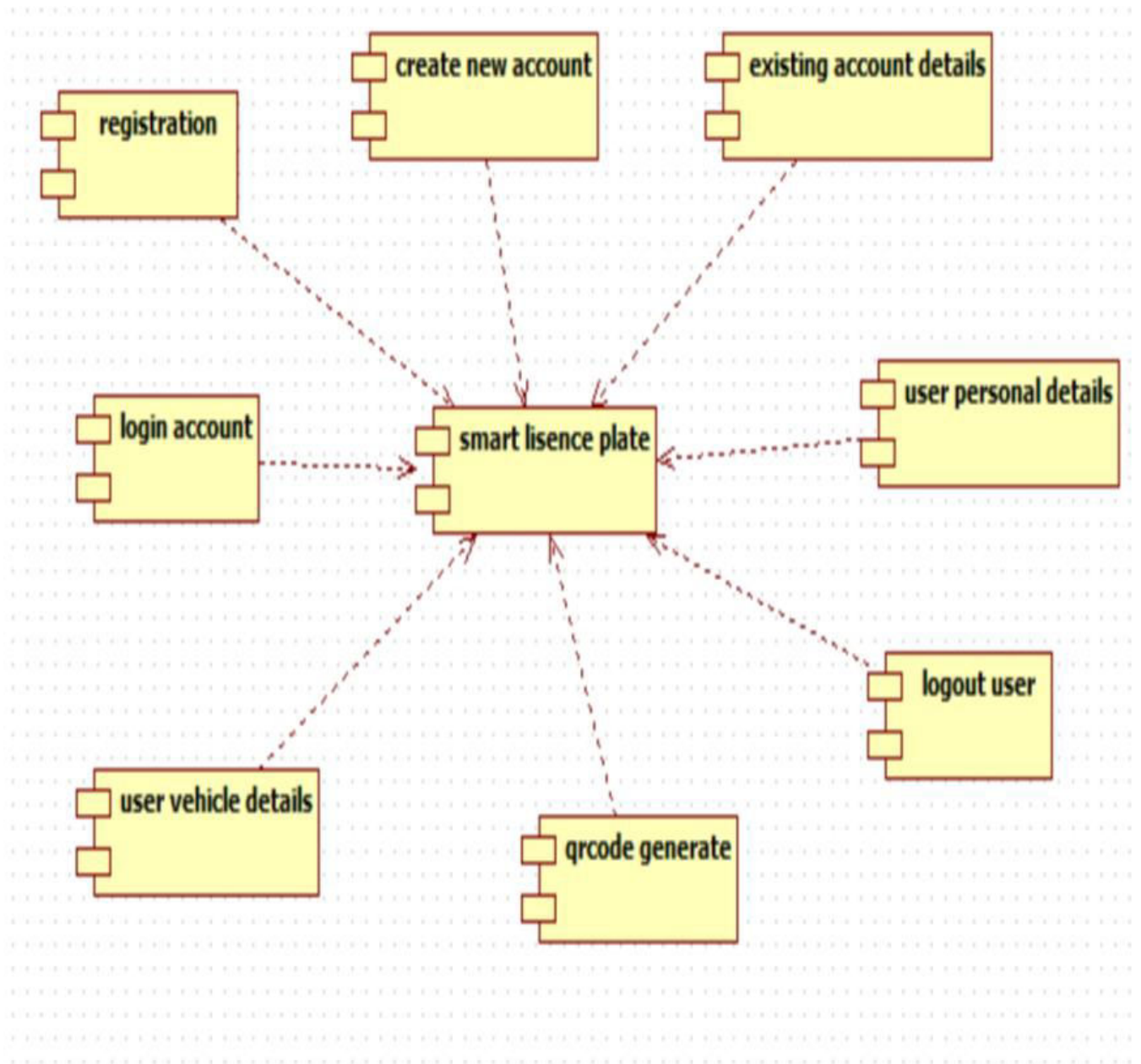


Fig-9: Component Diagram

Deployment Diagram:

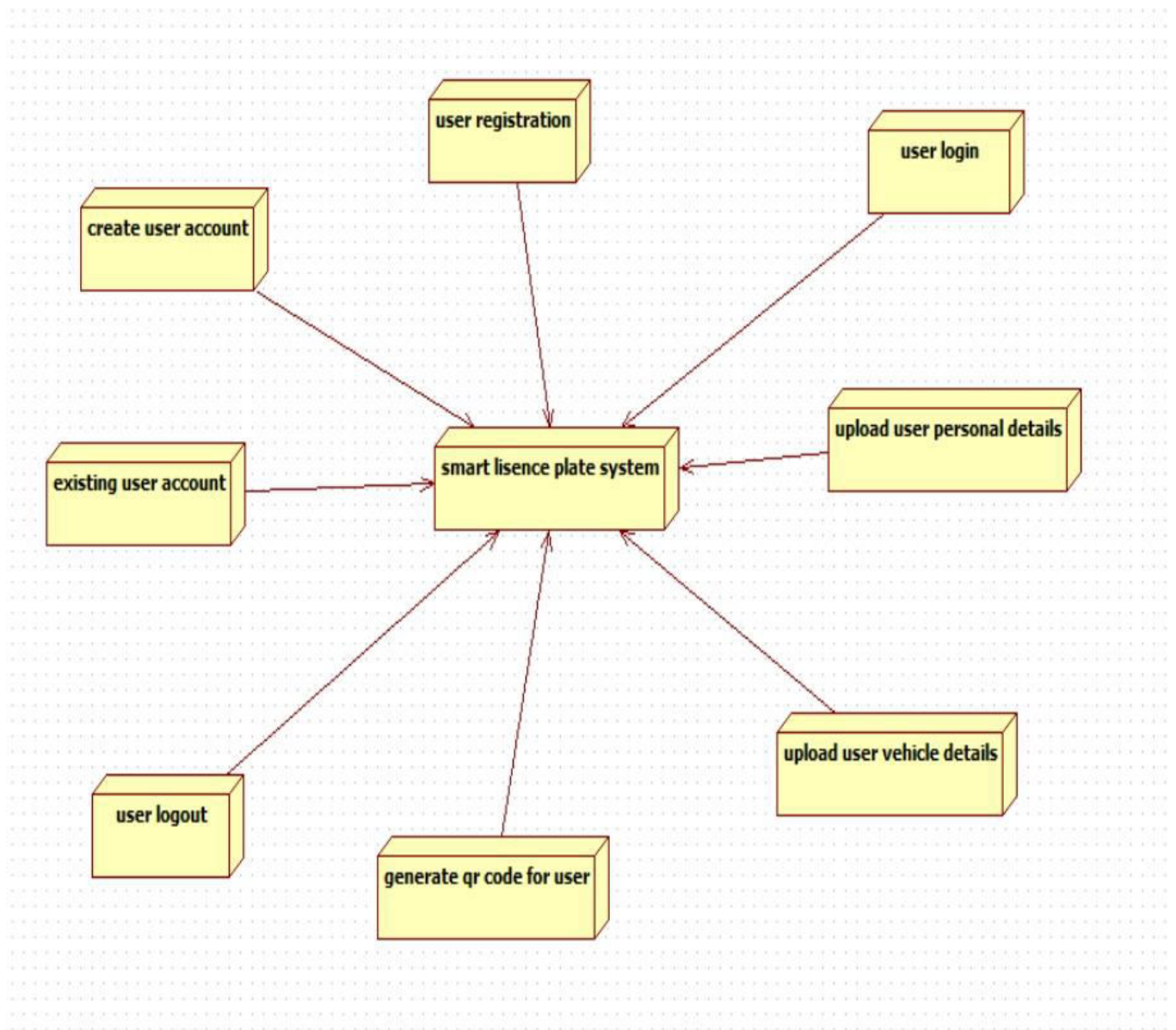


Fig-10: Deployment Diagram

TECHNOLOGY DESCRIPTION

5. TECHNOLOGY DESCRIPTION

5.1 HTML

Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document by denoting certain text as headings, paragraphs, lists, and so on and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behaviour of web browsers and other HTML processors.

Basic HTML Tags:

Tag	Description
<! -- -->	Specifies comments
<A>.....	Creates hypertext links
.....	Formats text as bold
<BIG>.....</BIG>	Formats text in large font.
<BODY>...</BODY>	Contains all tags and text in the HTML document
<CENTER>...</CENTER>	Creates text
<DD>...</DD>	Definition of a term
<DL>...</DL>	Creates definition list
...	Formats text with a particular font
<FORM>...</FORM>	Encloses a fill-out form
<FRAME>...</FRAME>	Defines a particular frame in a set of frames
<H#>...</H#>	Creates headings of different levels (1 – 6)
<HEAD>...</HEAD>	Contains tags that specify information about a document
<HR>...</HR>	Creates a horizontal rule
<HTML>...</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>...</SCRIPT>	Contains client-side or server-side script
<TABLE>...</TABLE>	Creates a table

Advantages

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

5.2 JavaScript:

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser's display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client-side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags.

```
<SCRIPTS>...</SCRIPT>.  
<SCRIPT LANGUAGE = "JavaScript">  
    JavaScript statements  
</SCRIPT>
```

Here are a few things we can do with JavaScript:

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

Advantages

- JavaScript can be used for Sever-side and Client-side scripting.
- It is more flexible than VBScript.
- JavaScript is the default scripting languages at Client-side since all the browsers supports it

5.3 Java Technology:

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.
- Finally, Java is to Internet programming where C was to system programming.

Features of Java Security:

Portability

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed .

The Bytecode

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Java Virtual Machine (JVM)

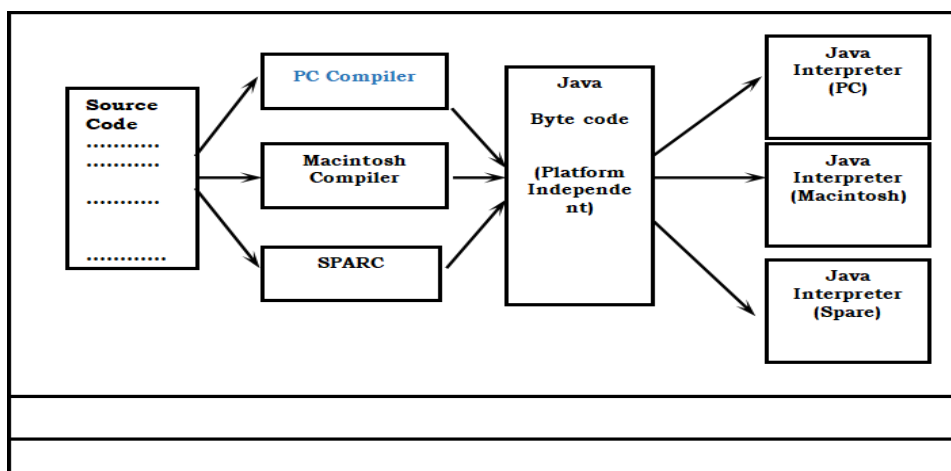
Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a

web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So bytecode verification is integral to the compiling and executing of Java code.

Java Architecture:

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compiling and interpreting Java Source Code



Simple

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object-oriented features of C++.

Object-Oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean

usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time.

5.4 Java Database Connectivity

What Is JDBC?

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

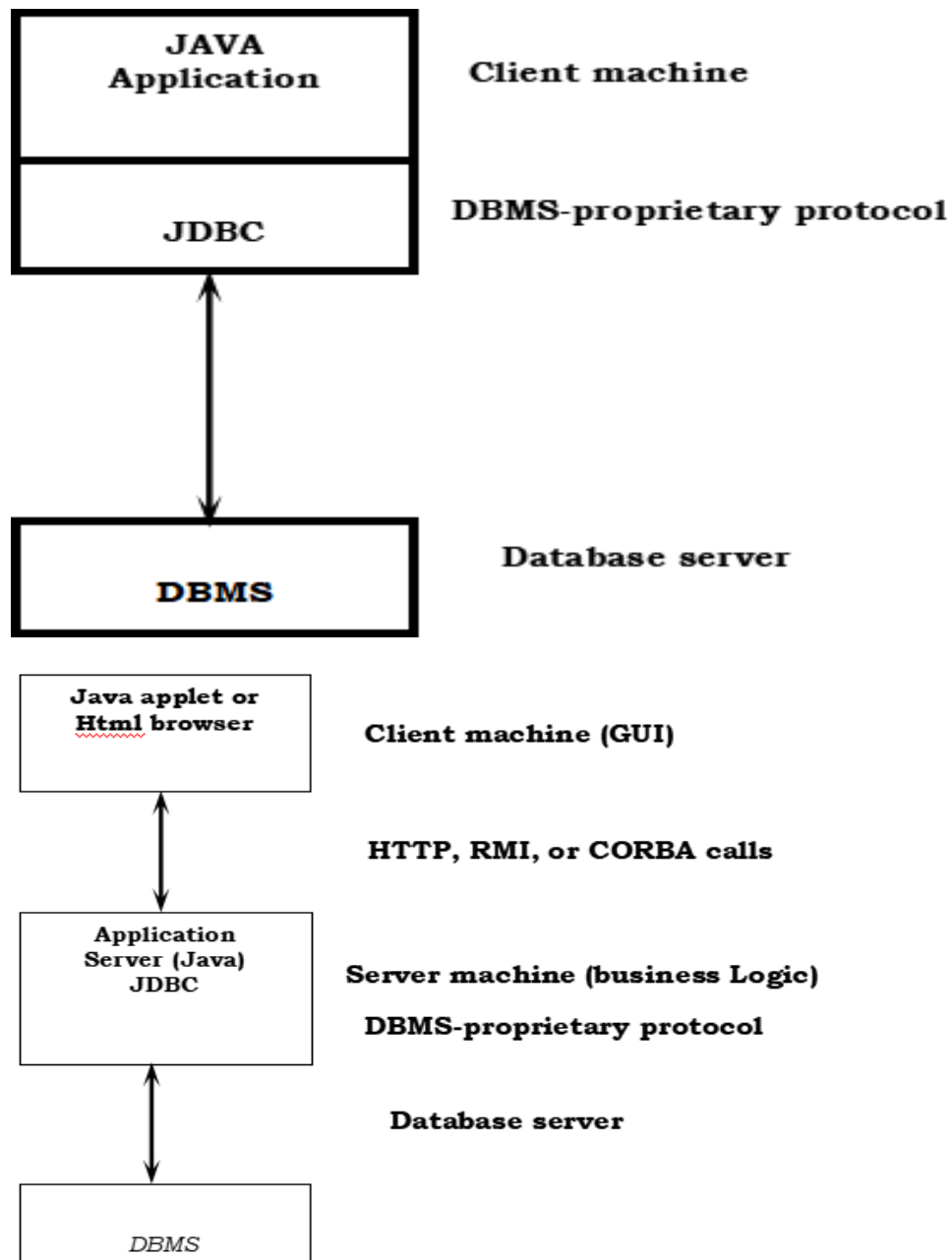
Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

JDBC versus ODBC and other APIs

1. ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
2. A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void *". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.
3. ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.

4. A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

Two-tier and Three-tier Models



JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

- JDBC-ODBC bridge plus ODBC driver
- Native-API partly-Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

JDBC connectivity

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

- Perform connection and authentication to a database server
- Manager transactions
- Move SQL statements to a database engine for pre-processing and execution
- Execute stored procedures
- Inspect and modify the results from Select statements

5.5 Database:

A database management system (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

SQL

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model. In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes).

Data manipulation: Used to manipulate the data within those schema objects (DML Inserting, Updating, Deleting the data, and Querying the Database). A schema is a collection of database objects that can include tables, views, indexes and sequences

List of SQL statements that can be issued against an Oracle database schema are:

SQL Statements	DESCRIPTION
ALTER	Change an existing table, view or index definition
AUDIT	Track the changes made to a table (DDL)
COMMENT	Add a comment to a table or column in a table (DDL)
COMMIT	Make all recent changes permanent
CREATE	Create new database objects such as tables or views
DELETE	Delete rows from a database table (DML)
DROP	Drop a database object such as a table, view or index
GRANT	Allow another user to access database objects
INSERT	Insert new data into a database table (DML)
No AUDIT	Turn off the auditing function (DDL)
REVOKE	Disallow a user access to database objects such as tables
ROLLBACK	Undo any recent changes to the database
SELECT	Retrieve data from a database table (DML)
TRUNCATE	Delete all rows from a database table
UPDATE	Change the values of some data items in a database

5.6 SERVLETS

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces. Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side - object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform independent, dynamically loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-side functionality. For example, an HTTP Servlets can be used to generate dynamic HTML content. When you use Servlets to do dynamic content you get the following advantages:

- They're faster and cleaner than CGI scripts
- They use a standard API (the Servlets API)
- They provide all the advantages of Java (run on a variety of servers without needing to be rewritten).

Advantages of the Servlet API

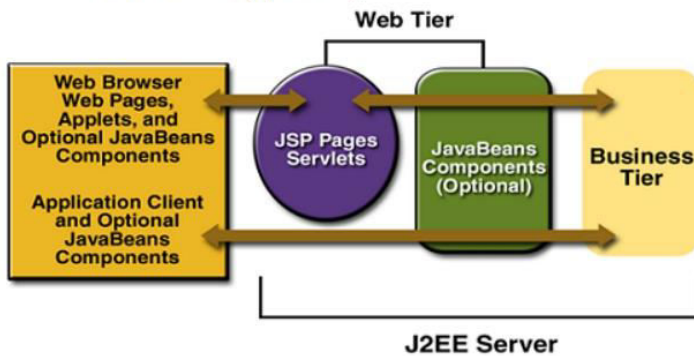
One of the great advantages of the Servlet API is protocol independence. It assumes nothing about:

- The protocol being used to transmit on the net
- How it is loaded
- The server environment it will be running in

These qualities are important, because it allows the Servlet API to be embedded in different kinds of servers. There are other advantages to the Servlet API as well. These include:

- It's extensible - you can inherit all your functionality from the base classes made available to you.
- It's simple, small, and easy to use.

- Web components
 - Servlets or JSP pages
 - JavaBeans (optional)



Loading Servlets:

Servlets can be loaded from three places:

- From a directory that is on the CLASSPATH. The CLASSPATH of the Java Web Server includes service root/classes/ which is where the system classes reside.
- From the <SERVICE_ROOT /Servlets/ directory. This is **not** in the server's class path. A class loader is used to create Servlets from this directory. New Servlets can be added - existing Servlets can be recompiled and the server will notice these changes.
- From a remote location, for this a code base like `http: // nine.eng / classes / foo /` is required in addition to the Servlets class name. Refer to the admin GUI docs on Servlet section to see how to set this up.

Java Server Pages (JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model. The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches; it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work.

5.7 Eclipse IDE

Eclipse is an open-source software framework written primarily in Java. In its default form it is an Integrated Development Environment (IDE) for Java developers, consisting of the Java Development Tools (JDT) and the Eclipse Compiler for Java (ECJ). Users can extend its capabilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Language packs are available for over a dozen languages.

Architecture

The basis for Eclipse is the Rich Client Platform (RCP). The following components constitute the rich client platform:

- ✓ OSGi - a standard bundling framework
- ✓ Core platform - boot Eclipse, run plug-ins
- ✓ the Standard Widget Toolkit (SWT) - a portable widget toolkit
- ✓ JFace - viewer classes to bring model view controller programming to SWT, file buffers, text handling, text editors
- ✓ the Eclipse Workbench - views, editors, perspectives, wizards

TESTING & DEBUGGING TECHNIQUES

6. TESTING & DEBUGGING TECHNIQUES

6.1 Testing Methodologies:

1. Black box testing
2. White box testing
3. Grey box testing

- **Black box Testing** is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.
Usually Test Engineers are involved in the black box testing.
- **White box Testing** is the testing process in which tester can perform testing on an application with having internal structural knowledge.
Usually the Developers are involved in white box testing.
- **Grey Box Testing:** is the process in which the combination of black box and white box tonics are used.

6.2 Levels of Testing:

- Unit Testing.
- Module Testing.
- Integration Testing.
- System Testing.
- User Acceptance Testing.

6.3 Types of Testing:

Smoke Testing: It is the process of initial testing in which tester looks for the availability of all the functionality of the application in order to perform detailed testing on them. (Main check is for available forms)

Sanity Testing: It is a type of testing that is conducted on an application initially to check for the proper behaviour of an application that is to check all the functionality are available before the detailed testing is conducted by on them.

Regression Testing: It is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.

Re-Testing: It is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environment's issues if at all any defects are there.

Static Testing: It is the testing, which is performed on an application when it is not been executed: GUI, Document Testing

Dynamic Testing: It is the testing which is performed on an application when it is being executed: Functional testing.

Alpha Testing: It is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.

Beta-Testing: It is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.

Monkey Testing: It is the process in which abnormal operations, beyond capacity operations are done on the application to check the stability of it in spite of the user's abnormal behaviour.

Compatibility testing: It is the testing process in which usually the products are tested on the environments with different combinations of databases (application servers, browsers...etc) In order to check how far the product is compatible with all these environments platform combination.

Installation Testing: It is the process of testing in which the tester tries to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.

Adhoc Testing: It is the process of testing in which unlike the formal testing where in test case document is used, without that test case document testing can be done of an application, to cover that testing of the future which are not covered in that test case document. Also it is intended to perform GUI testing which may involve the cosmetic issues.

TCD (Test Case Document):

- **Test Scope (or) Test objective**
- **Test Scenario**
- **Test Procedure**
- **Test Case**

Test scope:

- Test coverage is provided for the screen “ Login check” form of an Administration module of Warehouse executer application
- Areas of the application to be tested

Test Scenario:

- When the office personals use this screen for the data entry, calculate the storages stock, confirm payments, discharging products and saving the information on s basis and quit the form.

Test Procedure:

- The procedure for testing this screen is planned in such a way that the data entry, status calculation functionality, saving and quitting operations are tested in terms of GUI testing, Positive testing, Negative testing using the corresponding GUI test cases, Positive test cases, Negative test cases respectively

6.4 SOFTWARE TESTING LIFE CYCLE:

Test Planning:

This document involves the

1. scope of testing
2. Objective of testing
3. Areas that need to be tested
4. Areas that should not be tested
5. Scheduling Resource Planning
6. Areas to be automated, various testing tools used.

Test Development:

1. Test case Development (check list)
2. Test Procedure preparation. (Description of the Test cases).

Result Analysis:

1. Expected value : It is nothing but expected behaviour of application.
2. Actual value : It is nothing but actual behaviour of application
3. Bug Tracking : Collect all the failed cases, prepare documents.
4. Reporting : Prepare document (status of the application)

6.5 TEST CASES:

Test cases can be divided in to two types. First one is Positive test cases and second one is negative test cases. In positive test cases are conducted by the developer intention is to get the output. In negative test cases are conducted by the developer intention is to don't get the output.

+VE TEST CASES

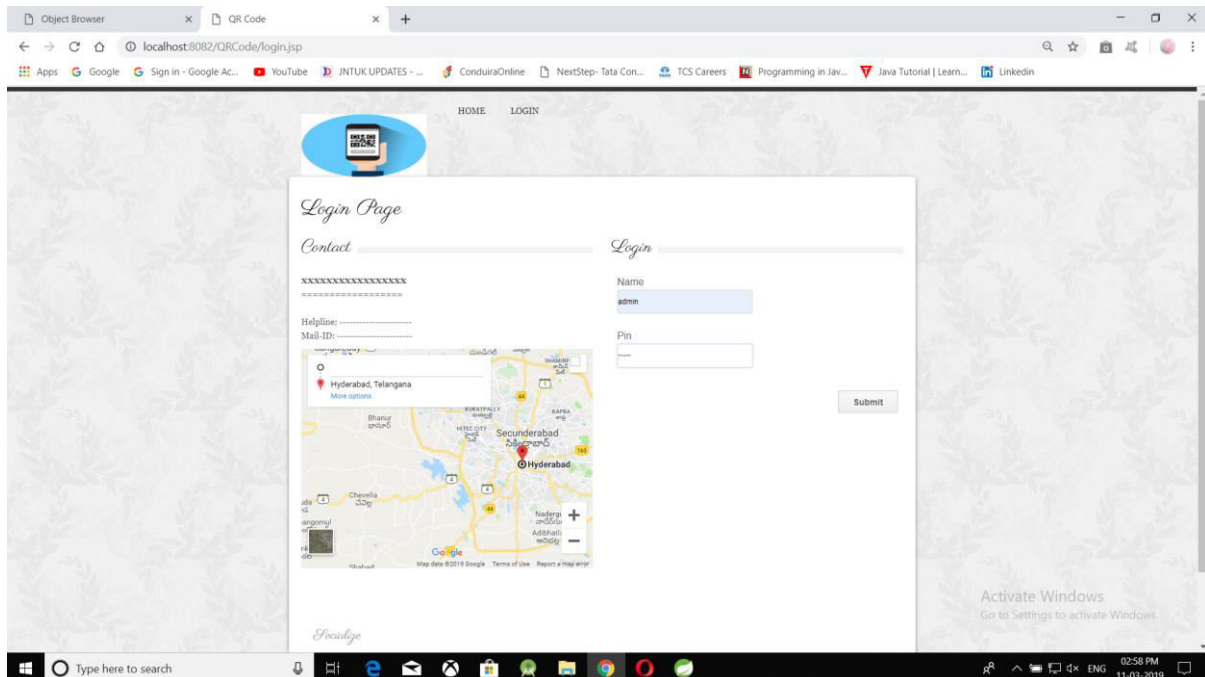
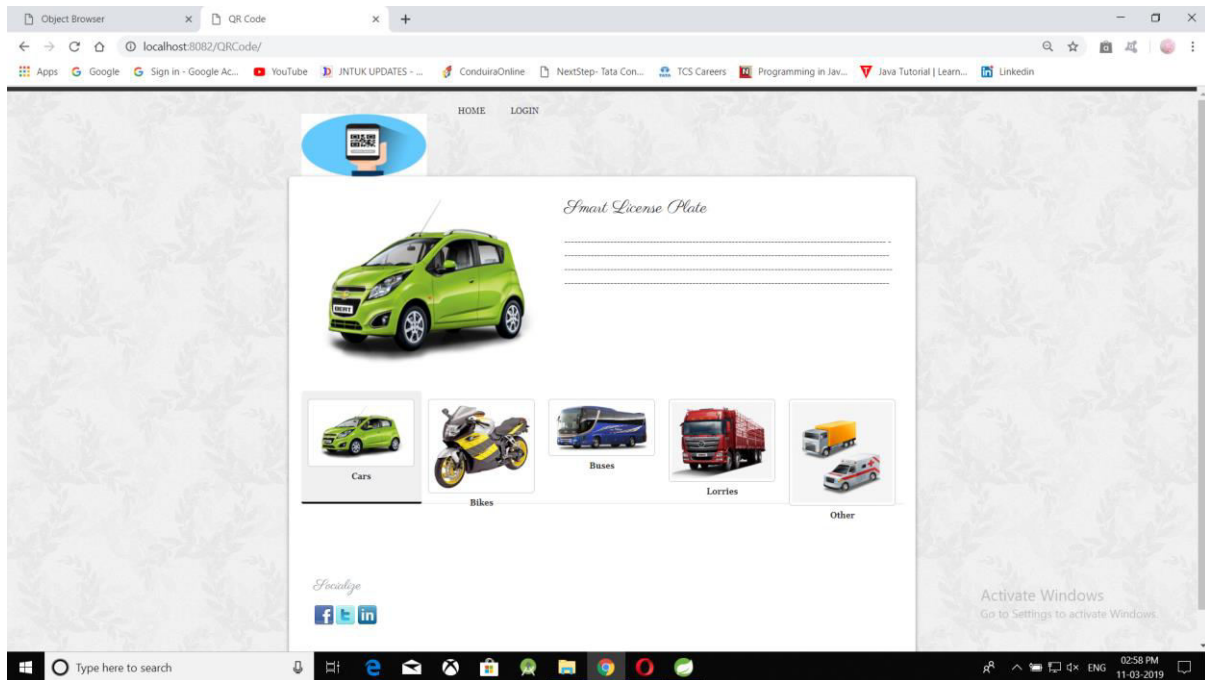
S. No	Test case Description	Actual value	Expected value	Result
1	Create new user registration process	Enter personal info and address info.	Update personal info and address info in to database successfully	True
2	Enter the username and password	Verification of login details.	Login Successfully	True
3	Upload information	Enter all fields	Web data uploaded successfully	True
4	Generate QR code	Retrieve all fields	Store the data in database	True

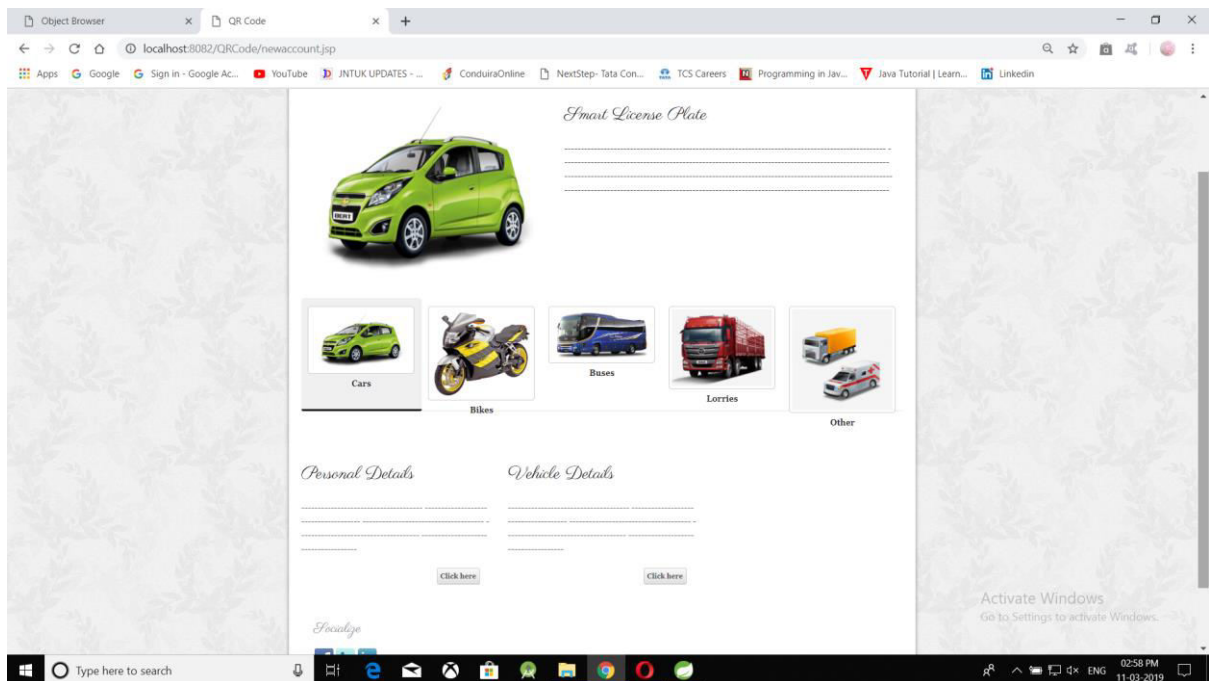
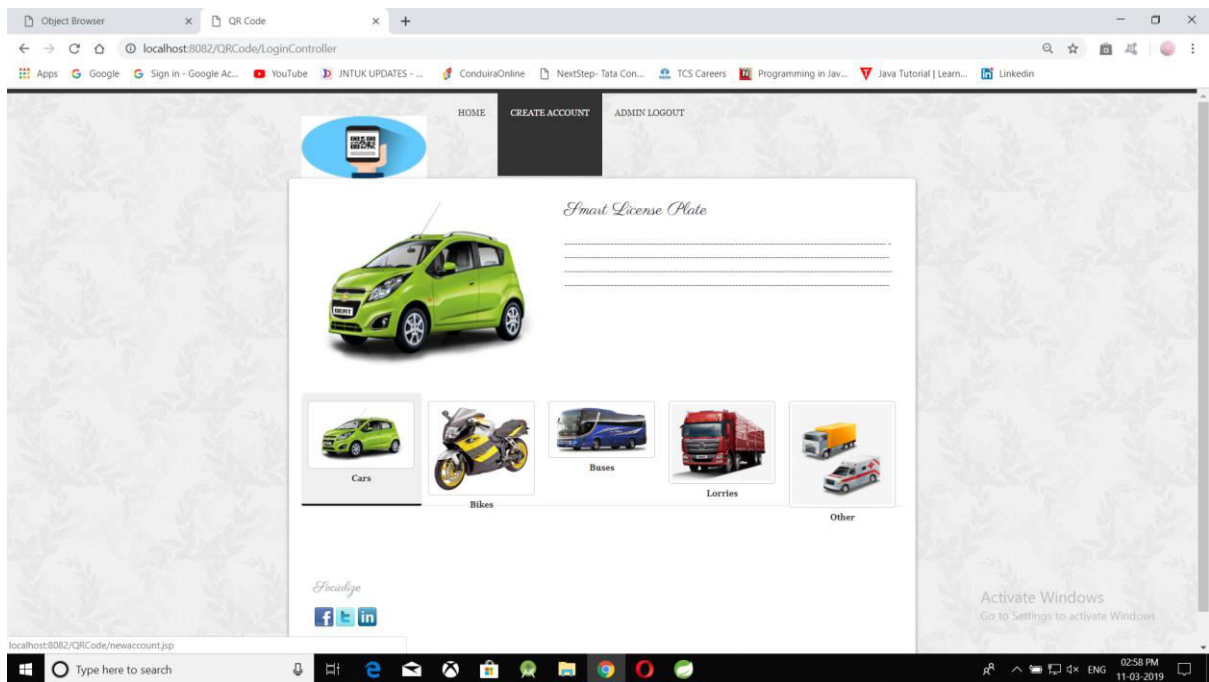
-VE TEST CASES

S. No	Test case Description	Actual value	Expected value	Result
1	Create the new user registration process	Enter the personal info and address info.	Personal info and address info it's not update into database successfully.	False
2	Enter the username and password	Verification of login details.	Login failed	False
3	Upload information	Enter all fields	Web data is not created successfully.	False
4	Generate QR code	Retrieve all fields	Web data is not available in database	False

OUTPUTS

7. OUTPUT






Object Browser x QR Code x +

localhost:8082/QRCode/personaldetails.jsp

Apps Google Sign in - Google Ac... YouTube JNTUK UPDATES - ... ConduiraOnline NextStep- Tata Con... TCS Careers Programming in Jav... Java Tutorial | Learn... LinkedIn

HOME CREATE ACCOUNT ADMIN LOGOUT



Personal Details

Details

Name

User Id

(Vehicle Number)

Pin

Confirm Pin

Address

Phone No

Activate Windows
Go to Settings to activate Windows.

Type here to search

02:58 PM 11-03-2019

Object Browser x QR Code x +

localhost:8082/QRCode/personaldetails.jsp

Apps Google Sign in - Google Ac... YouTube JNTUK UPDATES - ... ConduiraOnline NextStep- Tata Con... TCS Careers Programming in Jav... Java Tutorial | Learn... LinkedIn

Address

Phone No

Email

Aadhar No

Gender ☒ Male ☐ Female

Date Of Birth

Upload Image

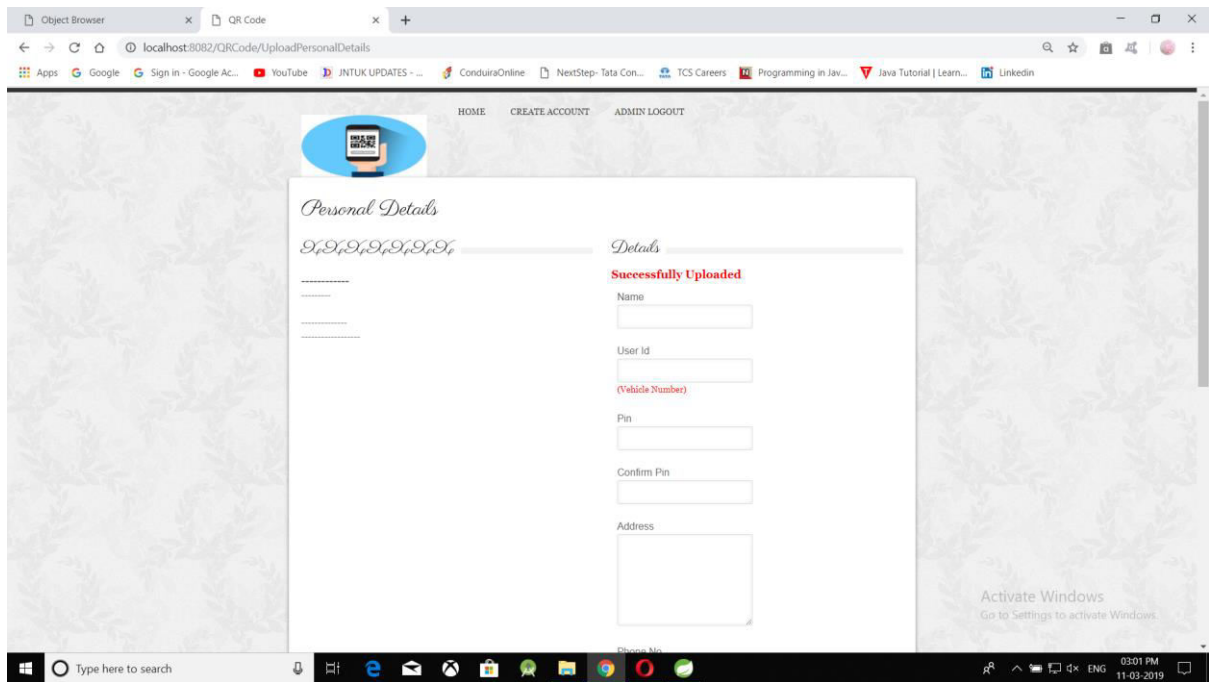
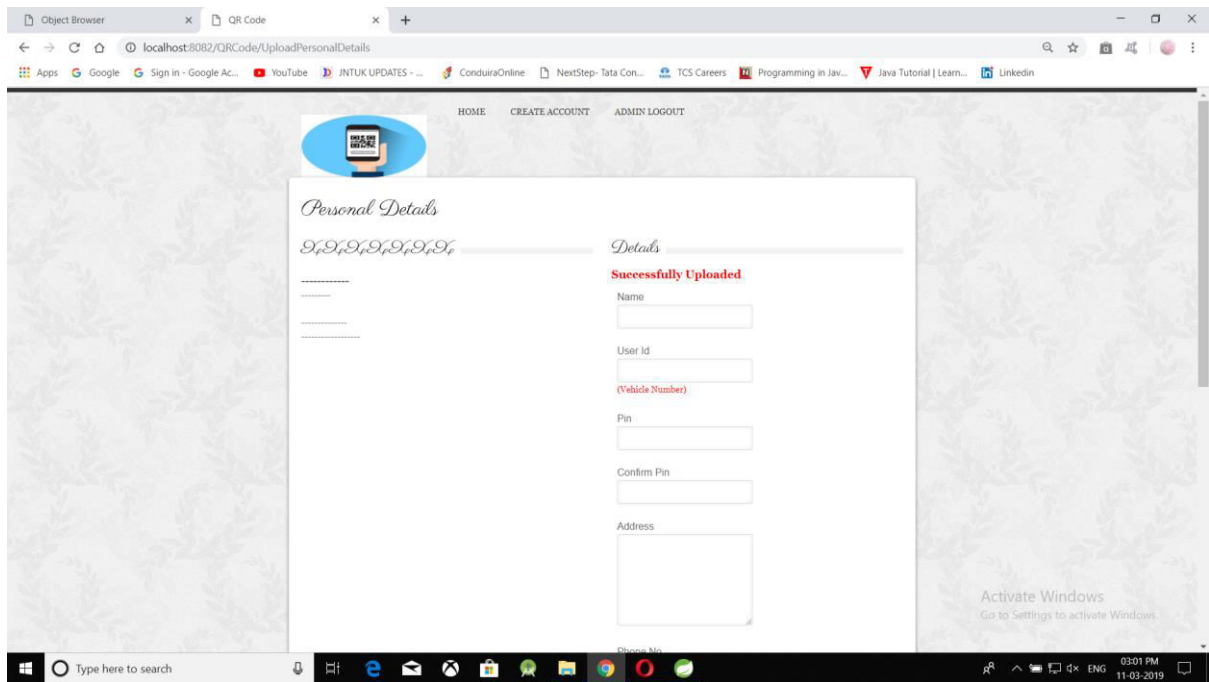
Choose File No file chosen

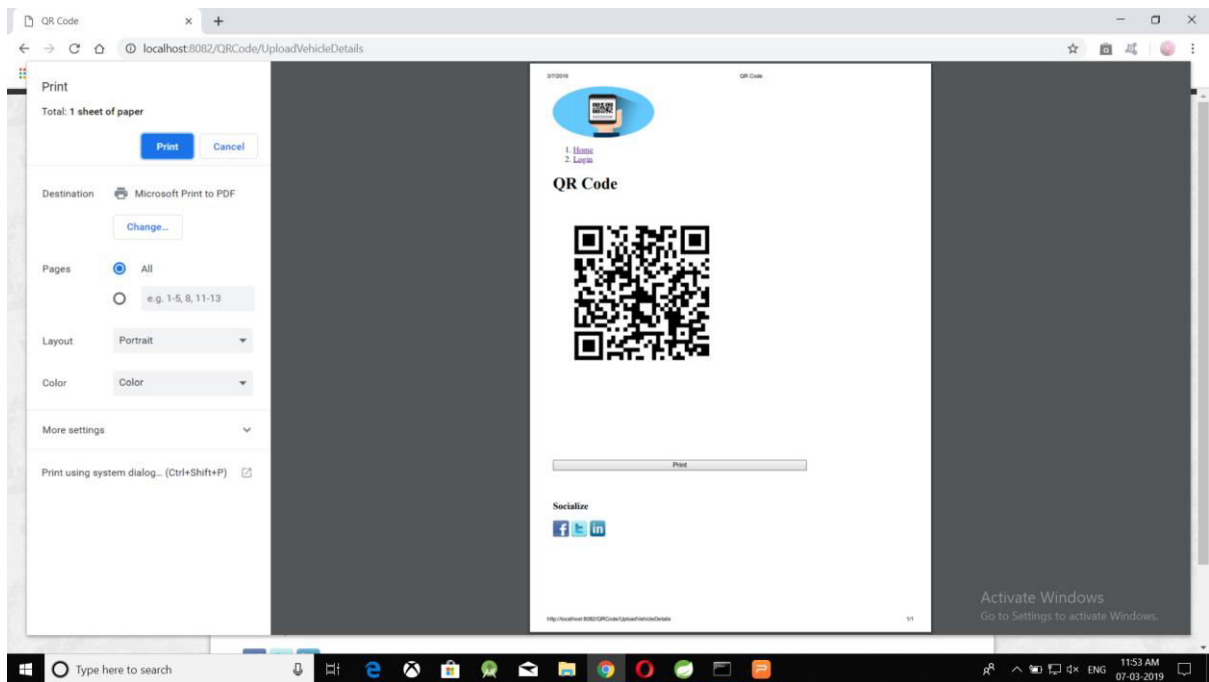
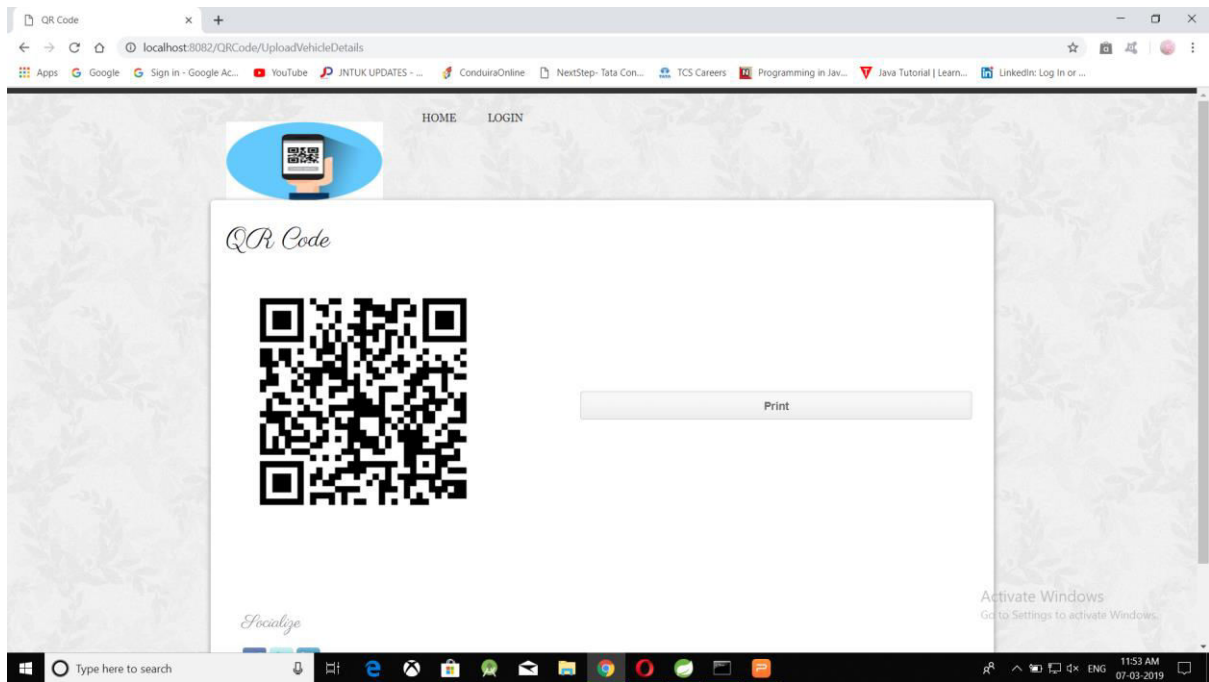
Submit

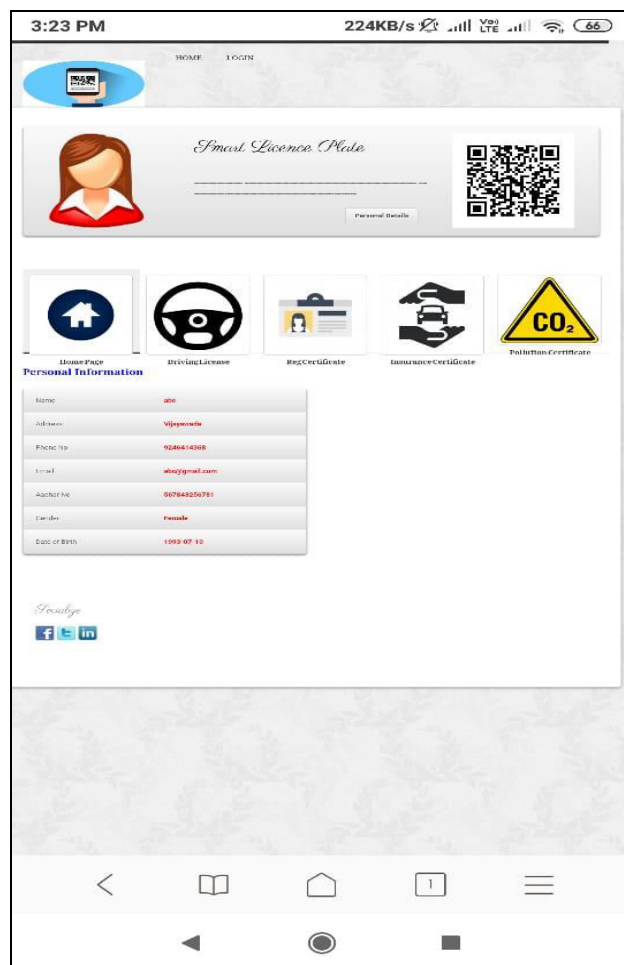
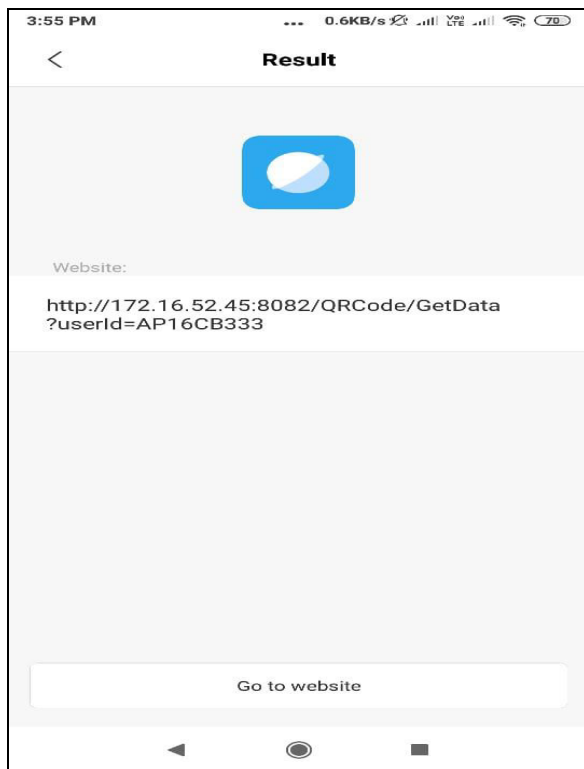
Activate Windows
Go to Settings to activate Windows.

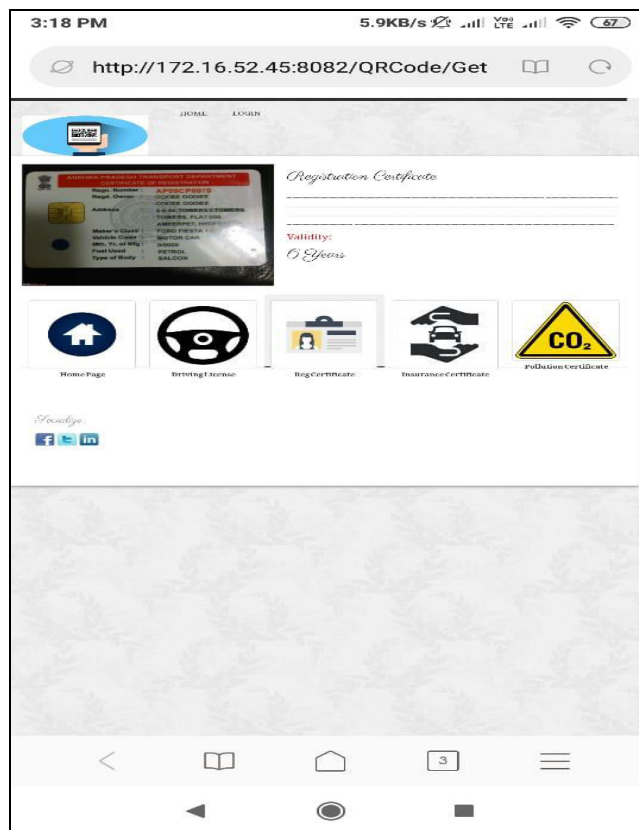
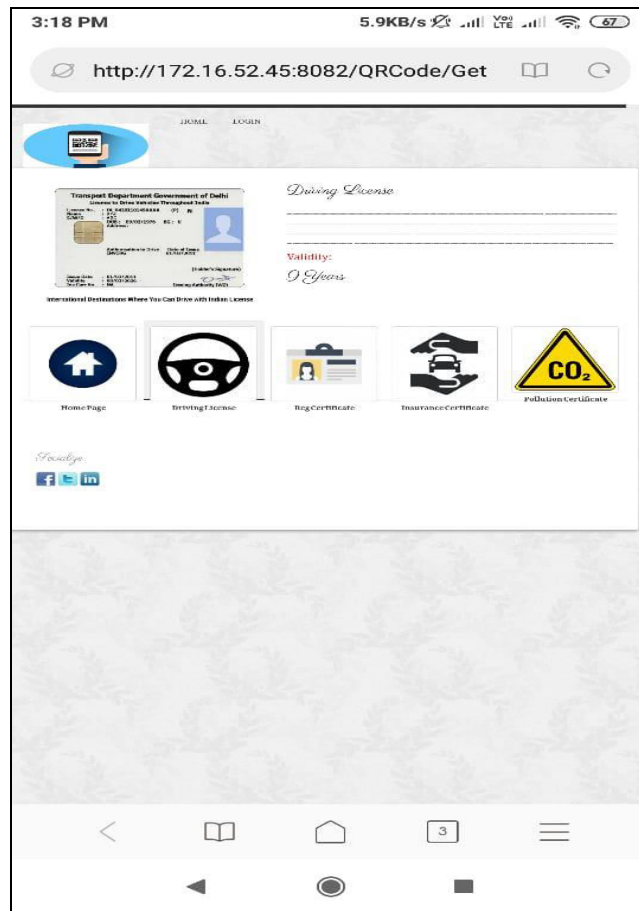
Type here to search

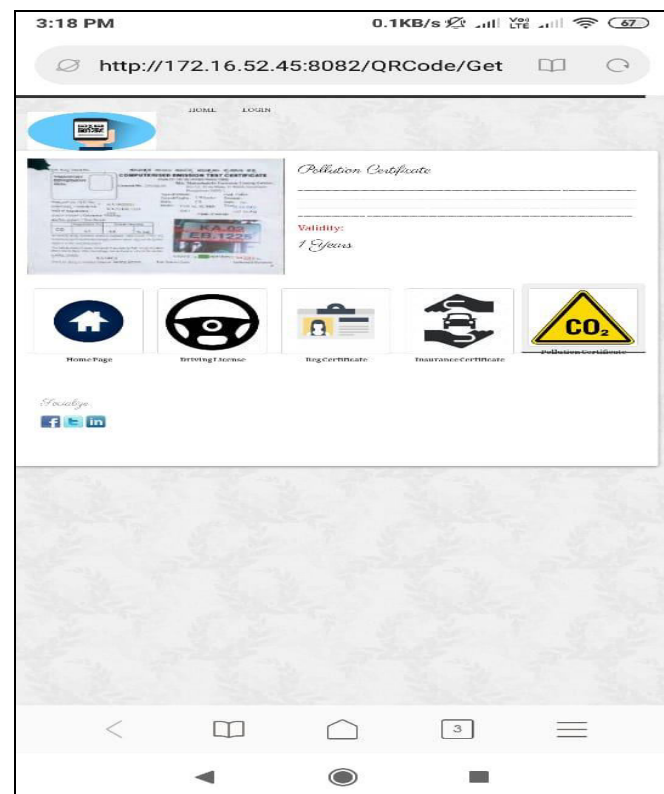
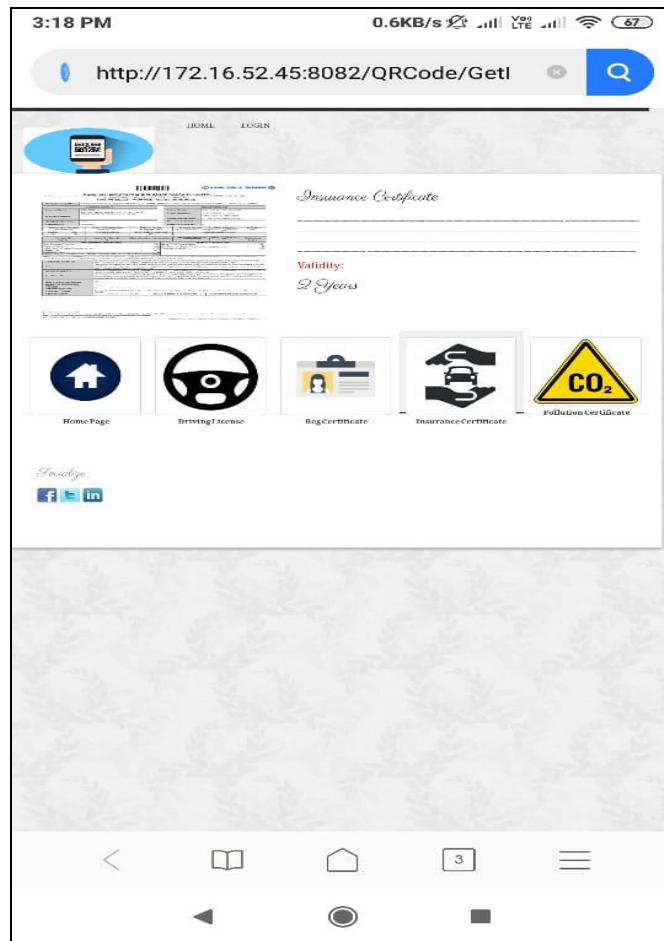
02:58 PM 11-03-2019











CONCLUSION

8. CONCLUSION

WORK DONE:

- The “Smart License Plate” was successfully designed and is tested for accuracy and quality.
- During this project we have accomplished all the objectives and this project meets the needs of the organization.
- The developed will be used in retrieving and generating information for the concerned requests.

GOALS:

- ✓ Reduced entry work
- ✓ Easy retrieval of information
- ✓ Reduced errors due to human intervention
- ✓ User friendly screens to enter the data
- ✓ Portable and flexible for further enhancement
- ✓ Web enabled.
- ✓ Fast finding of information requested

REFERENCES

9. REFERENCES

1. Peer-review under responsibility of the organizing committee of RAEREST 2016.
<http://creativecommons.org/licenses/by-nc-nd/4.0/>
2. Hogpracha, W., & Vongpradhip, S. (2015, July). Recognition system for QR code on moving car. In Computer Science & Education (ICCSE), 2015 10th International Conference on (pp. 14-18). IEEE.
3. Denso Wave Incorporated “<http://www.qrcode.com/en/about/>”
4. Java Complete Reference by Herbert Shield
5. Database Programming with JDBC and Java by George Reese
6. Java and XML by Brett McLaughlin
7. Wikipedia, URL: <https://www.wikipedia.org>.
8. Answers.com, Online Dictionary, Encyclopaedia and much more,
URL: <https://www.answers.com>
9. Google, URL: <https://www.google.co.in>
10. Project Management URL: <https://www.startwright.com/project.html>
11. QR Images: Optimized Image Embedding in QR Codes Gonzalo J. Garateguy, Member, IEEE, Gonzalo R. Arce, Fellow, IEEE, Daniel L. Lau, Senior Member, IEEE, Ofelia P. Villarreal, Member, IEEE
12. Encryption in QR Code Using Steganography Suraj Kumar Sahu* Guided By- Mr. Sandeep Kumar Gonnade** (Department of Computer Science, MATS University, Raipur (Department computer science, MATS University, Raipur)

13. QR Images: Optimized Image Embedding in QR Codes Gonzalo J. Garateguy, Member, IEEE, Gonzalo R. Arce, Fellow, IEEE, Daniel L. Lau, Senior Member, IEEE, Ofelia P. Villarreal, Member, IEEE
14. Evaluating the Use of Quick Response (QR) Code at Sulaimani University Libraries Ako Muhammad Abdullah*, Roza Hikmat Hama Aziz Computer Science Department Faculty of Physical & Basic Education Sulaimani University, Kurdistan Region, Iraq
15. S. Ma, Y. Zheng and O. Wolfson. 2015. Real-Time City-Scale Taxi Ridesharing. in IEEE Transactions on Knowledge and Data Engineering.
16. M. Furuhata, M. Dessouky, F. Ordonez, M.E. Brunet, X. Wang, S. Koenig. 2013. Ridesharing: the State-of-the-art and Future Directions. In Transportation Research Part B: Methodological, University of Southern California.
17. 2013. http://planningcommission.nic.in/reports/genrep/NTD_PC_Vol_01.pdf.
18. R. Balan, K. Nguyen and L. Jiang. 2011. Real-time trip information service for a large taxi fleet. in Proc. 9th International Conference on Mobile System Application Services.
19. Amey A.M. 2010. Real-time ridesharing: Exploring the opportunities and challenges of designing a technology-based rideshare trial for MIT community. Master's thesis, Dept of Urban Studies and Planning, Dept of Civil and Environmental Engineering, MIT, MA, USA.
20. P.Y. Chen, J.W. Liu and W.T. Chen. 2010. A fuel saving and pollution-reducing dynamic taxi sharing protocol in VANETs. in Proc. IEEE 72nd Vehicle Technology Conference.