

ITM COLLEGE OF ARTS AND SCIENCE

(Recognized by Govt. of Kerala& Affiliated to Kannur University)

P.O Pavvannoormotta, Mayyil, Kannur, Kerala – 670602



PROJECT REPORT

ON

“IONROAD”

Submitted by:

NAJIL NOUSHAD M P

IM21BCAR27

Under the guidance of

Mr.JITHESH P

(Asst.Professor Department of Computer Application)

VI Semester BCA 2021 - 2024

ITM COLLEGE OF ARTS AND SCIENCE

(Recognized by Govt. of Kerala & Affiliated to Kannur university)

P.O Pavvannoormotta, Mayyil, Kannur, Kerala – 670602



CERTIFICATE

This is to certify that the project entitled “**IONROAD**” is a bonafide record of work done by **NAJIL NOUSHAD M P** under our supervision and guidance towards partial fulfillment for the award for Bachelor Degree of Computer Application of Kannur University.

Internal Guide

Head of the department

Principal

External Examiners

1.

2.

ITM COLLEGE OF ARTS AND SCIENCE

(Recognized by Govt. of Kerala & Affiliated to Kannur university)

P.O Pavvannoormotta, Mayyil, Kannur, Kerala – 670602



DECLARATION

We hereby declare that the project entitled “**IONROAD**” submitted for the BCA Degree is our original work and the project is not used for the award of any degree, associate ship, fellowship or any other similar titles.

Signature of the student:

Place:

Date:

ACKNOWLEDGEMENT

With a sense of gratitude and respect, we would like to extend our heartiest thanks to all those who are involved directly or indirectly in preparation of this project. It was pleasant and highly educative and a good experience developing this project.

Our sincere thanks to **Mr. Aswin Krishnan VK**, project manager of RISS TECHNOLOGIES to perform project here at RISS Technologies. We also thank him for constant interaction with us. We owe the success of the project to my project Guide, **Mrs.Haritha P**, who was a tremendous supporter and eager teacher, for providing excellent guidance for this project. They are one of the major sources behind the success of the project. We thank the management and all staff members of RISS Technologies who helped us throughout the analysis of the project. Gratefulacknowledgment is extended to all people who made this project possible. We express our heartily gratitude to our beloved principal **Prof Dr. Majeed P.P** and **Mrs.Reshma Thoonoli** head of Computer Science Department, for all their enthusiastic attention and timely help in completion of our project.

It is an immense pleasure to acknowledge the assistance and guidance received from our guide, **Mr.Jithesh P**. We owe a great deal to her patience and many valuable suggestions. Despite her busy schedule, she had helped us throughout the project, which helped us to develop our knowledge in this concept.

We also express our heartfelt gratitude to **Mrs.Reshma T**, **Mrs.Nishitha M**, and **Mrs.Vipina Vishwanath** for their great technical comments throughout our project. We also appreciate the encouragement and constructive criticism received from our parents and friends which went a long way to make this project a satisfying experience for us. Most of all and more than ever, we would like to thank our parents, family members and friends, for their great support, encouragement, kindness and patience.

ABSTRACT

In recent years, the increasing number of vehicles on roads leads to an increase in traffic accidents. It is found that distracted driving was related to one-tenth of fatal crashes. Distracted driving fatalities have increased more rapidly than those caused by drunk driving, speeding and failing to wear a seat belt. A driver is considered to be distracted when there is an activity that attracts his/her attention away from the task of driving. Distracted driving is an established cause of motor vehicle crashes for all ages. With the rapidly growing elderly population and more adults embracing technology, distracted driving is also increasing in prevalence within that population—particularly cell phone usage behind the wheel. Another cause of accidents is drowsiness during driving.

One of the main challenges faced by drivers is drowsiness, which can result in accidents and fatalities on the roads. Additionally, potholes pose a substantial risk to both drivers and the overall condition of the road. In response to these challenges, my project aims to develop a safety feature that addresses both drowsiness and potholes.

INDEX

SL NO	TITLE	PAGE NO.
1	INTRODUCTION	10
1.1	PROJECT OVERVIEW	11
1.1	PURPOSE OF THE PROJECT	12
1.2	SCOPE OF THE PROJECT	12
2	SYSTEM STUDY AND ANALYSIS	13
2.1	METHODOLOGY	14
2.2	EXISTING SYSTEM	14
2.3	PROPOSED SYSTEM	15
3	FEASIBILITY STUDY	16
3.1	TECHNICAL FEASIBILITY	17
3.2	BEHAVIORAL FEASIBILIT	17
3.3	ECONOMIC FEASIBILITY	18

3.4	OPERATIONAL FEASIBILITY	18
4	SOFTWARE REQUIREMENT ANALYSIS	20
4.1	SYSTEM ANALYSIS AND DESIGN	21
4.2	CODE GENERATION	20
4.3	TESTING	20
4.1	SYSTEM REQUIREMENTS	21
5	MODULE DESCRIPTION	23
5.1	ADMIN	24
5.2	PARTNER	24
5.3	DRIVER	24
5.4	CAMERA	24
6	SOFTWARE TOOL USED	25
6.1	PYCHARM	26
6.2	ANDROID	28

6.3	DREAMWEAVER	31
6.4	MySQL	31
6.5	DATABASE SERVICES	32
7	SYSTEM DESIGN AND DEVELOPMENT	33
7.1	SYSTEM DESIGN	34
7.2	INPUT DESIGN	35
7.3	OUTPUT DESIGN	36
7.4	LOGICAL DESIGN	38
7.5	DATABASE DESIGN	38
8	SYSTEM DESIGN	39
8.1	DATA FLOW DIAGRAM	40
8.2	ENTITY RELATIONSHIP DIAGRAM	47
8.3	USE CASE DIAGRAM	50
9	DATABASE DESIGN	55

9.1	NORMALIZATION TECHNIQUE	56
9.2	TABLE DESIGN	58
10	SYSTEM TESTING	63
10.1	TEST TYPES	63
10.2	TESTING METHODS	64
11	IMPLEMENTATION	67
11.1	IMPLEMENTATION PROCEDURES	69
12	SYSTEM MAINTANANCE	70
12.1	CORRECTIVE MAINTENANCE	71
12.2	ADAPTIVE MAINTENANCE	71
12.3	PERFECTIVE MAINTENANCE	72
12.4	PREVENTIVE MAINTENANCE	72
13	TIME LINE CHART	73
14	FUTURE ENHANCEMENT	75

15	CONCLUSION	77
16	APPENDIX	79
16.1	SCREENSHOTS(WEB & ANDROID)	80
17	SAMPLE CODE	90

INTRODUCTION

1. INTRODUCTION

Our project title is the IONROAD and there are many functions include in IONROAD like Driver Drowsiness Detection,Pothole Detection,Viewing Current Location of Driver,Messaging to Driver etc.

In recent years, the increasing number of vehicles on roads leads to an increase in traffic accidents. It is found that distracted driving was related to one-tenth of fatal crashes. Distracted driving fatalities have increased more rapidly than those caused by drunk driving, speeding and failing to wear a seat belt. A driver is considered to be distracted when there is an activity that attracts his/her attention away from the task of driving. Distracted driving is an established cause of motor vehicle crashes for all ages. With the rapidly growing elderly population and more adults embracing technology, distracted driving is also increasing in prevalence within that population—particularly cell phone usage behind the wheel. Another cause of accidents is drowsiness during driving.

Consistently numerous individuals lose their lives because of deadly street mishaps all throughout the planet and sleepy driving is one of the essential drivers of street mishaps and demise. Weariness and micro-sleep at the driving controls are frequently the underlying driver of genuine mishaps. However,initial signs of fatigue can be detected before a critical situation arises and therefore, detection of driver's fatigue and its indication is an ongoing research topic. The vast majority of the conventional techniques to identify tiredness depend on social viewpoints while some are meddling and may divert drivers, while some require costly sensors.

Consequently, in this,Project a light-weight, continuous driver's tiredness identification framework is created and carried out in an application. The framework records the recordings and distinguishes the driver's face in each casing by utilizing picture handling procedures.The system is capable of detecting facial landmarks, computes Eye Aspect Ratio (EAR) and Eye Closure Ratio (ECR) to detect driver's drowsiness based on adaptive thresholding. Machine learning algorithms have been employed to test the efficacy of the

proposed approach. Empirical results demonstrate that the proposed model achieved an accuracy of 84% using machine learning algorithms.

The potholes create the bad condition on the road. Especially in monsoon, potholes are occur much more because of poor road .so avoiding this potholes we introduce “IONROAD” which detect the potholes on the road of the particular area. So user can alert about that road pothole is published to all the users.

1.1 PURPOSE OF THE PROJECT

Nowadays the driver's safety in the car is one of the most wanted systems to avoid accidents. To overcome this issue, we propose a drowsiness detection system that has the potential to be implemented in real vehicles. The proposed work also focuses on road conditions. This can alert users on time about road conditions so that users can adjust their speed.

Poor road condition damage to the vehicles and occur the serious road accident, Although we complain about the bad road but there is no way to detect or report them at scale. To avoid this issue we develop a system to detect potholes and warning about those road condition.

1.2 SCOPE OF THE PROJECT

The scope of my project will include designing and implementing a dual-function warning system that detects both drowsiness in drivers and alerts them of potholes on the road. The system will consist of a combination of sensors, cameras, and software algorithms that work together to monitor the driver's behavior and road conditions in real-time.

SYSTEM STUDY AND ANALYSIS

2. SYSTEM STUDY AND ANALYSIS

2.1 METHODOLOGY

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration. The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.

2.2 EXISTING SYSTEM

The current system in place for addressing drowsiness and potholes for drivers is severely lacking. It fails to address the individual needs of drivers and lacks real-time assistance. They require drivers to constantly switch between different devices and interfaces, causing distraction and potential accidents. Furthermore, they do not provide a comprehensive solution, as they only address one issue at a time. This leaves drivers vulnerable to the dangers of drowsiness and potholes, putting their safety at risk. In conclusion, it lacks innovation and fails to provide a user-friendly interface for drivers. It is time for a change in the way we address these issues, with a more personalized and engaging system that truly caters to the needs of drivers.

DRAWBACKS

- Reliance on Technology and Devices.
- The System may not be Compatible with all Types of Vehicles
- Not Feasible.
- Different Interfaces for each Functions.
- Not User Friendly.

2.3 PROPOSED SYSTEM

My project, IONROAD, is a revolutionary innovation that aims to address two major issues faced by drivers - drowsiness and potholes. My proposed project aims to address some of the most common issues faced by drivers on the road. One of the key features is a drowsiness detection system that alerts drivers when they start to feel sleepy behind the wheel. This will help prevent accidents caused by driver fatigue. Additionally, the project includes a pothole detection system that uses sensors to identify and warn drivers of potential hazards on the road. Another important feature is the ability to voice out important text messages to the driver, allowing them to stay connected without taking their eyes off the road. Lastly, the project also allows for the viewing of the driver's current location, providing peace of mind for loved ones and ensuring drivers stay on track with their route. With these features, my project aims to make driving safer and more convenient for all.

Advantages of Proposed System:

- Drowsiness Detection.
- Real-Time Pothole Mapping.
- Access to All Users.
- Convenience of Viewing the Current Location of the Driver.
- Send Messages to Driver which will Voice-Out.
- Partner can View Drowsiness Detected Time(if any).
- Easy Setup.
- User Friendly Interface.
- Cost Effective.
- Facility to send feedback and complaints to admin.

FEASIBILITY STUDIES

3.FEASIBILITY STUDIES

A feasibility study is a preliminary study undertaken to determine and document projects viability. Results of this study are used to make a decision whether to proceed with the project, or not. If it indeed leads to a project being approved, it will-before the real work of problem and the recommendation on the best alternative. The feasibility study was carried out to find whether the proposed system can be developed and implemented without any problem within the time limit. The following feasibility was considered for the project in order to ensure that the project is viable and it does not have any major obstructions. In this regard the following feasibility analysis was conducted.

- Technical Feasibility
- Behavioral Feasibility
- Economic Feasibility
- Operational Feasibility

3.1 TECHNICAL FEASIBILITY

In this technical feasibility was done to find out whether the necessary technology exist the equipment have the capacity to hold data required by the use of new system. Checker can develop by using the existing technology Microsoft SQLserver.html for front-end development.

3.2 BEHAVIORAL FEASIBILITY

When doing project in an external firm there can be in cooperation and other unpredictable egoism from the colleagues especially when a new idea is being introduced. These are behavioral and depend upon individual behavior of person involved. Behavioral feasibility was done in order to analyze the attitude of the people who are directly involved with my project.

3.3 ECONOMIC FEASIBILITY

The proposed system is cost effective because of its user-friendly interface. The administrator can directly view and make updates on the records.

3.4 OPERATIONAL FEASIBILITY

The proposed project is said to be beneficial only if they can be modules and carved out in to a system that will meet all the requirement. The best for the feasibility prohibits if the system would run without faults when it is deployed or there are any major hurdles to be the implementations of the proposed system. Theproposed system is not supposed to cause any harm to the user or the computer that is being used and the system is safe and secure.

SOFTWARE REQUIREMENT ANALYSIS

4.1 SOFTWARE REQUIREMENT ANALYSIS

This process is also known as feasibility study. The need for this software was investigated. A Software Requirement Specification document was created at the end of this step. It holds all information required to develop the project which includes functional and non-functional requirements. The requirement gathering process is intensified and focused specially on software. The information domain for the software, as well as required function, behavior, performance and interfacing are well understood.

4.1 SYSTEM ANALYSIS AND DESIGN

In this phase, the software development process, the software's overall structure is defined. The logical system of the product is developed in this phase. A data flow diagram which indicates the flow of data through the system is created. This phase produces a high-level design and a low-level design. The high-level design comprises the modules to be generated and their purpose. The low-level design consists of a more elaborated view of the modules which includes the method to be defined.

4.2 CODE GENERATION

The design created in the design phase must be translated into a machine-readable form. The code generation step performs this task. Since this project processing which involves connection to database, the language has chosen as Python, JAVA. Front end is created with ANDROID, HTML and Back-end with MySQL is used for storing data.

4.3 TESTING

After the code was generated, the software program testing begins. An error in the program flow was identified in this step.

4.4 SYSTEM REQUIREMENT

HARDWARE SPECIFICATION

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

- Input Device : Mouse,Keyboard
- Output Device : Monitor
- Memory : 4 GB Ram(Minimum)
- Processor : Intel core i3 or above

SOFTWARE SPECIFICATION

One of the most difficult tasks is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system requirements. The application requirement:

- Operating System: Windows 8 /10for Better Performance
- Front End : Python (Flask)
- Back End : MySQL
- Software Used : Pycharm
- Web Browser :Internet Explorer/Google Chrome/Firefox(for web application)

MODULES

5. Modules

5.1 Admin

- Login
- View drivers
- View complaints
- Send reply
- View reviews

5.2 Partner

- Login
- View drivers (partners)
- View Drowsiness detection
- Messaging to driver
- View Locations

5.3 Driver

- Registration
- Login
- Partner registration
- Location Sharing
- Voice assistance support
- Camera assistance while closing eye lids(sleeping)
- Get road condition notifications using GPS, accelerometer

5.4 Camera

- Drowsiness detection

SOFTWARE TOOLS USED

6. SOFTWARE TOOL USED

6.1 PyCharm (Python Interpreter)

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by Meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding),

which binds method and variable names during program execution.

6.2 Android

Android is an operating system based on the Linux kernel. The project responsible for developing the Android system is called the Android Open Source Project(AOSP) and is primarily lead by Google. The Android system supports background processing, provides a rich user interface library, supports 2-D and 3-D graphics using the OpenGL-ES (short OpenGL) standard and grants access to the file system as well as an embedded SQLite database. An Android application typically consists of different visual and non-visual components and can reuse components of other applications. The Android system is a full software stack, which is typically divided into the four areas. The levels can be described as:

- Applications - The Android Open Source Project contains several default applications, like the Browser, Camera, Gallery, Music, Phone and more.
- Application framework - An API which allows high-level interactions with theAndroid system from Android applications
- Libraries and run time - The libraries for many common functions (e.g.: graphic rendering, data storage, web browsing, etc.) of the Application Framework and the Dalvik runtime, as well as the core Java libraries for running Android applications.

Linux kernel - Communication layer for the underlying hardware .The Linux kernel, the libraries and the run time are encapsulated by the application framework. The Android application developer typically works with the two layers on top to create new Android applications. The Android Software Development Kit (Android SDK) contains the necessary tools to create, compile and package Android applications. Most of these tools are command line based. The primary way to develop Android applications is based on the Java programming language. The Android SDK contains the Android debug bridge (adb), which is a tool that allows you to connect to a virtual or real android device, for the purpose of managing the device or debugging your application. Google provides two integrated development

environments (IDEs) to develop new applications. The Android Developer Tools (ADT) are based on the Eclipse IDE. ADT is a set of components (plug-ins), which extend the Eclipse IDE with Android development capabilities. Google also supports an IDE called Android Studio for creating Android applications. This IDE is based on the IntelliJ IDE. Both IDEs contain all required functionality to create, compile, debug and deploy Android applications. They also allow the developer to create and start virtual Android devices for testing. Both tools provide specialized editors for Android specific files. Most of Android's configuration files are based on XML. In this case these editors allow you to switch between the XML representation of the file and a structured user interface for entering the data. Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on Equinox, an implementation of the OSGi core framework specification. In addition to allowing the android Platform to be extended using other programming languages such as C and Python, the plug-in framework allows the Eclipse Platform to work with type setting languages like LaTeX networking applications such as telnet and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with support for other version control systems provided by third-party plug-ins. The Eclipse SDK includes the Eclipse Java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced re factoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of meta data over a flat files pace allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards. Eclipse implements use the graphical control elements of the Java toolkit called SWT, whereas most Java applications use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate graphical user interface layer called

JFace, which simplifies the construction of applications based on SWT.

6.3 DREAMWEAVER

Adobe Dreamweaver is a software program for designing web pages, essentially a more fully featured HTML web and programming editor. The program provides a what-you see-is-what-you-get (WYSIWYG) interface for users to create and edit web pages in a more user-friendly environment. Dreamweaver supports multiple mark-up languages including HTML and Extensible Mark-up Language (XML), style sheet languages like Cascading Style Sheets (CSS), and programming languages including JavaScript, C#, Visual Basic (VB), Active Server Pages (ASP), and others. The program is also available in a number of languages, including English, Spanish, French, German, Japanese, Chinese (both Simplified and Traditional), Italian, Russian, and more. Dreamweaver was originally developed and published by Macromedia in 1997. Adobe purchased Macromedia (which included the rights to Dreamweaver) in 2005 and continued the development of the program. The many features of Dreamweaver make it a versatile web editing tool, where it be for creating complex or very simple sites.

6.4 MySQL

It is one of the most popular open source relational database management system that makes the best use of SQL to process data in a database. Currently managed by a company called MySQL AB, it offers excellent quick-start capability with the average time from software download to installation completion being less than fifteen minutes. Some of the features of MySQL are:

- Cross-platform support, stored procedures, triggers, cursors and updatable views
- Independent storage engines – MyISAM, InnoDB and MySQL.
- SSL support, query caching, nested selects, replication support.
- Full-text indexing and searching using MyISAM engine.
- Embedded database library.
- Partial Unicode support (UTF-8 and UCS-2).

6.5 DATABASE SERVICES

A data base is a collection of inter related data stored with minimum redundancy to serve many quickly and efficiently. The general way is to make information accessing easy, quick, inexpensive and flexible for the user. In data base design several objectives are considered controlling redundancy, ease of learning and use, data dependence, more information at low cost, accuracy and integrity are some of them.

SYSTEM DESIGN AND DEVELOPMENT

7. SYSTEM DESIGN AND DEVELOPMENT

System design is the process of defining the components, modules, interfaces and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.

7.1 SYSTEM DESIGN

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design involves designing a new system or computer application that will meet the requirements identified during system analysis. System design is concerned with the computer oriented design of the system-details of the input transactions, the details of the printed reports, screens and other outputs, database structure, the contents of records, the processing required and the efficiency of the system from a computer processing point of view. In general, the following design objectives must be kept in mind:

- Practicality of the System
- Efficiency of the System
- Minimized cost of the system
- Flexibility of the System
- Security of the System

The system design process is to achieve objectives like practicability, efficiency, minimized cost, flexibility, and security of the system. It involves, first, logical design and then, physical construction of the system. The logical design specifies the structure and characteristics of output, input, files, database and procedures. The physical design involves the construction of actual program files and a working system.

7.2 INPUT DESIGN

Input design plays a vital role in the life cycle of the software development. It requires the very careful attention of the developers. It specifies the manner in which the data enters the system for processing. Input design can ensure the reliability of the system and the producer's results from the accurate data or it may result in the output of an enormous data. According to the software engineering concepts, the input forms of screens are designed to provide to have a validation control over the input limit. This system has input screens in almost all the modules. Error messages have been developed to alert the user whenever he commits some mistakes and guides him in the right way. Input design determines whether the user interacts with the system efficiently. It consists of developing specifications and procedures for the data preparations. So, structured steps are necessary to put the transaction data into usable form for processing. Input design is the process of converting the user-originated inputs to a computer based format. It also includes the method of input and entry into the system.

There can be many factors that can be taken into account

- Flexibility of format
- Speed
- Accuracy
- Type of input
- Verification methods
- Offline contact facility
- Automatic features
- Ease of use

Keyboard is the most commonly used input media. Inaccurate input data are the most common cause of errors in data processing. Errors entered by the user can be controlled by the input design. The design for handling input specifies how the data are accepted for computer processing. Input design is a part of the overall system design that needs careful attention and it includes

specifying the means by which actions are taken. A system user interacting through a workstation must be able to tell the system whether to accept input produced a report or end the processing. The collection of the input data is considered to be the most extensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information, extreme care has to be taken to obtain the information. If the data going into the system is incorrect, then the processing and the outputs will magnify these errors.

7.3 OUTPUT DESIGN

Often the most important feature of the information system for users is the output it produces. Without quality outputs we may even fill the entire system is so unnecessary that they avoid using it and that possibly causes the system fail. Output design refers to the generation of the results and information of the project. It should be in an organized and well-out manner. It involves conceiving, planning out and specifying the external observable characteristics of the product. It includes user displays and the report formats. The normal procedure is to design the outputs in detail first and then to work back with the inputs. The outputs can be in the form of the operational documents and report. The input records have to be validated, edited, organized and accepted by the system before being processed to produce the output. Output is the most important and direct source of information to the users. Designing the computer output should produce in an organized manner. The right output must be developed while ensuring that each output element is designed so that people will find the system easy for the use or user friendly. The output design has to be done so that the results of the processing should be communicated to the user. Effective output design will improve the clarity and performance of the outputs. Output is the main reason for developing the system and the basis on which the usefulness of the application will be evaluated. Output design phase of the system is concerned with the convergence of the information to the end user in a user-friendly manner. The output design should be efficient, intelligible so that the system relationship

with the end user improves and thereby enhancing the process of the decision making.

7.4 LOGICAL DESIGN

A logical design is a conceptual, abstract design. You do not deal with the physical implementation details yet; you deal only with defining the types of information that you need. The process of logical design involves arranging data into a series of logical relationships called entities and attributes.

7.5 DATABASE DESIGN

The data base design is a logical development in the methods used by the computer to access and manipulate data stored in the various parts of the computer system. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as organization resources and as an integrated whole. The main objectives of database are data integration, data integrity and data independence. Database management system (DBMS) allows the data to be protected and organized separately from other resources like hardware, software and programs. DBMS is a software package, which contains components that are not found in other data management packages. The significance of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. I.e. the difference between the logical and physical data. Database design is the first step of the three design activities that are conducted during software engineering process. The impact of data structure on program structure and procedural complexity causes data design to have a profound influence on software quality. The concepts of shared memory and semaphores provide the foundation for an approach to data design. The database is structured in fixed format records of several types. Each record type defines a fixed number of fields or attributes and each field is usually of a fixed length. Usually centralized data management software is called RDBMS.

SYSTEM DESIGN

8. SYSTEM DESIGN

8.1 DATA FLOW DIAGRAM

Dataflow Diagrams (DFD) is the most commonly used way of documenting the processing of the required system. They are the pictorial way of showing the flow of data into, around and out of the system. They can be understood by the users and are less prone to misinterpretation than textual description. A complete set of DFDs provide a compact top-down representation of the system, which makes it easier for users and analysts to envisage the system as a whole. DFD also known as Bubble Chart has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design phase that functionally decomposes the requirements specifications down to the level of details. It does not show the information about the timing processes or information about whether processes will operate in sequence or in parallel. A DFD shows, what kind of data will be put into and out of the system, where data will come from and go to and where data will be stored. A DFD is often a preliminary way of creating the overview of the system.

DFDs mainly use the following symbols:

Process

Process shows the work of the system. Each process has one or more data inputs and produce one or more data outputs. Processes are represented by circles in data flow diagrams.

Data Stores

A data store is a repository of data. Processes can enter data into a store or retrievedata from the data store. Data stores are represented by two parallel lines, which may be depicted horizontally or vertically.

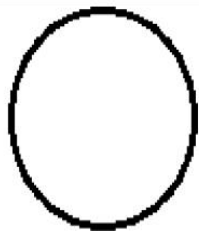
Data Flows

The arrows represent data flow. A data flow is data in motion. A data flow represents an input of data to a process or the output of the data from a process. A data flow is also used to represent the creation, reading, deletion, or updating of data in a file or database.

External Entities

External entities are outside the system but they either supply input the system or use other systems output. They are entities on which the designer has control. External entities that supply data into the system are sometimes called source. External entities that use the system data are called sinks. These are represented by rectangles in the data flow diagram.

In DFD there are four main symbols: -



Circles are used to represent processes. Processes represent activities in which data is manipulated by being stored or retrieved or transformed in some way.



Rectangles are used to represent external entities, which are the sources of data that enter the system or the recipients of data that leave the system.

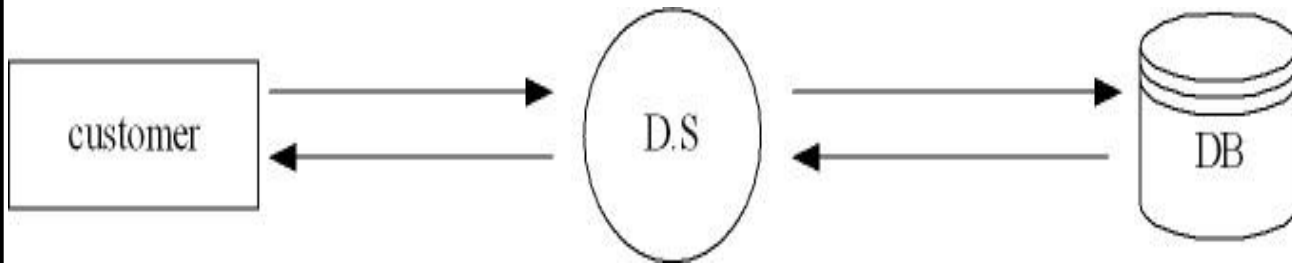


Open rectangles are used for representing data stores or databases.

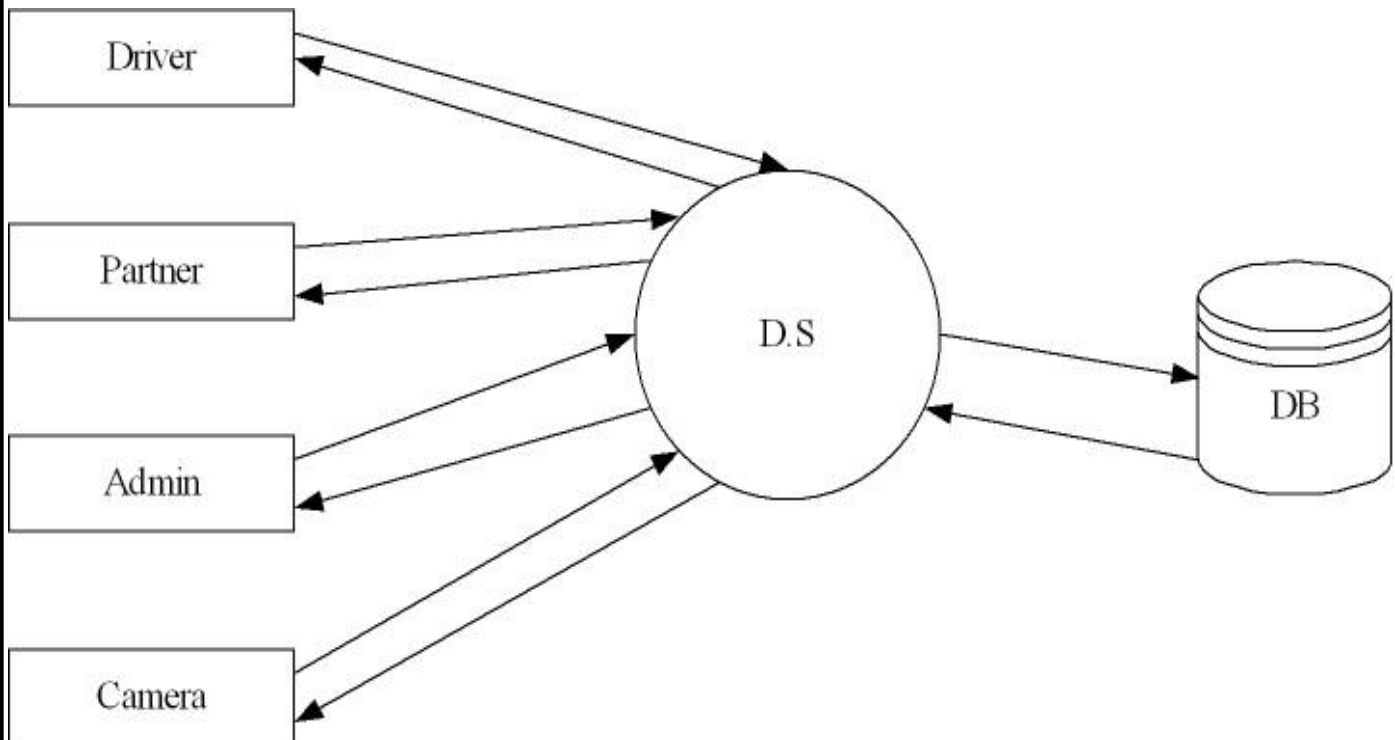


Arrows are used to represent the data flow. Data flows represent the movement of data between other.

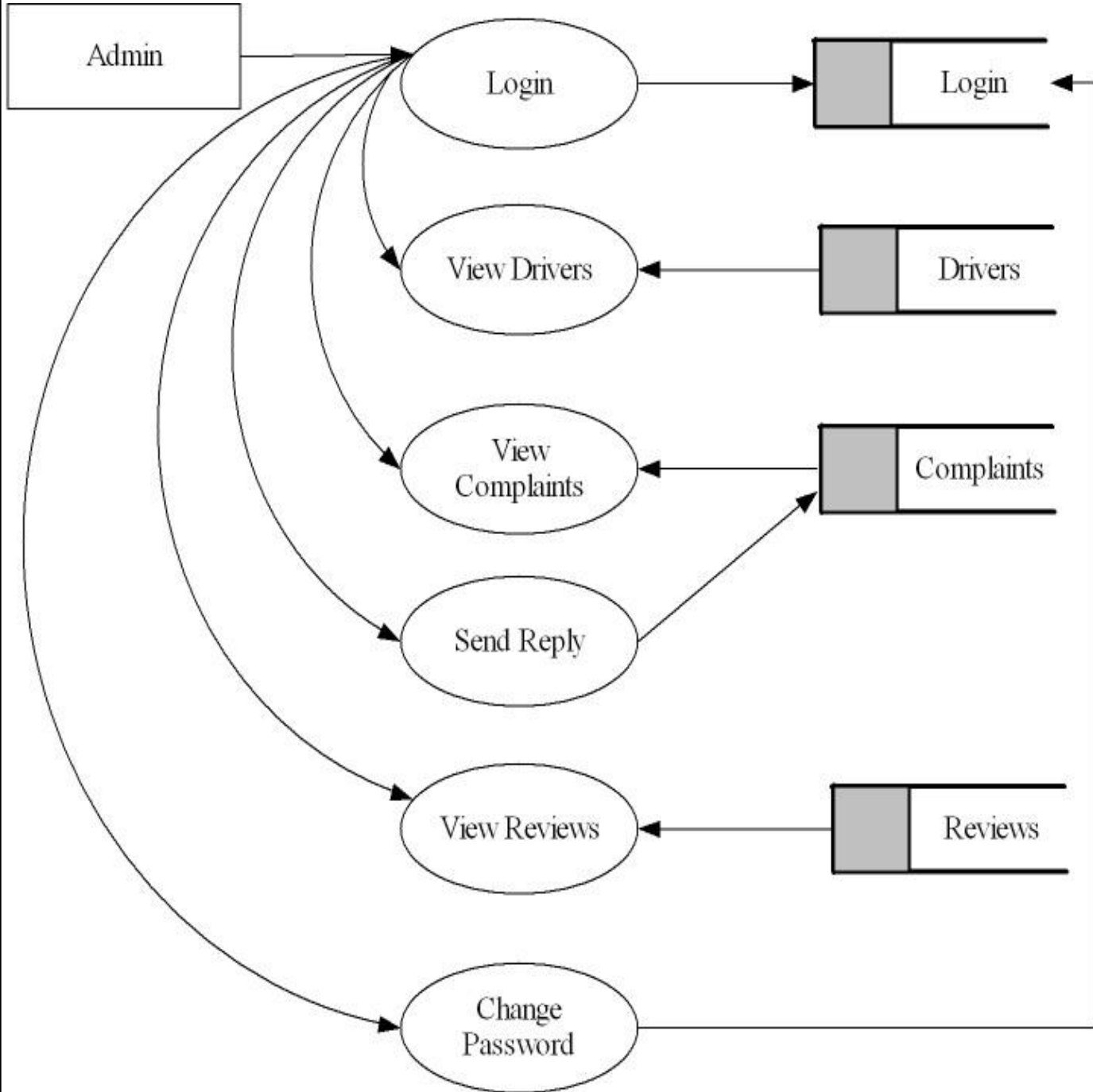
LEVEL 0:



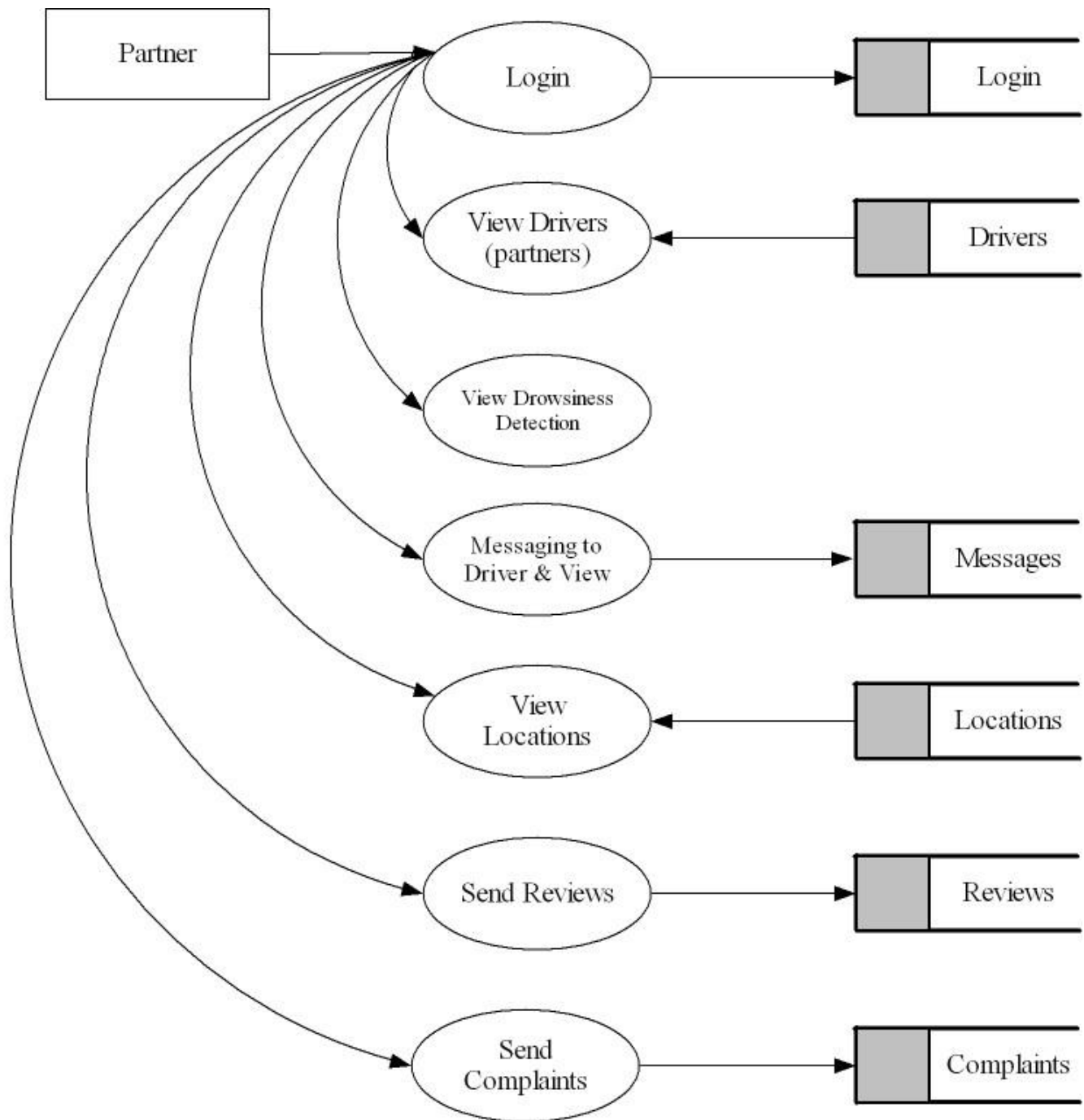
LEVEL 1:



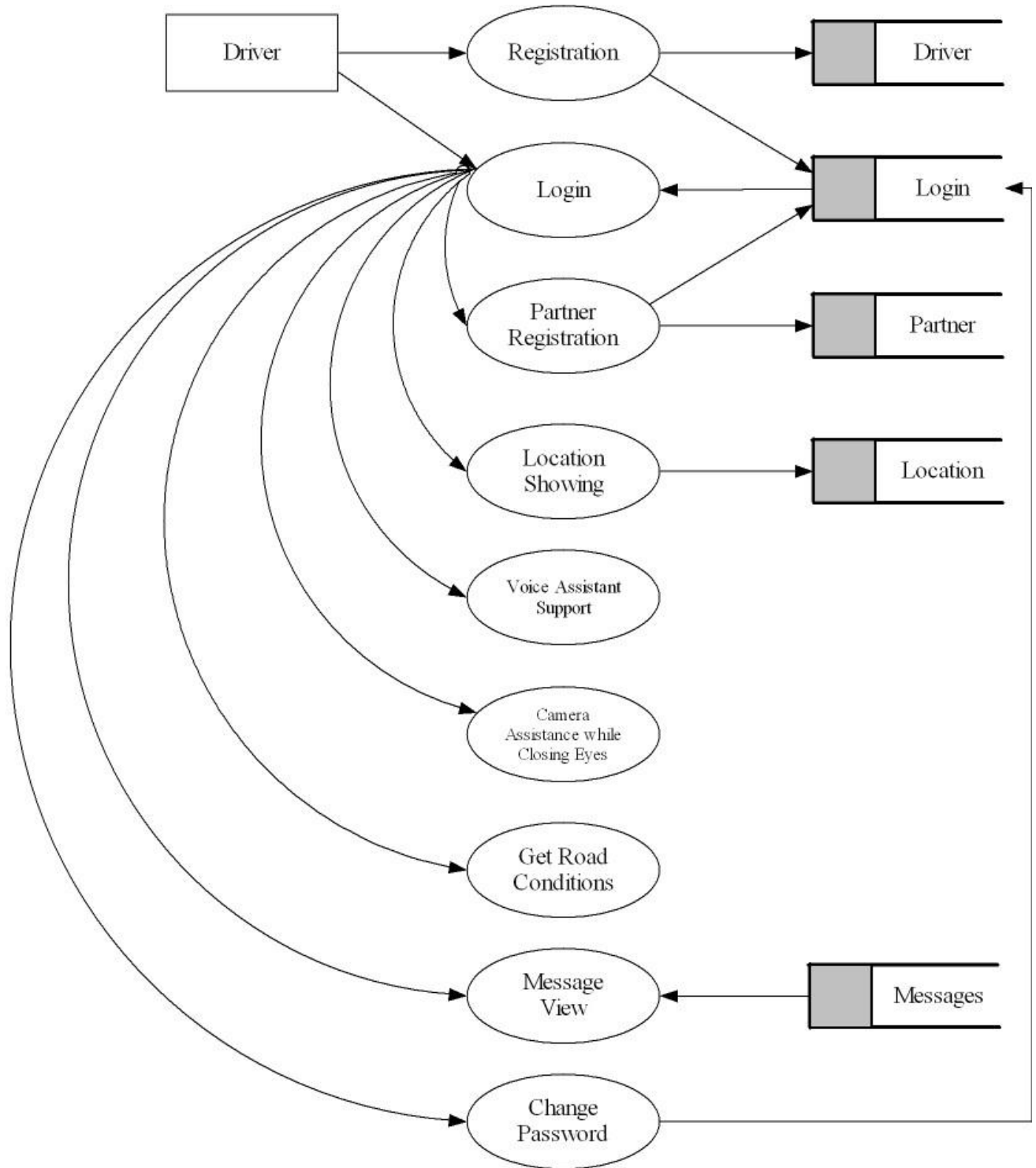
LEVEL 1.1 ADMIN



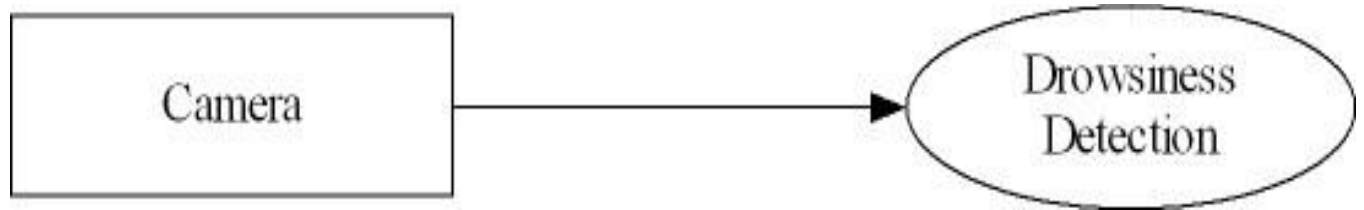
LEVEL 1.2 PARTNER



LEVEL 1.3 DRIVER



LEVEL 1.4 CAMERA



8.2 ENTITY RELATIONSHIP DIAGRAM

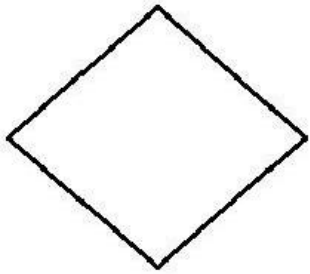
An Entity Relationship (ER) Diagram is a specialized graphics that illustrates the inter relationship between entities in a database. ER diagrams often use symbols to represent 3 different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. An entity Relationship model (ERM), in software engineering is an abstract and conceptual represent of data. Entity Relationship modelling is a relational schema database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relation database, and its requirements in a top- down fashion.

ENTITY:



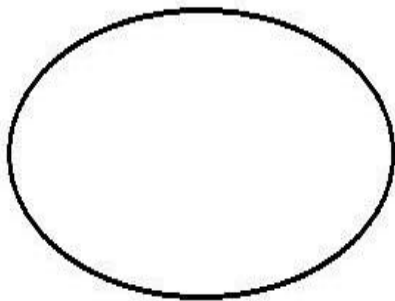
Entity is the thing which we want to store information. It is an elementary basic building block of storing information about business process. An entity represents an object defined within the information system about which you want to store information. Entities are distinct things in the enterprise.

RELATIONSHIPS:

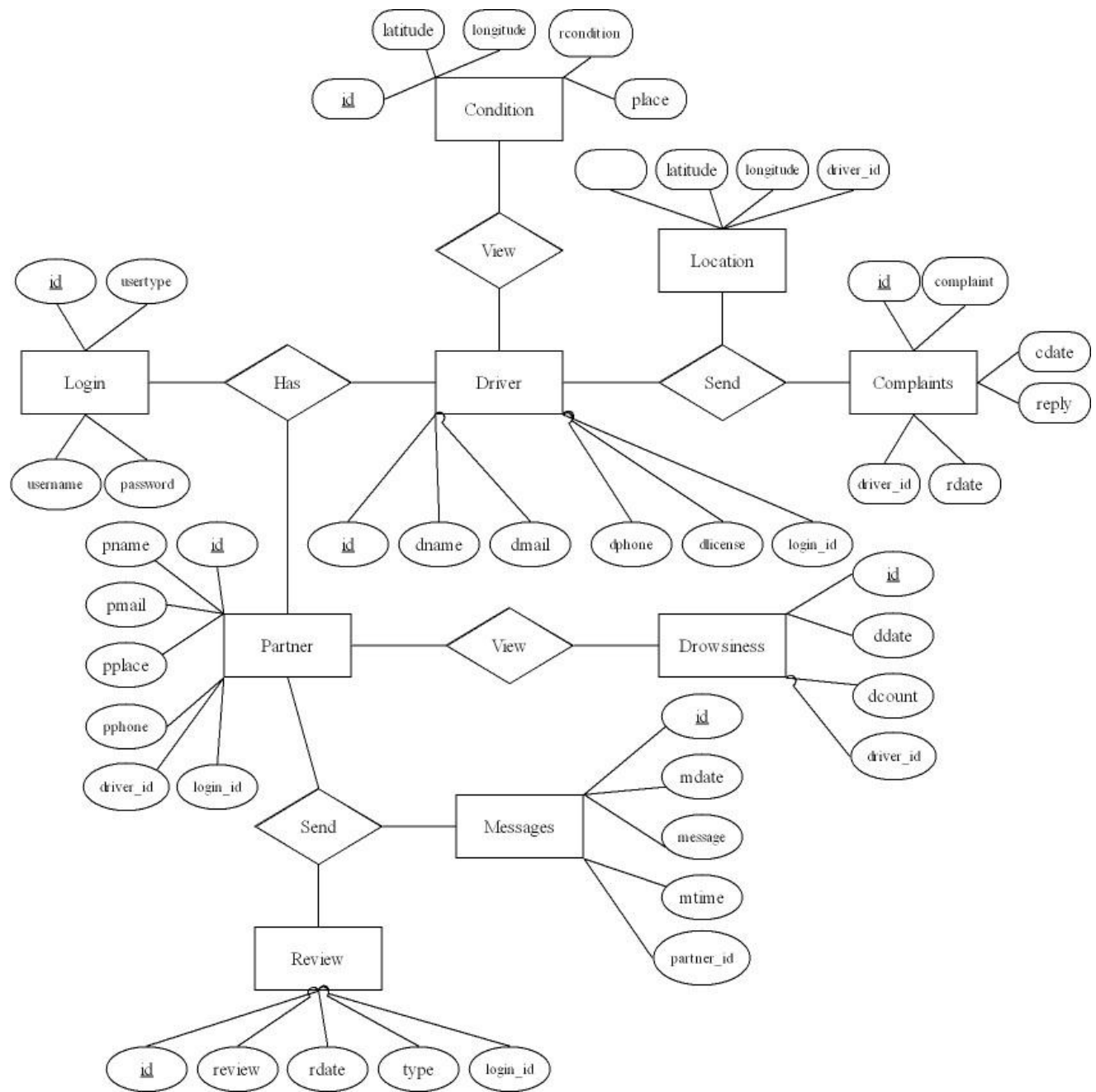


A relationship is a named collection or association between entities or used to relate two or more entities with some common attributes or meaningful interaction between the objects.

ATTRIBUTES:



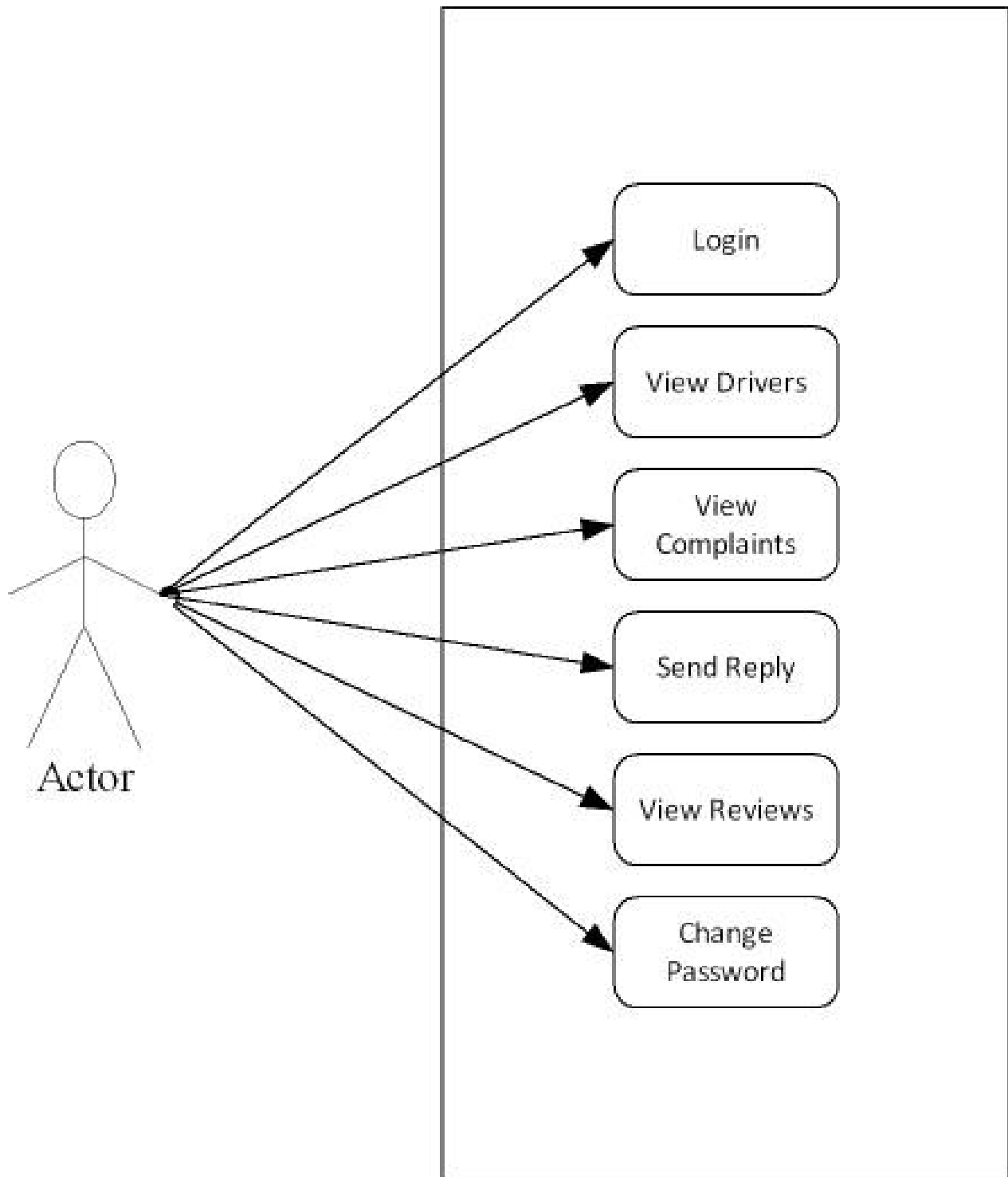
Attributes are the properties of the entities and relationship, Descriptor of the entity. Attributes are elementary pieces of information attached to an entity.



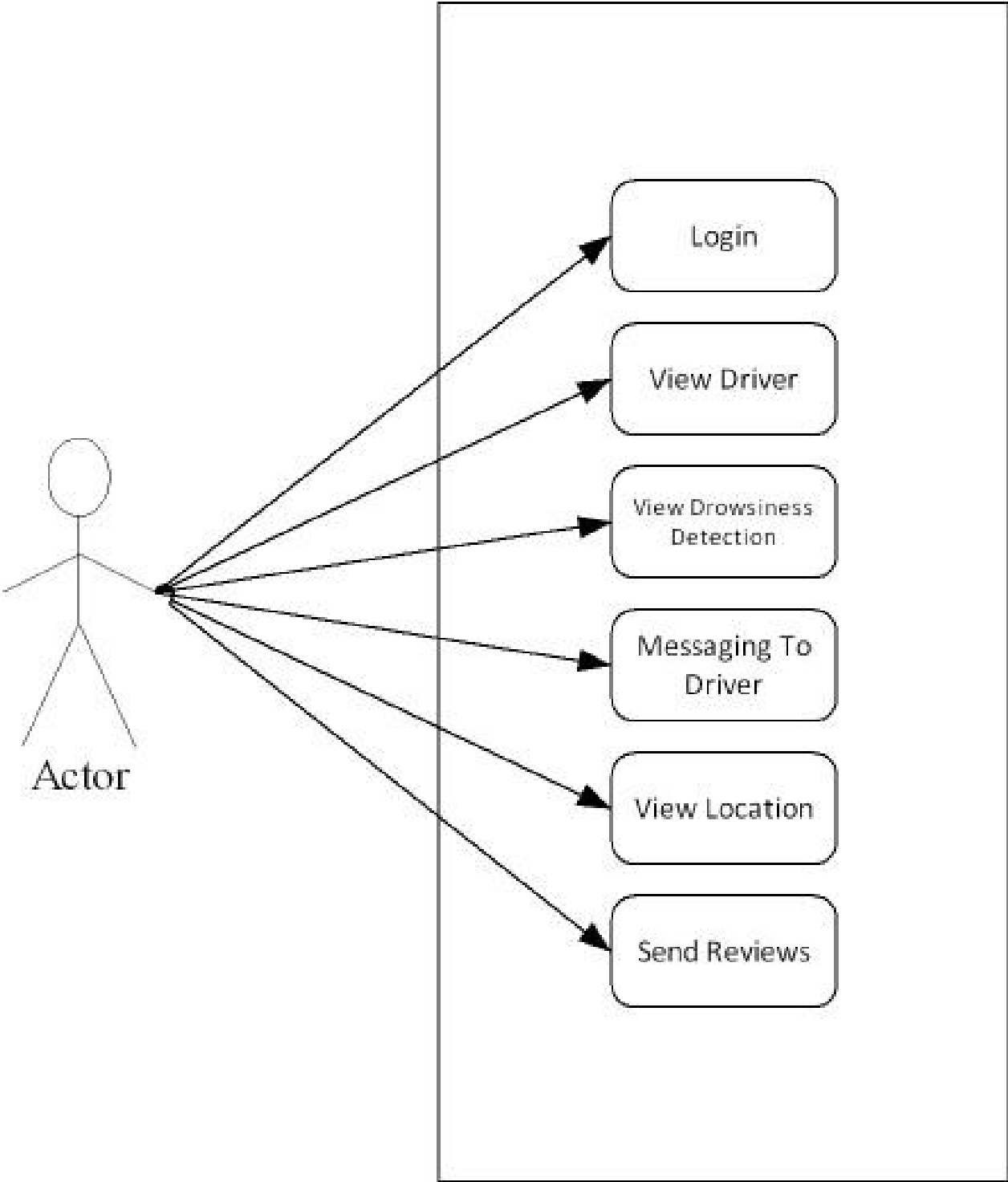
8.3 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system". They provide the simplified and graphical representation of what the system must actually do. Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. The purpose of the use case diagrams is simply to provide the high-level view of the system and convey the requirements in lay-people's terms for the stakeholders. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system. To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors. Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

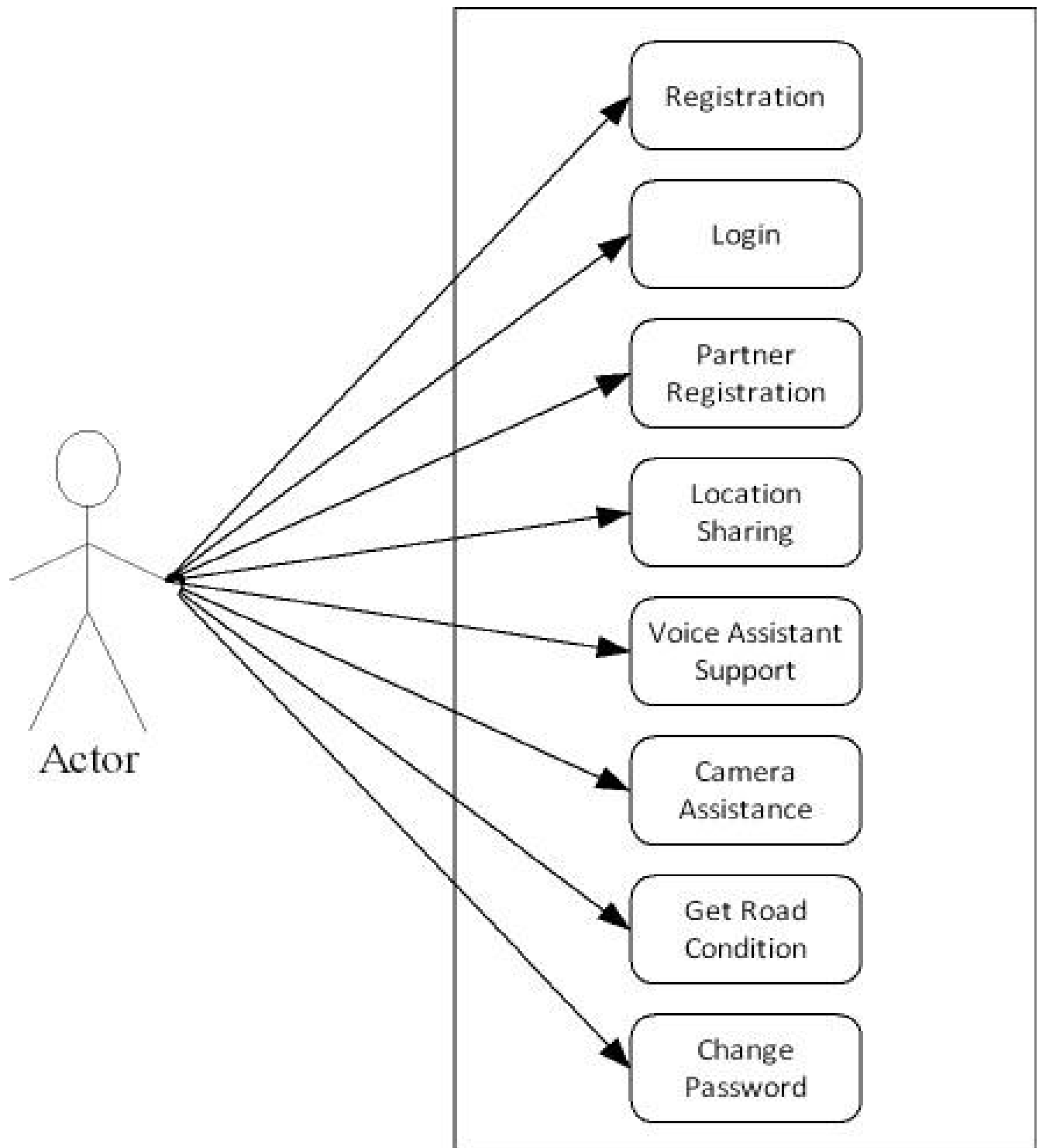
ADMIN



PARTNER



DRIVER



CAMERA



DATABASE DESIGN

9. DATABASE DESIGN

Database design is an important place in designing a system. During this phase care should be taken to avoid redundancy of information storing into a database, since it leads to wastage of memory space.

9.1 NORMALIZATION TECHNIQUES:

Normalization is a process of simplifying the relationship between data elements in a record. Through normalization a collection of data in a records structure is replaced by successive record structures that are simpler and more predictable and therefore more manageable.

First Normal Form

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or as it sometimes expressed, a repeating group.

Second Normal Form

A relation is said to be in second normal form if it is in first normal form and it should satisfy any one of the following rules. Primary key is not a composite primary key. No non-key attribute is fully functionally dependent on full set of primary keys.

Third Normal Form

A relation is said to be in third normal form if it is in second normal form and if there exists no transitive dependencies.

Transitive Dependency: If two non-key attributes depend on each other as well as on the primary key then they are said to be

transitively dependent.

The above normalization principles where applied to decompose the data in multiple tables there by making the data to be maintained in a consistent state.

Boyce-Code Normal Form

Even when the database is in Third Normal Form, still there would be anomalies resulted if it has more than one Candidate Key. Sometimes BCNF is also referred as 3.5 Normal Form.

TABLE DESIGN

LOGIN:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary key
username	VARCHAR(30)	Not Null
Password	VARCHAR(30)	Not Null
usertype	VARCHAR(10)	Not Null

DRIVER:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
dname	VARCHAR(10)	Not Null
dmail	VARCHAR(10)	Not Null
dphone	VARCHAR(10)	Not Null
dLicense	VARCHAR(10)	Not Null
LOGIN_id	INT	Foreign Key

PARTNER:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
pname	VARCHAR(10)	Not Null
pmail	VARCHAR(10)	Not Null
pplace	VARCHAR(10)	Not Null
pphone	VARCHAR(10)	Not Null
DRIVER_id	INT	Foreign Key
LOGIN_id	INT	Foreign Key

DROWSINESS:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
ddate	VARCHAR(10)	Not Null
dcount	VARCHAR(10)	Not Null
DRIVER_id	INT	Foreign Key

CONDITION:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
latitude	VARCHAR(10)	Not Null
longitude	VARCHAR(10)	Not Null
rcondition	VARCHAR(200)	Not Null
place	VARCHAR(10)	Not Null

MESSAGES:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
mdate	VARCHAR(10)	Not Null
message	VARCHAR(10)	Not Null
mtime	VARCHAR(200)	Not Null
PARTNER_id	INT	Foreign Key

LOCATION:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
latitude	VARCHAR(10)	Not Null
longitude	VARCHAR(10)	Not Null
DRIVER_id	INT	Foreign Key

REVIEW:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
review	VARCHAR(100)	Not Null
rdate	VARCHAR(10)	Not Null
Type	VARCHAR(10)	Not Null
LOGIN_id	INT(20)	Foreign Key

COMPLAINTS:

COLUMN NAME	DATA TYPE	KEY
id	INT	Primary Key
complaint	VARCHAR(10)	Not Null
cdate	VARCHAR(10)	Not Null
reply	VARCHAR(100)	Not Null
rdate	VARCHAR(10)	Not Null
DRIVER_id	INT	Foreign Key

10. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limiting type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

10.1 TEST TYPES

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

UNIT TESTING

In unit testing different modules are tested against the specifications produced during design phase for the modules in the project and the goal is to test the internal logic of the modules. In order to perform the unit testing, the best approach we adopted in this project is functional testing in which inputs are given to the system for which the expected results are known, boundary values and special values. For this the module selected was advance details where the balance amount falls to negative indicating there is no more payment required. Secondly performance testing was done to determine the amount of execution time spent in various aspects like the module, program throughput, response time and the device utilization by the program unit.

INTEGRATION TESTING

The primary goal of the integration testing is to see if the modules can be integrated properly. The integration testing is therefore considered as testing the design. Thus, in the integration testing step, all the errors uncovered are connected for the next testing steps.

SYSTEM TESTING

System testing is similar to integration testing, but instead of integrating modules into programs for testing, programs are integrated into systems for testing the interfaces between programs in a system. System testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that can be reasonably expected by the customer.

ACCEPTANCE TESTING

Once the system test has been satisfactory completed, the system is ready for acceptance testing. Acceptance testing is the process whereby actual users test a completed information system in a environment where it will eventually be used, the end result of which is the users acceptance or rejection. The admin and staff at company accepted proposed system after testing.

10.2 TESTING METHODS

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

BLACK BOX TESTING

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-

based testing, traceability matrix, exploratory testing and specification-based testing. The black box tester has no "bonds" with the code, and a tester's perception is very

Simple a code must have bugs. Using the principle, "Ask and you shall receive," blackbox testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.

WHITE BOX TESTING

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these. The following types of white box testing exist:

API testing (application programming interface) - Testing of the application using Public and Private APIs.

Code coverage - creating tests to satisfy some criteria of code coverage. For example, the test designer can create tests to cause all statements in the program to be executed at least once.

- Fault injection methods.
- Mutation testing methods.
- . Static testing - White box testing includes all static testing.

GREY BOX TESTING

Grey box testing involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as grey box, because the input and output are clearly outside of the "black-box" that we are calling the system under test. This distinction is Particularly

important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test. However, modifying a data repository does qualify as grey box, as the user would not normally, not be able to change the data outside of the system under test. Grey box testing may also include reverse engineering.

IMPLEMENTATION

11. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can cause chaos and confusion. Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be totally new, replacing an existing manual or automated system or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will be preventing. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover. The system implementation has three main aspects. They are education and training, system testing and changeover. The implementation stage involves following tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.
- Evaluation of the changeover method.

- The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users are trained in the new procedures.

11.1 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage, they doubt about the software but we have to ensure that the resistance does not build up as one has to make sure that

- The active user must be aware of the benefits of using the system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

SYSTEM MAINTENANCE

12. SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which a Software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affects a system, which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”. We may define maintenance by describing four activities that are undertaken to after a program is released for use.

12.1 CORRECTIVE MAINTAINANCE

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called corrective maintenance.

12.2 ADAPTIVE MAINTENANCE

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, adaptive maintenance an activity that modifies software to properly interfere with a changing environment is both necessary and common place.

12.3 PERFECTIVE MAINTENANCE

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancements are received from users. To satisfy requests in this category, perfective maintenance is performed.

12.4 PREVENTIVE MAINTENANCE

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

TIME LINE CHART

13. TIME LINE CHART

Analysis	Start Date	End Date
System analysis	24/08/2023	15/09/2023
System study	20/10/2023	02/12/2023
Requirement Analysis	06/01/2024	18/02/2024

FUTURE ENHANCEMENT

14. FUTURE ENHANCEMENT

The future of “IONROAD” is exciting because new technologies can be used to make them even better. For example, One way to combat the negative effects of mobile phone distraction is to integrate a new feature into projects that give warnings when users are utilizing their mobile phones. This feature could function by detecting when a user is using their mobile phone and providing a gentle reminder to put it away and focus on the task at hand.

And also an extra feature that will utilize advanced sensors and facial recognition software to detect any unfamiliar individuals in the vehicle. This will not only help prevent potential car thefts, but also alert drivers to any potential danger or threat. With this new feature, drivers can rest assured that their car is always under surveillance, providing them with a sense of security and control. Our project aims to make driving a safer and more secure experience for everyone.

CONCLUSION

15. CONCLUSION

In conclusion, the project "IONROAD" has emerged as a groundbreaking solution to addressing the critical issues of drowsiness and pothole-related accidents that plague drivers on the road. Through the innovative use of advanced technology and artificial intelligence, IONROAD has succeeded in providing drivers with a reliable tool to enhance their safety and prevent potentially disastrous outcomes.

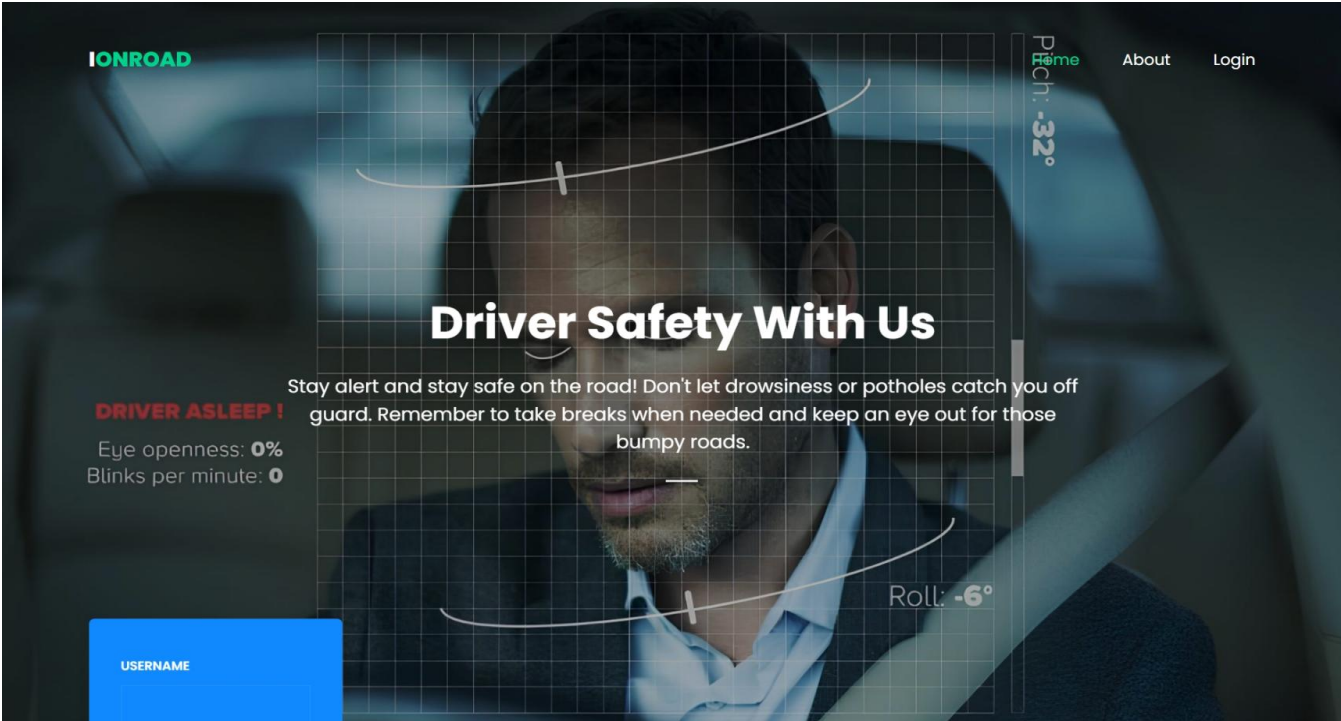
One of the key strengths of IONROAD lies in its ability to detect and alert drivers of impending dangers such as drowsiness and potholes in real-time. By leveraging the power of smart phones and sensors, the system is able to monitor and analyze the driver's behavior and the road conditions, thus providing timely warnings and prompts to avert potential accidents. This proactive approach not only enhances the safety of the driver but also contributes to the overall reduction of road accidents and fatalities.

Moreover, the user-friendly interface of IONROAD makes it accessible and easy to use for drivers of all experience levels. The clear and concise alerts provided by the system are designed to be easily understood and acted upon, ensuring that drivers can respond quickly and effectively to potentially hazardous situations. This intuitive design not only enhances the user experience but also encourages drivers to remain vigilant and alert while on the drive.

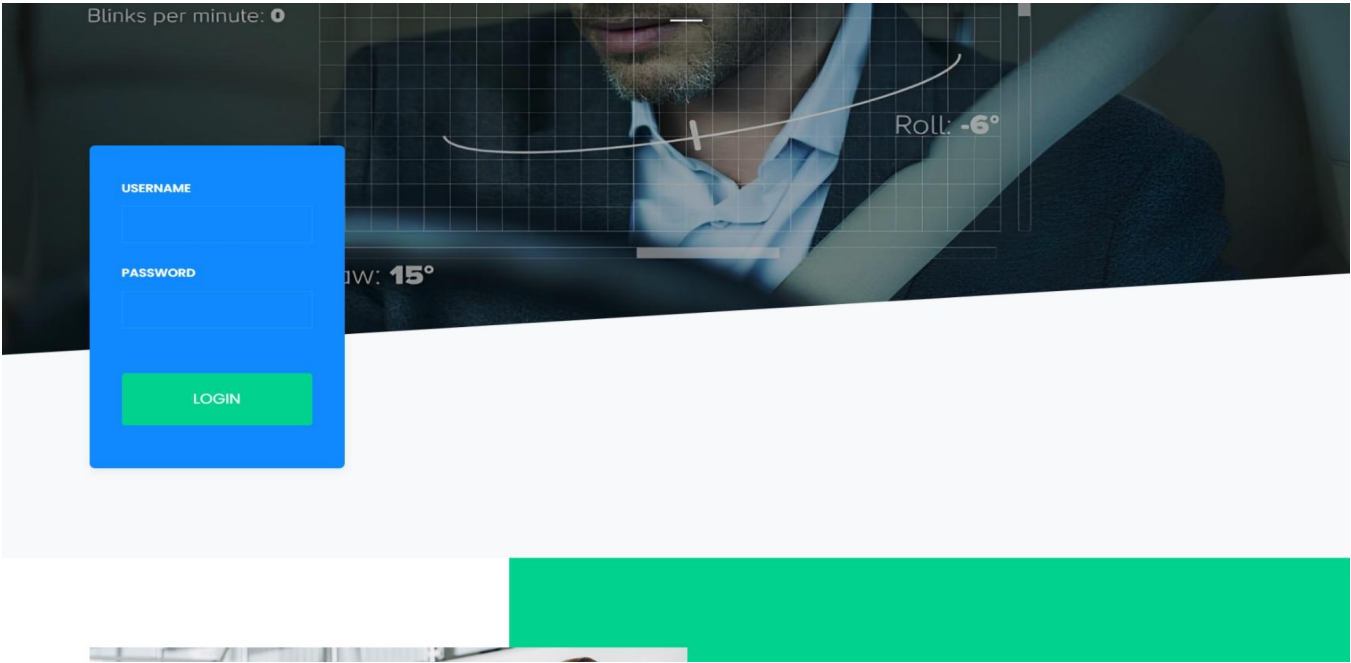
APPENDIX

16. SCREENSHOTS

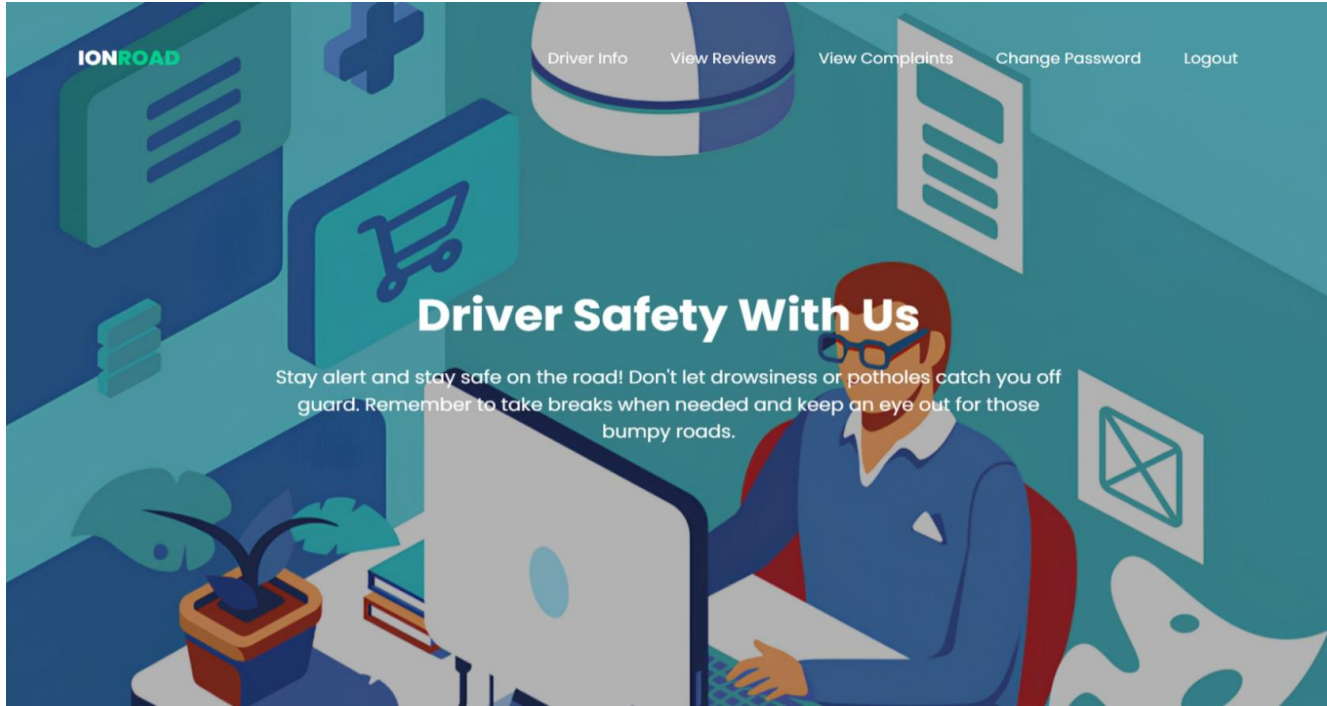
HOME



LOGIN



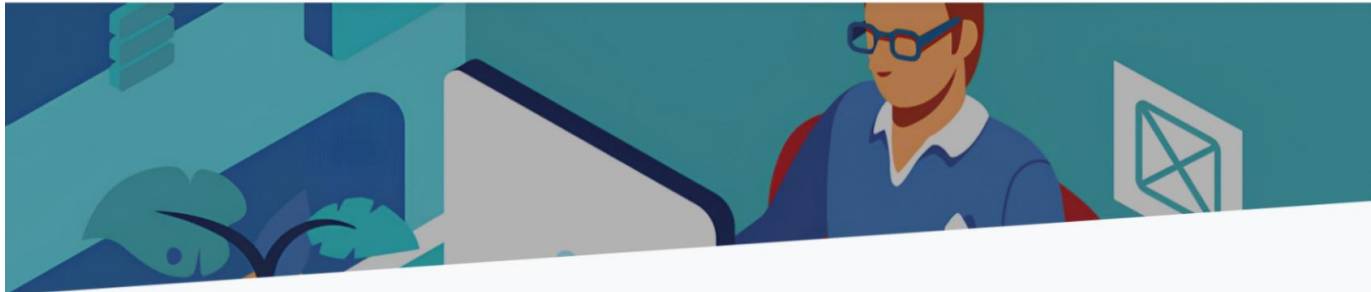
ADMIN HOME



DRIVERS INFO

IONROAD

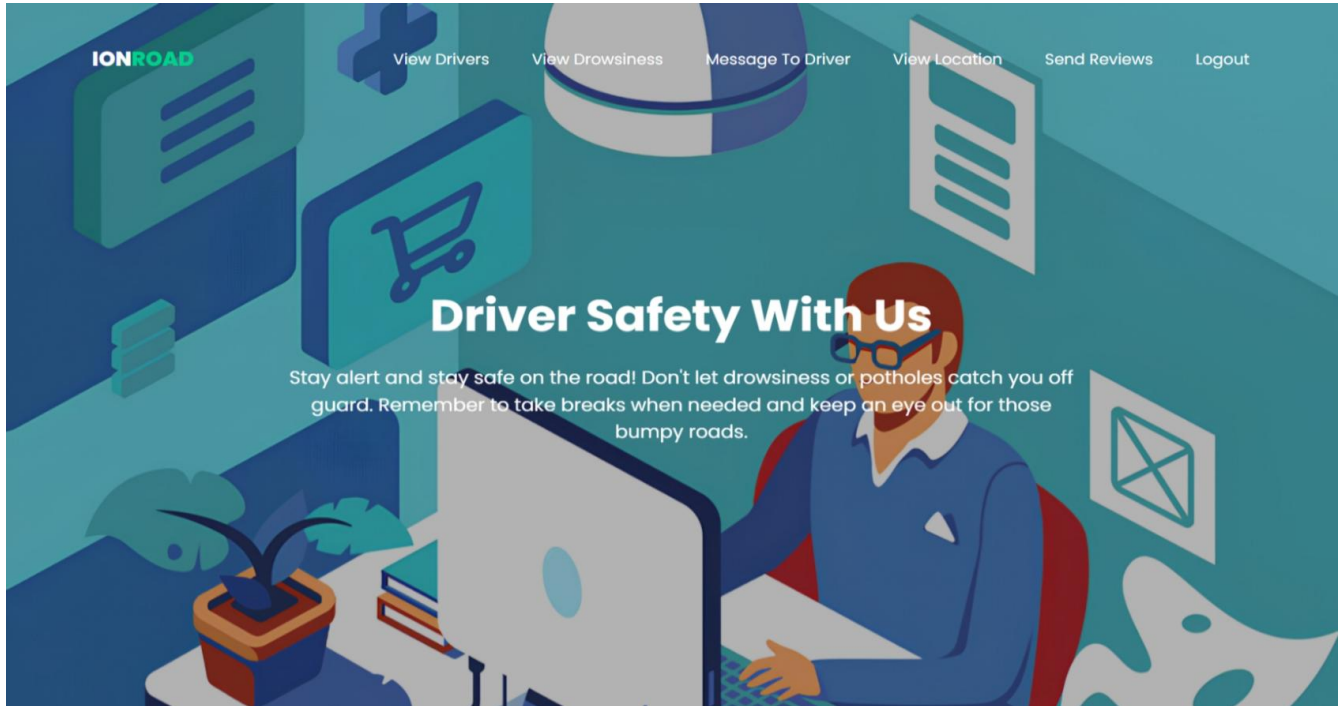
Driver InfoView ReviewsView ComplaintsChange PasswordLogout



Driver Info

#	Name	Email	Phone	License no	Location
1	najil	najilmp@gmail.com	12345	12345	Open

PARTNER HOME



VIEW PARTNER'S DRIVER

IONROAD[View Drivers](#)[View Drowsiness](#)[Message To Driver](#)[View Location](#)[Send Reviews](#)[Logout](#)

Driver Info

Name	najil
Email	najilmp@gmail.com
Phone	12345
License no	12345

VIEW DROWSINESS DETECTED INFO

IONROAD

View Drivers

View Drowsiness

Message To Driver

View Location

Send Reviews

Logout

Detection Info

#	Detected Date&Time	Detected Count
1	2024-02-21	1
2	2024-02-23	1
3	2024-02-29	1
4	2024-03-01	4

MESSAGE TO DRIVER

IONROAD

View Drivers

View Drowsiness

Message To Driver

View Location

Send Reviews

Logout

Detection Info

hiiii

hiiiiiii


helo

helo

najil

good morning

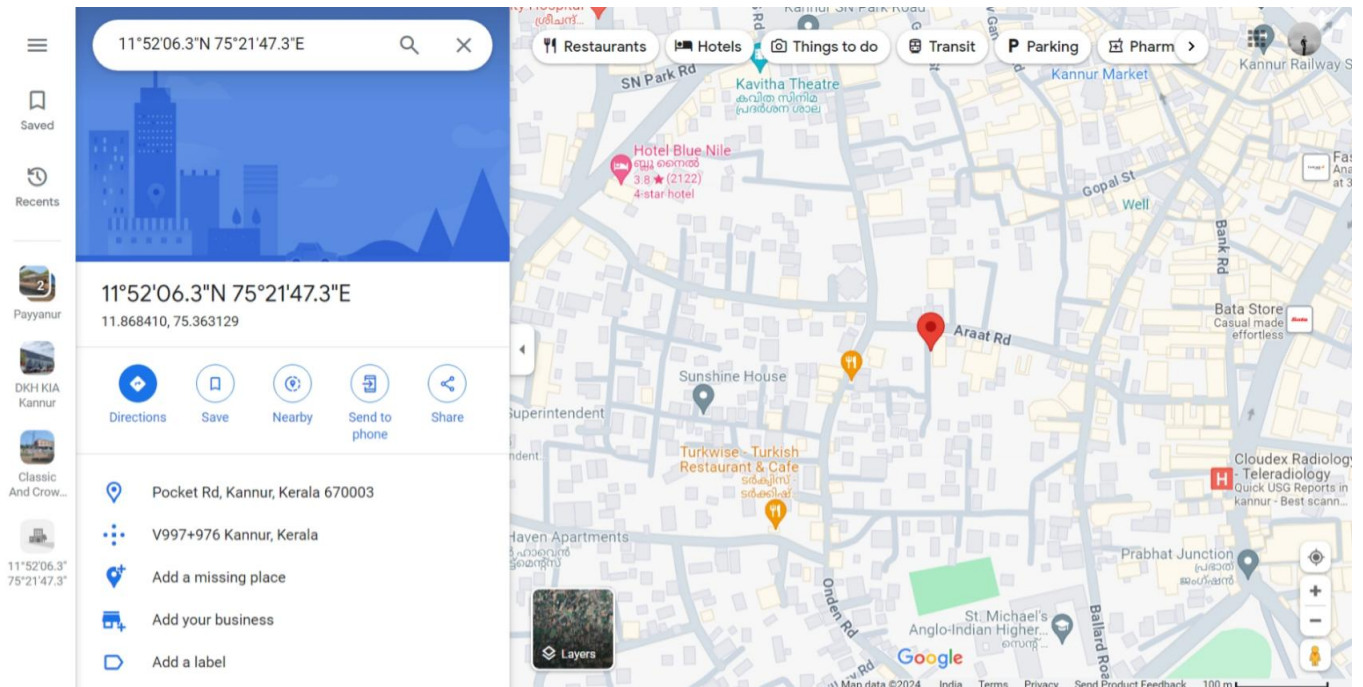
good morning



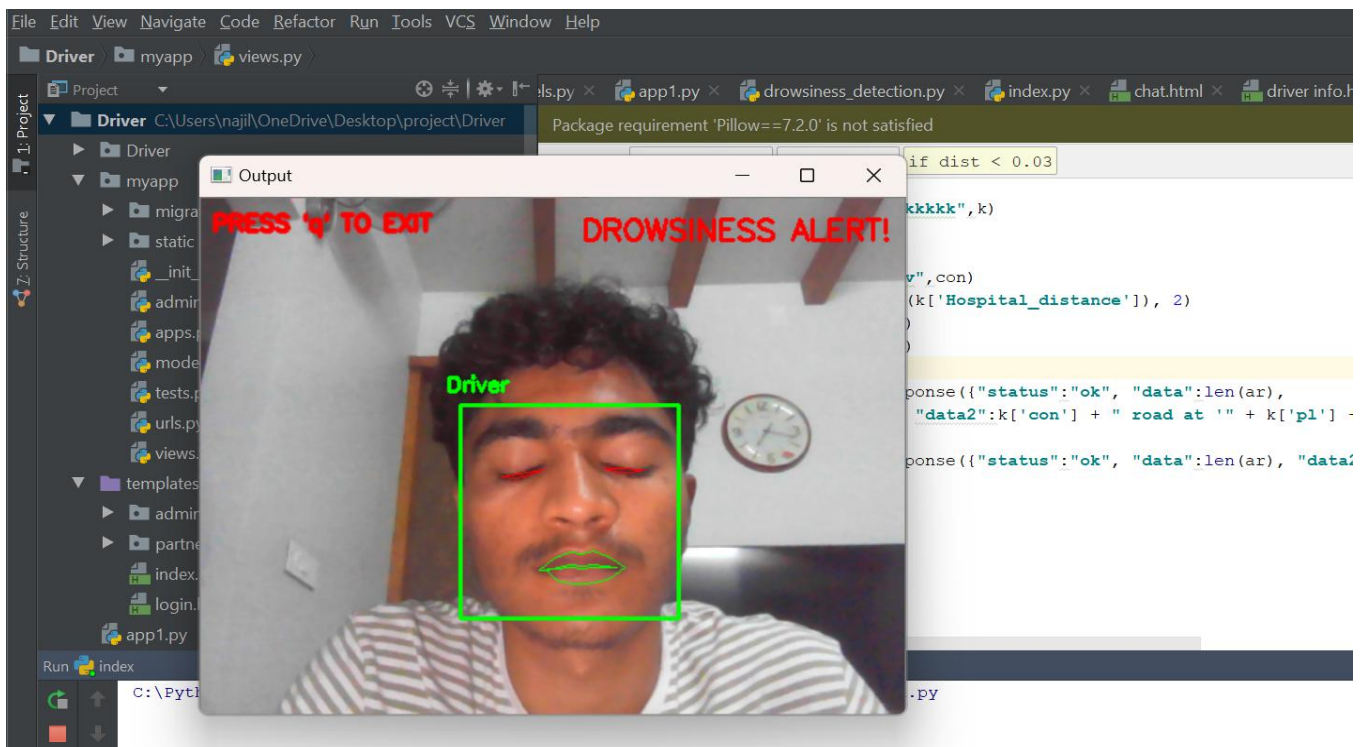
good morning

chat

VIEW DRIVER LOCATION

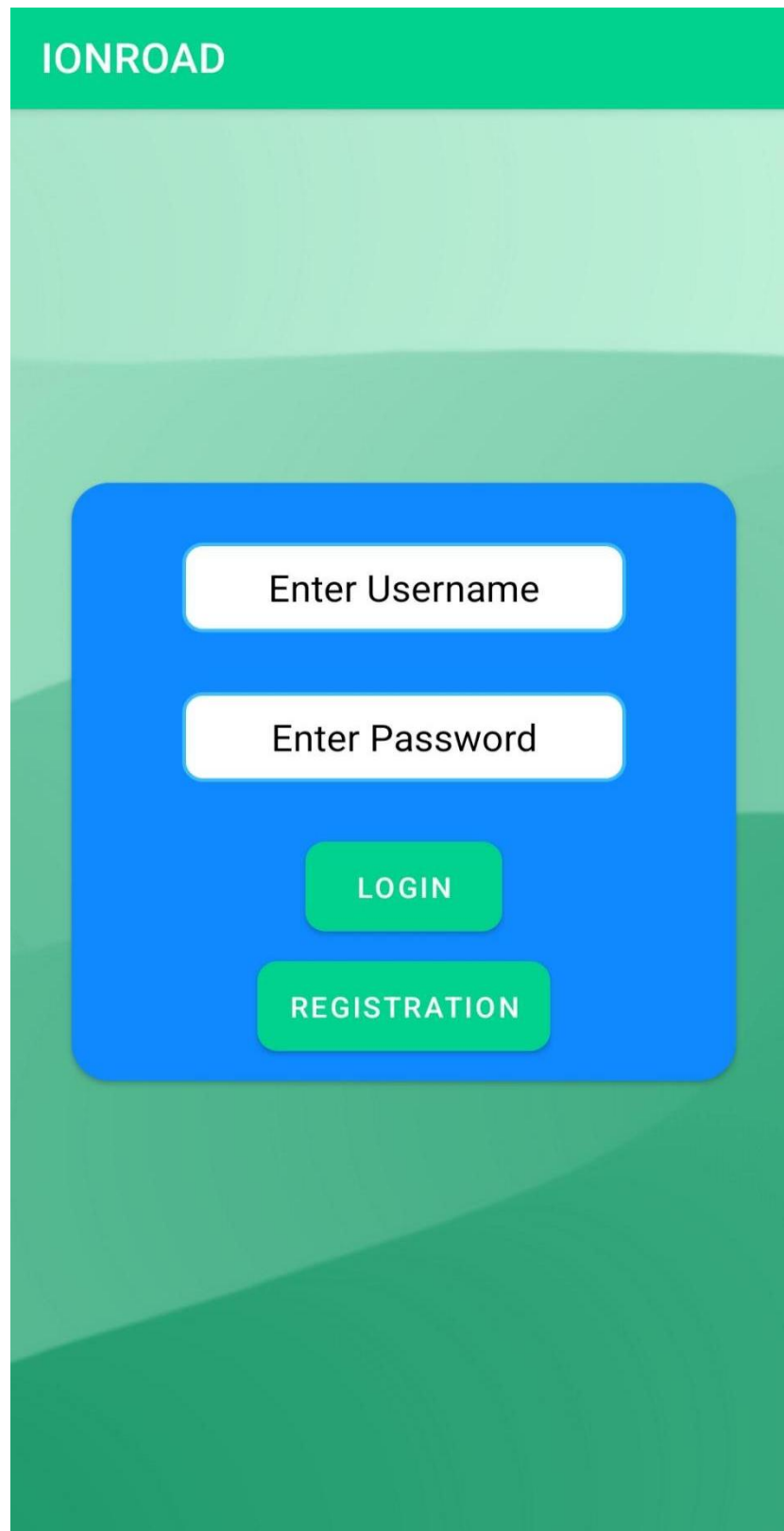


DROWSINESS DETECTION



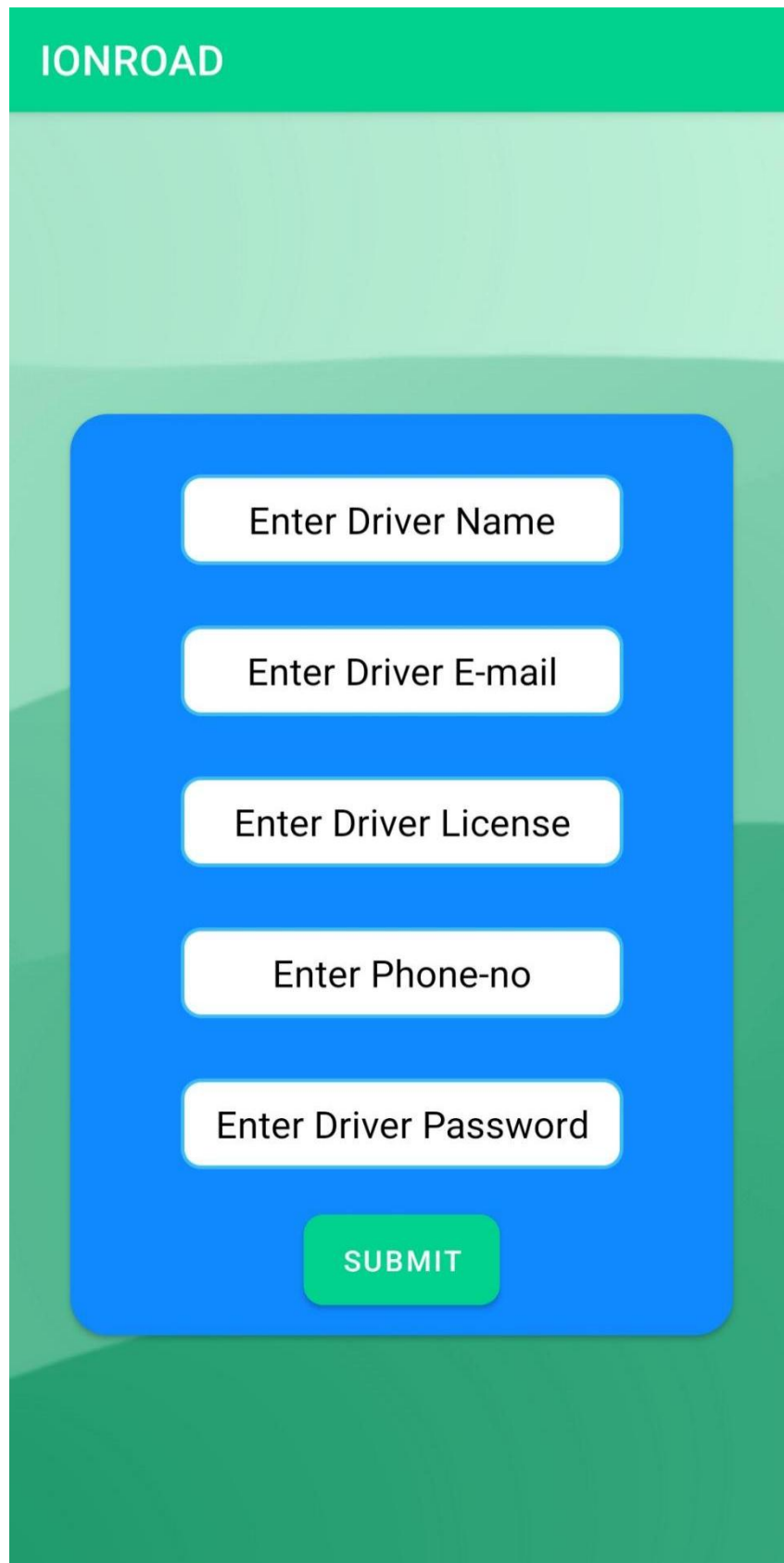
SCREENSHOTS(ANDROID)

USER LOGIN



A screenshot of the IONROAD mobile application's user login screen. The app has a green header bar with the text "IONROAD" in white. The background of the screen is a light green gradient with a subtle mountain silhouette. In the center, there is a blue rounded rectangle containing the login form. The form consists of two white input fields with rounded corners: the top one is labeled "Enter Username" and the bottom one is labeled "Enter Password". Below these fields are two green buttons with rounded corners: the top one is labeled "LOGIN" and the bottom one is labeled "REGISTRATION".

REGISTRATION



The image shows a registration form for 'IONROAD'. The form is displayed on a mobile device screen with a green header bar at the top containing the text 'IONROAD'. Below the header, the background is a light green gradient. The registration form itself is a blue rounded rectangle centered on the screen. It contains five white input fields with rounded corners, each with a placeholder text: 'Enter Driver Name', 'Enter Driver E-mail', 'Enter Driver License', 'Enter Phone-no', and 'Enter Driver Password'. At the bottom of the blue form is a green rounded rectangle button with the text 'SUBMIT' in white capital letters.

IONROAD

Enter Driver Name

Enter Driver E-mail

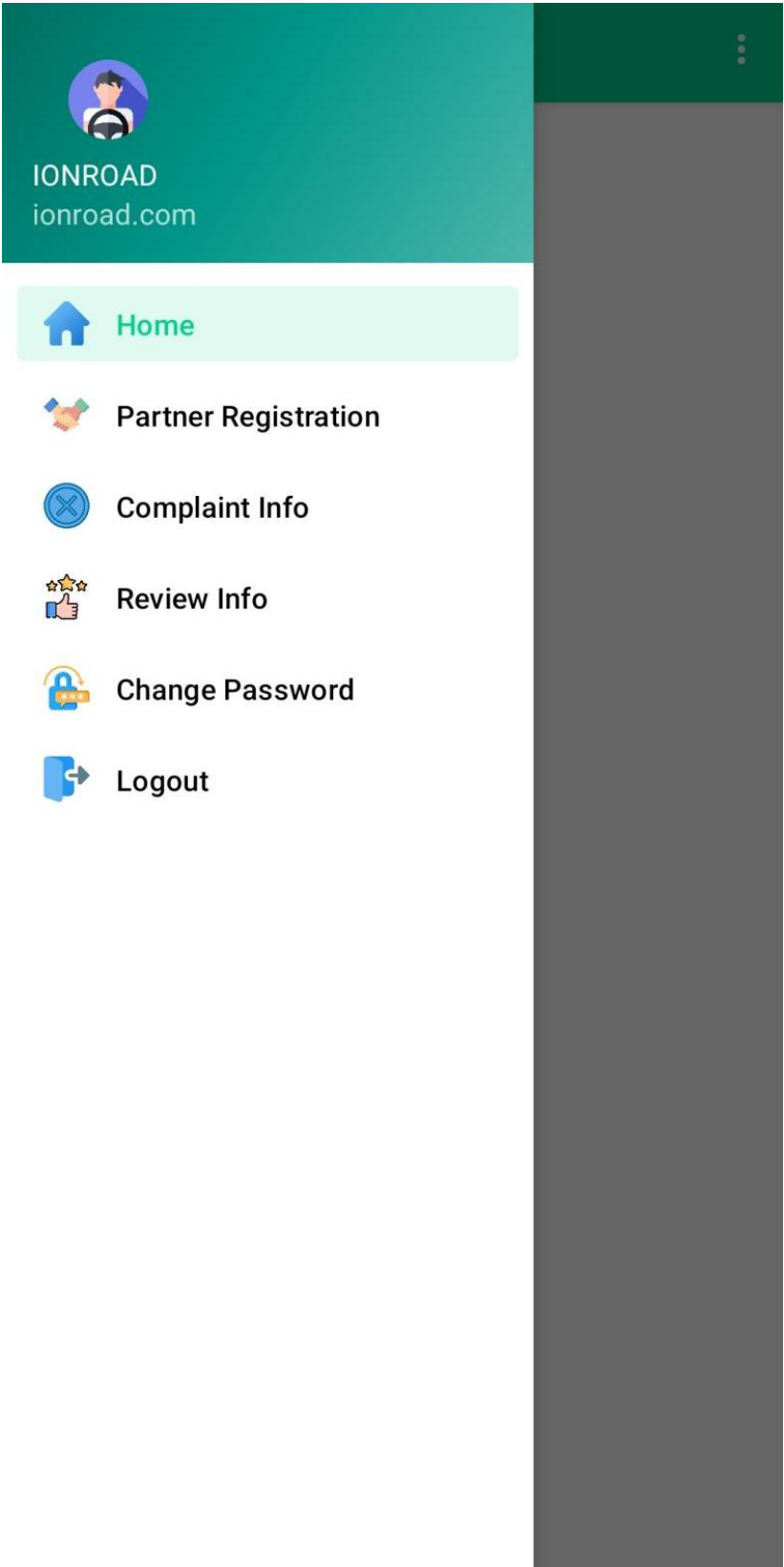
Enter Driver License

Enter Phone-no

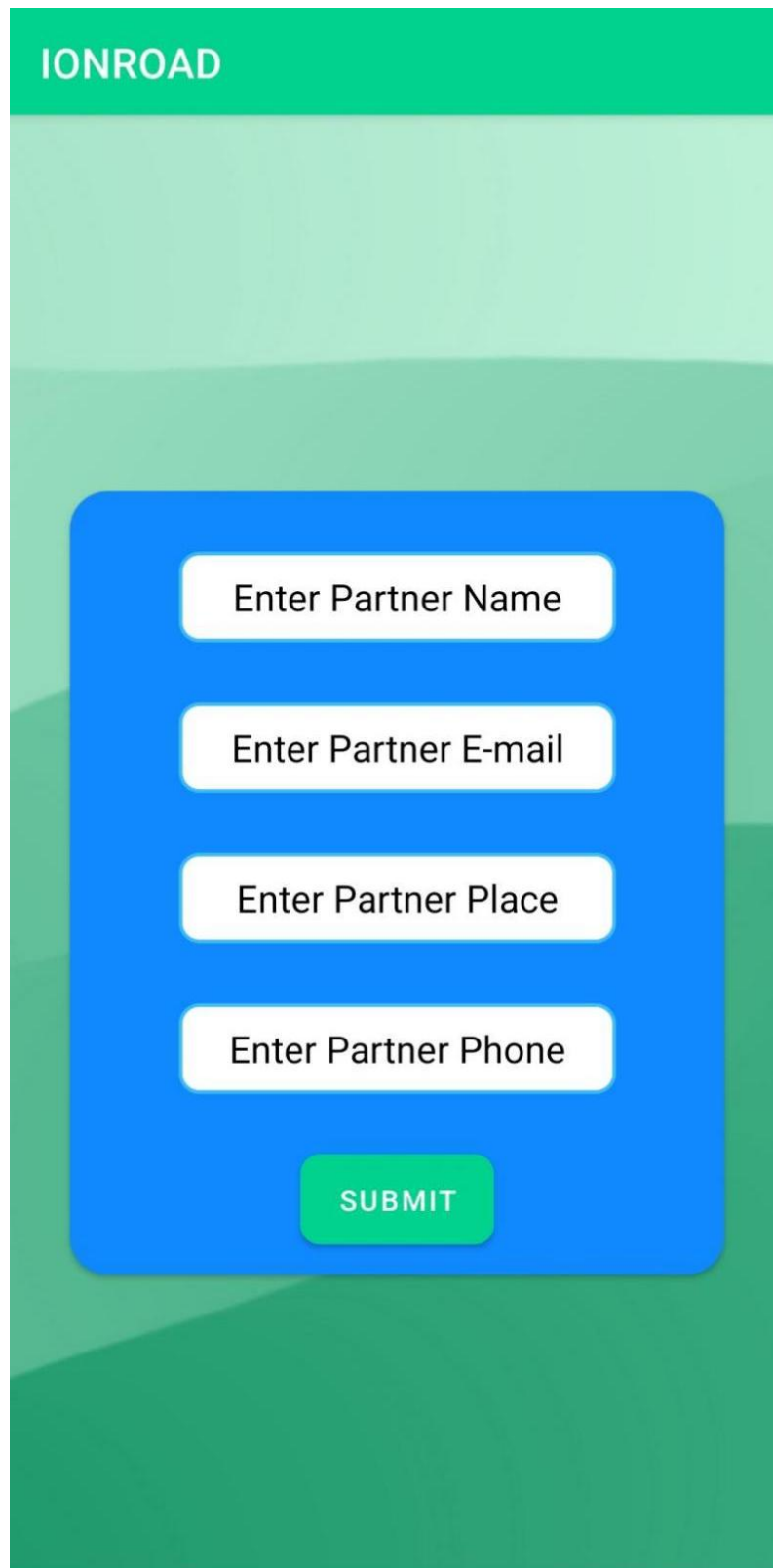
Enter Driver Password

SUBMIT

HOME PAGE



PARTNER REGISTRATION



The image shows a mobile application interface for partner registration. At the top, a green header bar contains the text "IONROAD" in white. Below this is a light green background with a subtle landscape pattern. In the center, there is a blue rounded rectangle containing four white input fields and a green submit button. The input fields are labeled "Enter Partner Name", "Enter Partner E-mail", "Enter Partner Place", and "Enter Partner Phone". The submit button is labeled "SUBMIT" in white capital letters.

IONROAD

Enter Partner Name

Enter Partner E-mail

Enter Partner Place

Enter Partner Phone

SUBMIT

COMPLAINT INFO

IONROAD

Complaint

App always crashes.

Complaint Date

2024-03-02

Reply

pending

Reply Date

pending

Complaint

I've noticed that there are a few bugs that pop up every now and then.

Complaint Date

2024-03-02

Reply

pending

Reply Date

pending

Complaint

The registration submit button doesnt work sometimes.

Complaint Date

2024-03-02

Reply

pending

+

17.CODE

HOME

```
import os
import random
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from django.db.models.expressions import RawSQL
from django.http import HttpResponseRedirect, JsonResponse
from django.shortcuts import render, redirect

# Create your views here.
from myapp.models import *
from datetime import datetime

def login(request):
    return render(request, 'index.html')

def login_post(request):
    u = request.POST['textfield']
    p = request.POST['textfield2']
    obj = Login.objects.filter(username=u, password=p)
    if obj.exists():
        obj = obj[0]
        request.session['head']=" "
        request.session['log']="lo"
    if obj.usertype=='admin':
        request.session['lid'] = obj.id
        return redirect('/admin_home')
    if obj.usertype=='partner':
        request.session['lid'] = obj.id
        return redirect('/partner_home')
    else:
        return HttpResponseRedirect('<script>alert("Username or Password is Incorrect!");window.location="/"</script>')
```

ADMIN

```
def admin_home(request):
    return render(request, 'admin/admin_index.html')

def view_review(request):
    if request.session['log'] != "lo":
        return HttpResponse('<script>alert("Logout Successfully");window.location="/"</script>')
    request.session['head'] = "View Reviews"
    return render(request, 'admin/Review.html')

def view_complaint(request):
    if request.session['log'] != "lo":
        return HttpResponse('<script>alert("Logout Successfully");window.location="/"</script>')
    request.session['head'] = "View Complaints"
    obj1 = Complaints.objects.all()
    return render(request, 'admin/complaint.html', {"data": obj1})

def change_password(request):
    if request.session['log'] != "lo":
        return HttpResponse('<script>alert("Logout Successfully");window.location="/"</script>')
    request.session['head'] = "Change Password"
    return render(request, 'admin/changepass.html')
```

PARTNER

```
def partner_home(request):
    try:
        drv =
Driver.objects.get(id=Partner.objects.get(LOGIN=request.sess
        loc = Location.objects.get(DRIVER=drv.id)
        request.session['l'] = loc.latitude
        request.session['lg'] = loc.longitude
        return render(request, 'partner/partner_index.html',
{'data': loc})
    except Exception as e:
        print(e, "eeeeeeeeeeeeeeeeeeeeeeeeeeee")

        return render(request, 'partner/partner_index.html',
{'data': "2"})

def partner_driverinfo(request):
    if request.session['log'] != "lo":
        return HttpResponseRedirect('<script>alert("Logout
Successfully");window.location="/"</script>')
    request.session['head'] = "Driver Info"
    data=Partner.objects.get(LOGIN=request.session['lid'])
    return render(request, 'partner/driver
info.html', {"i":data})

def chatsnd(request):
    import datetime
    d=datetime.datetime.now().strftime("%Y-%m-%d")
    t=datetime.datetime.now().strftime("%H:%M:%S")
    c = request.session['lid']
    b=request.POST['e']
    obj=Messages()
    obj.PARTNER=Partner.objects.get(LOGIN=c)
    obj.message=b
    obj.mdate=d
    obj.mtime=t
    obj.save()
    return redirect('/chatt#adhom')

def partner_send_review_post(request):
    Reviews=request.POST['textarea']
```

```
obj=Review()  
obj.LOGIN_id=request.session['lid']  
obj.review=Reviews  
obj.type = 'partner'  
obj.rdate=datetime.now().strftime("%y-%m-%d")  
obj.save()  
return HttpResponse('<script>alert("Review Sent  
Successfully");window.location="/partner_home"</script>')
```

DRIVER

```
package com.example.driver_drowsinessandroid;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class login extends AppCompatActivity {
    EditText e1,e2;
    Button b1,b2;
    SharedPreferences sh;
    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        e1 = findViewById(R.id.editTextTextPersonName2);
        e2 = findViewById(R.id.editTextTextPassword);
        b1 = findViewById(R.id.button2);
        b2 = findViewById(R.id.button8);
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(),
registration.class));
            }
        });
        sh =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```

String user = e1.getText().toString();
String pass = e2.getText().toString();
if(user.isEmpty()){
    e1.setError("Enter User Name");
    return;
}
if(pass.isEmpty()){
    e2.setError("Enter Password");
    return;
}
else {

    String hu = sh.getString("ip", "");
    String url = "http://" + hu + ":8000/android_login";
    Toast.makeText(login.this, "", Toast.LENGTH_SHORT).show();

    RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationContext());
    StringRequest postRequest = new
StringRequest(Request.Method.POST, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                // Toast.makeText(getApplicationContext(),
response, Toast.LENGTH_LONG).show();

                // response
                try {
                    JSONObject jsonObj = new
JSONObject(response);

                    if
(jsonObj.getString("status").equalsIgnoreCase("ok")) {
                        String id=jsonObj.getString("lid");
                        SharedPreferences.Editor ed=sh.edit();
                        ed.putString("lid",id);
                        ed.commit();
                        Intent intent = new
Intent(getApplicationContext(), MainActivity2.class);
                        startActivity(intent);
                        Intent i=new
Intent(getApplicationContext(),gpstracker.class);
                        startService(i);
                        Intent ij=new
Intent(getApplicationContext(),Shakeservice.class);
                        startService(ij);
                    }
                    else {
                        Toast.makeText(getApplicationContext(),
"Not found", Toast.LENGTH_LONG).show();
                    }

                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(),
"Error" + e.getMessage().toString(), Toast.LENGTH_SHORT).show();
                }
            }
        })
    requestQueue.add(postRequest);
}

```

```

    }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            // error
            Toast.makeText(getApplicationContext(),
error.getMessage().toString(), Toast.LENGTH_SHORT).show();
        }
    }
) {
    @Override
    protected Map<String, String> getParams() {
        SharedPreferences sh =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
        Map<String, String> params = new HashMap<String,
String>();

        String id=sh.getString("uid","");
        params.put("uid",id);
        params.put("user",user);
        params.put("pass",pass);
        params.put("lt", Locationservice.lati);
        params.put("lg", Locationservice.logi);
        return params;
    }
};

int MY_SOCKET_TIMEOUT_MS=100000;

postRequest.setRetryPolicy(new DefaultRetryPolicy(
    MY_SOCKET_TIMEOUT_MS,
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
requestQueue.add(postRequest);

}

}

});
}

```