**CHAPTER 1**

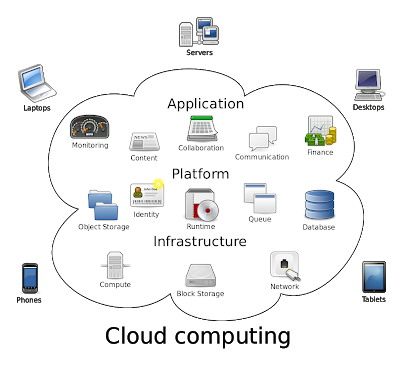
**INTRODUCTION**

* 1. **PROJECT OVERVIEW**

The popular trend in today's technology driven world is ‘Cloud Computing’. Cloud computing can be referred to as the storing and accessing of data over the internet rather than your computer's hard drive. This means you don't access the data from either your computer's hard drive or over a dedicated computer network (home or office network). Cloud computing means data is stored at a remote place and is synchronized with other web information.

Cloud computing is an Information Technology (IT) paradigm that enables ubiquitous access to shared pools of configurable system resources and higher level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing on resources to achieve coherence and economic of scale, similar to a utility over a network. Cloud computing, or in simpler short hand just “The cloud”, also focuses maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This approach should maximize the use of computing power thus reducing environmental damage as well since power less, air conditioning, rack space, etc., are required for variety of functions.

Third party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Advocates note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand. Cloud providers typically use a “pay-as-you-go” model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud pricing models.



Public Transport Management System keeps track of a Passengers Travel history. This information is accessible Nationwide with the Unique ID. This application makes the maintenance of record simple and handy. This System enables ease of access to the Users.

* 1. **AIM:**

A Framework which stores the travel details of Users. The admin maintains the Bus route details in cloud storage. Previously, Most of the past travel and trip details are lost by the users. These travel details can be collected and stored in Cloud storage so that there is no loss of data. The details of the Passengers can be accessed based on access rights by the passengers. Everything is Digitalized.

**PROBLEM DEFINITION**

The idea is to eliminate paper tickets and introduce E-Tickets even for local town bus and make the public transport service digitalize.

* 1. **OBJECTIVE:**

The main objective of the project is to digitalize the public transport system and enable passengers to access and acquire E-tickets even for Local town bus. By storing data in digital form helps to reduce the effort made in maintenance and verification. Registration and login module will be available for the Users to maintain a separate authenticated access. QR code plays a vital role in the application as the identification of bus as well the Ticket generated are QR codes .

**CHAPTER 2**

**SYSTEM ANALYSIS**

System analysis is a problem solving technique that decomposes a system in to its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purposes. System analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Analysis and synthesis, as a scientific methods, always go hand in hand; they complement one another. Every synthesis is built up on the results of a preceding analysis, and every analysis requires a subsequent synthesis in order to verify and correct its results.

* 1. **EXISTING SYSTEM**

The Existing System is a paper based ticket system. It is not possible for the Passengers to keep track of their travel details. The records maintained in paper format is not easily manageable for the users. In case of re-imbursments, the Passengers can’t keep track of the paper tickets over a long time .

* + 1. **Drawbacks of Existing System**

1. The Existing system has no unique identification for the Individuals. That is the biggest drawback of the existing system .
2. By this missing feature the passenger will not be able to use the system to full stretch.
3. This biggest drawback of this system is overcome by our system which allots a unique ID to each Person that lets us access the details anywhere.
   1. **NEED FOR PROPOSED SYSTEM**

The Proposed System provides Unique ID for the Passengers for efficient storage and access of data.​The Data is accessible by Registered passenger throughout the Nation.​

* 1. **FEATURES OF PROPOSED SYSTEM**

Public Transport Management System is installed in the passengers Mobile Phone. This System maintains the Passengers travel details. Users are Registered and provided with a unique ID. This ID is valid throughout the Nation. The Passengers can Login to view their travel details, Book tickets for their upcoming travels and plan their trip accordingly. The Admin maintains the Database.

* + 1. **Feasibility Study**

The feasibility of the project is analyzed in this phase and business proposal is but forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. Economical Feasibility

2. Technical Feasibility

3. Social Feasibility

**1. Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of found can company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the development system as well within the budget and this was achieved because most of the technologies are freely available. Only the customized products had to be purchased.

**2. Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**3. Social Feasibility**

The aspect of study is to check the level of acceptance of the system by user. This includes the process of training the user to the system efficiently. The user must not feel the threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods are employed to educate the user about the system and to make him familiar with nit. His level of confidence must be raised but also able to make some constructive criticism, which is welcomed, as the final user of the system.

* 1. **REQUIREMENT ANALYSIS**

**2.4.1 HARDWARE REQUIREMENTS**

Processor : Pentium –I Core

RAM : 2 GB

Hard Disk : 1.28GB

* + 1. **SOFTWARE REQUIREMENTS**

Operating System : Windows 7/8/10

Front End : HTML,CSS

Front End Tool : Ionic Framework

Back End : Typescript

* + - 1. **TOOLS:**

**IONIC 3**

Ionic is a complete open-source SDK for hybrid mobile app development created by Max Lynch, Ben Sperry and Adam Bradley of Drifty Co. in 2013.The original version was released in 2013 and built on top of Angular JS and Apache Cordova. The more recent releases, known as Ionic 3 or simply "Ionic", are built on Angular. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova. Users can build their apps, and they can then be customized for Android, iOS, Windows, or modern browsers. Ionic allows you to build and deploy your apps by wrapping around the build tool Cordova with a simplified 'ionic' command line too.Ionic is focused on building for modern Web standards and for modern mobile devices. For Android, Ionic supports Android 4.1 and up. For iOS, Ionic supports iOS 7 and up. Ionic 2 supports the Universal Windows Platform for building Windows 10 apps.Ionic Framework, powered by Angular.js, supports BlackBerry[2].

**2.4.2.2 PROGRAMING LANGUAGES:**

**HTML:**

HTML(Hypertext Markup Language)is the set of markup symbols or codes inserted in a file intended for display on a World Web browser page. The markup tells the Web browser how to display a Web page’s words and images for the users. Each individual markup code is referred to as an element(but many people also refer to it as a tag).Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

HTML markup consists of several key components, including those called tags(and their attributes), character-based data types, character references and entity references.HTML tags most commonly come in pairs like <h1> and </h1> although some represent empty elements and so are unpaired ,for example <img>. The first tag in such a pair is the start tag, and the second is the end tag(they are also called opening and closing tags). Another important component is the HTML document style type declaration, which triggers standards mode rendering.

HTML is a formal Recommendation by the World Wide Web consortium(W3C) and is generally adhered to by the major browsers, Microsoft Internet Explorer and Netscape Navigator, which also provide some additional non-standard codes. The current version of HTML is HTML 4.0. However,both Internet Explorer and Netscape implement some features differently and provide non-standard extensions.

Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications[4].

**CSS:**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging Webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS was first proposed by HåkonWium Lie on October 10, 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside World Wide Web Consortium resulted in the first W3C CSS Recommendation (CSS1) being released in 1996. In particular, a proposal by Bert Bos was influential; he became co-author of CSS1, and is regarded as co-creator of CSS[4].

* + - 1. **DATABASE:**

**FIREBASE**

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

**Firebase Cloud Messaging**

Formerly known as Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) is a cross-platform solution for messages and notifications for Android, iOS, and web applications, which as of 2016 can be used at no cost.

**Firebase Storage**

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage.

**Installation & Setup**

**Connect your app to Firebase**

• Install the Firebase SDK.

• In the Firebase console, add your app to your Firebase project.

**Add the Realtime Database to your app**

Add the dependency for Realtime Database to your app-level build.gradle file:

*implementation 'com.google.firebase:firebase-database:16.0.6'[11]*

**Configure Firebase Database Rules**

The Realtime Database provides a declarative rules language that allows you to define how your data should be structured, how it should be indexed, and when your data can be read from and written to.

By default, read and write access to your database is restricted so only authenticated users can read or write data. To get started without setting up Authentication, you can configure your rules for public access. This does make your database open to anyone, even people not using your app, so be sure to restrict your database again when you set up authentication[3].

**ASSUMPTIONS AND DEPENDENCIES:**

* To use this system the user must connected to internet.
* Assumptions is made in such a way that the file server is on working/running mode.

**CHAPTER 3**

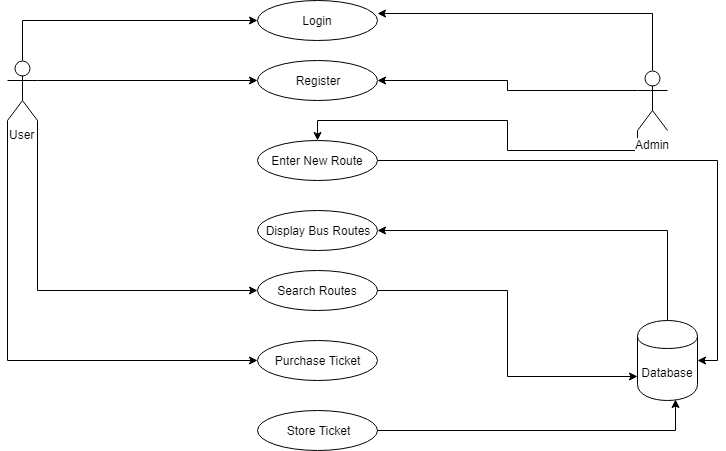
**SYSTEM DESIGN**

**3.1 UML DIAGRAMS**

The Unified Modelling Language is a standard language for specifying, visualizing, constructing, and documenting the artifacts of the software systems, as well as for business modelling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in modelling of large and complex systems.

**3.1.1 Use Case Diagram**

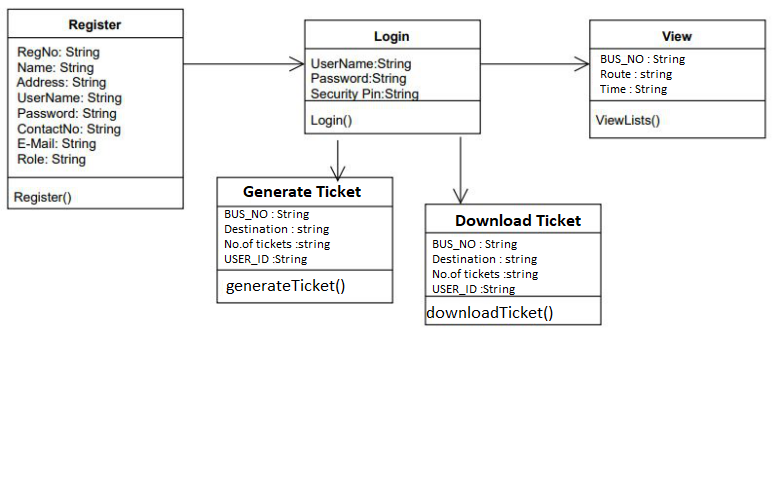
A use case is a set of scenarios that describing an interaction between a user and system. A use case diagram displays the relationship among the actors and use cases. The two main components of a use case diagram are use cases and actors. Here figure 3.1 describes the use case diagram for providing user file security which deals with actor such as admin and users.

****

**Figure 3.1 Use case diagram**

**3.1.2 Class Diagram**

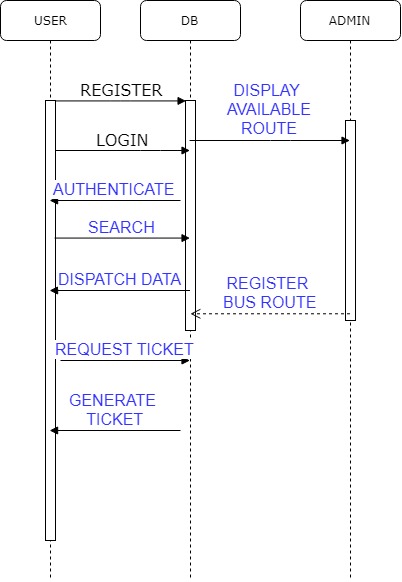
Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations and collaborations. Class diagrams are basically represent the object oriented view of the system which is static in nature. Active class is used in the class diagram to represent the concurrency of the system. This is the most widely used diagram at the time of system construction. In figure 3.2 describes the class diagram for register class, login class, upload file class, download file class and encryption class.

****

**Figure 3.2 Class diagram**

**3.1.3 Sequence Diagram**

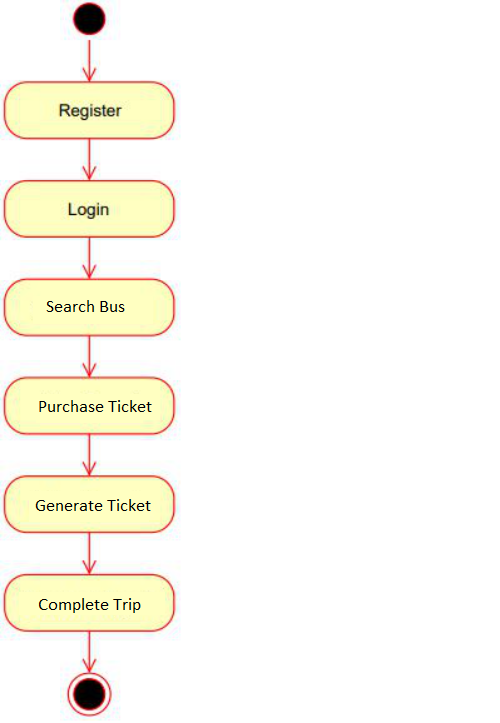
A sequence diagram is an interaction diagram. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality. In figure 3.3 represents the sequence of steps happened between user, admin, database (DB) and cloud storage.



**Figure 3.3 Sequence diagram**

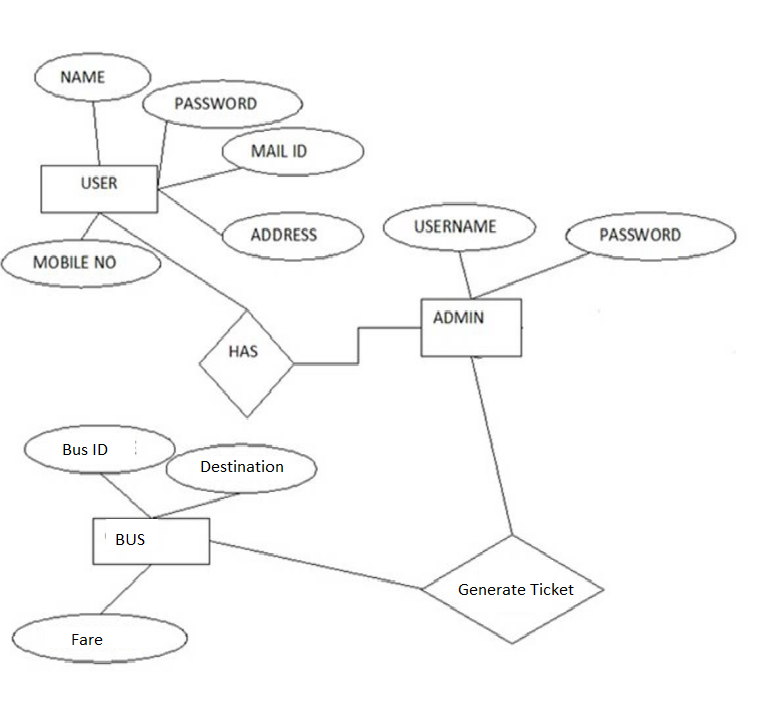
**3.1.4 Activity Diagram**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. This diagram is basically a flow chart to represent the flow from one activity to another activity. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Activity is the particular operation of the system. Figure 3.4 denotes the activities that are performed by the user for providing file security in cloud infrastructure.

****

**Figure 3.4 Activity diagram**

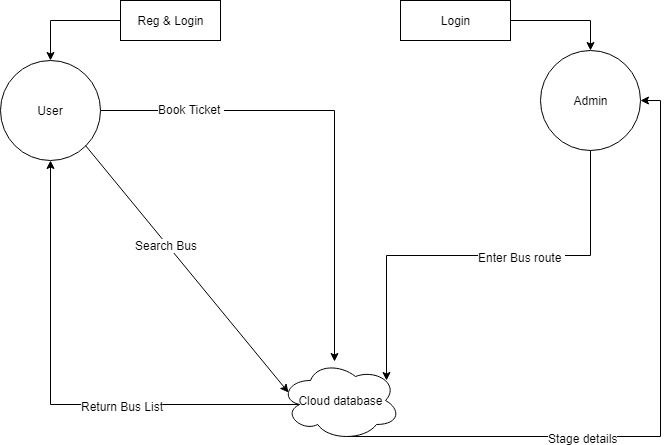
**3.1.5 ER Diagram**

An entity–relationship model (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. An ER model for the project is described in this figure 3.5 which consist of all entities with their relationships.

**Figure 3.5 ER diagram**

* + 1. **System Architecture Diagram**

This diagram represents the overall structure of the system. It also shows the behavior of the system in an abstracted way. The System architecture diagram for public transport management system which clearly describes the process of how the user and admin interacts with cloud database to purchase tickets is shown in figure 3.6. User should register for accessing data in cloud and then theycan purchase tickets. Admin logs in and uploads the bus route details into cloud database. When user purchase ticket a Ticket is generated and stored in database. To verify the ticket generated user need to display the QR generated to admin by scanning the QR code using admin app. Then the user ticket is verified and user entry is validated.



**Figure 3.6 System architecture diagram**

* 1. **DATABASE DESIGN:**

**3.2.1 USER REGISTRATION**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **FIELDS** | **DATATYPE** |
| 1 | EMAIL-ID | STRING |
| 2 | NAME | STRING |
| 3 | GENDER | STRING |
| 4 | USERNAME | STRING |
| 5 | PASSWORD | STRING |
| 6 | CONTACTNO | STRING |

**3.2.2 ROUTE REGISTRATION:**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **FIELDS** | **DATATYPE** |
| 1 | START POINT | STRING |
| 2 | END POINT | STRING |
| 3 | STOPS | ARRAY |
| 4 | FARES | ARRAY |
| 5 | TIME | ARRAY |
| 6 | BUS\_NO | STRING |

**3.2.3 FARE DATA:**

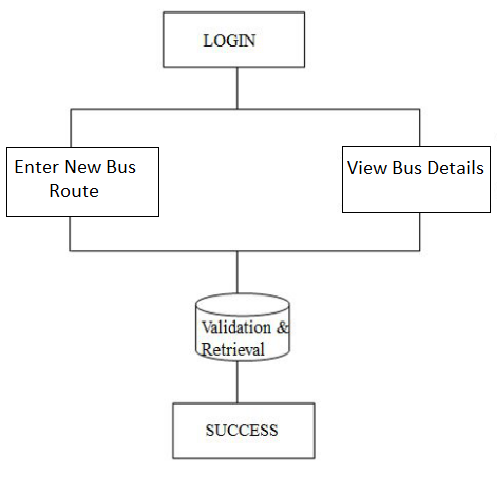
|  |  |  |
| --- | --- | --- |
| **S.NO.** | **FIELDS** | **DATATYPE** |
| 1 | BUS\_NO | STRING |
| 2 | START POINT | STRING |
| 3 | END POINT | STRING |
| 4 | TIME | STRING |
| 5 | NUMBER\_OF\_SEATS | STRING |
| 6 | FARE | STRING |

**CHAPTER 4**

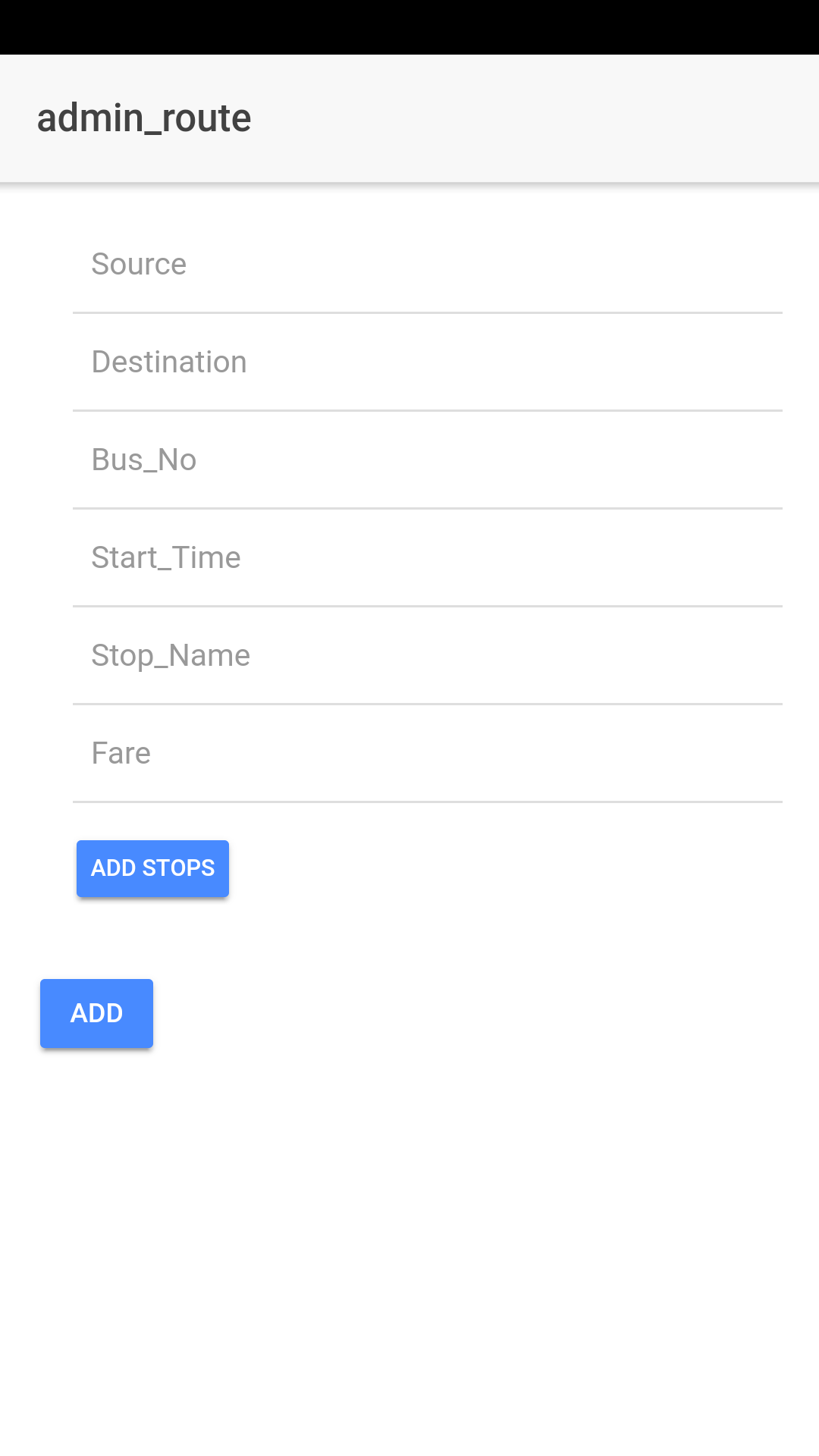
**IMPLEMENTATION AND TESTING**

**4.1 MODULE DESCRIPTION**

**4.1.1 ADMIN MODULE:**



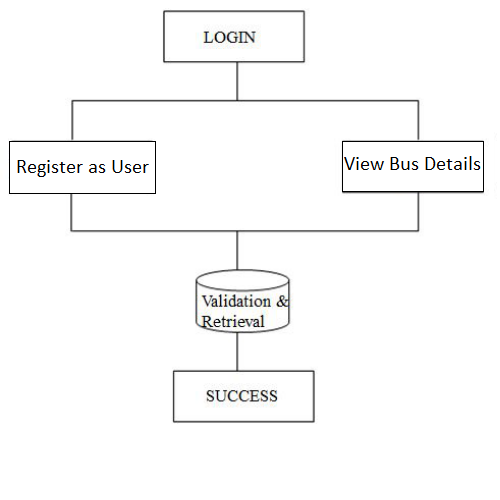
**Figure 4.1.1 Overview of Admin Module**

****

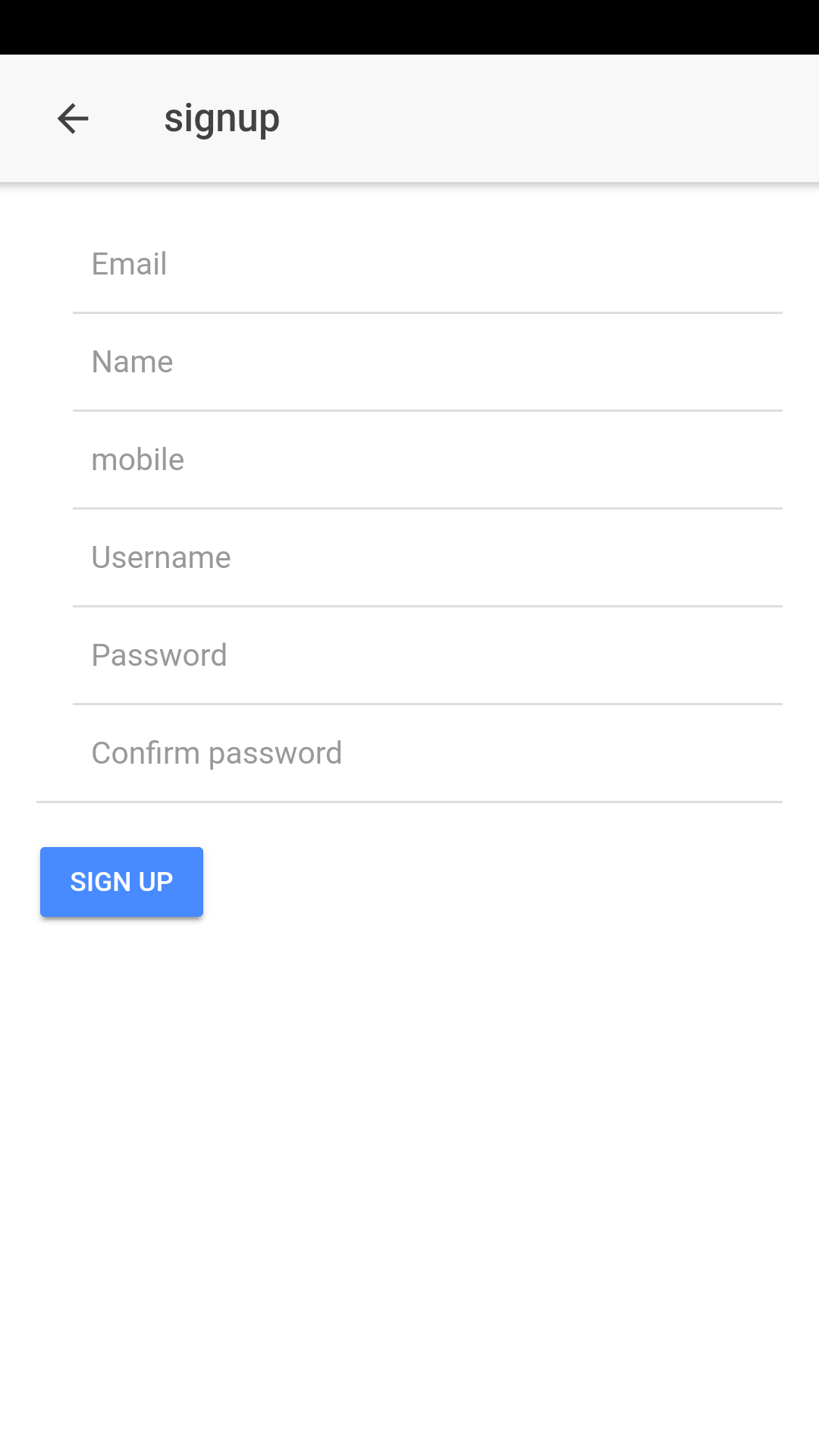
**Screenshot of Admin Module**

* Admin can be able to register the new Bus routes with their unique identification number.
* Bus routes are provided with a unique identifier by the admin.
* Admin can view the Bus details and the tickets purchased.

**4.1.2 REGISTRATION MODULE:**



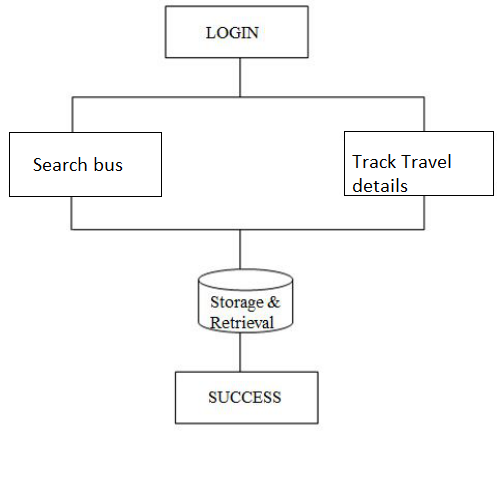
**Figure 4.1.2 Overview of Registration Module**

****

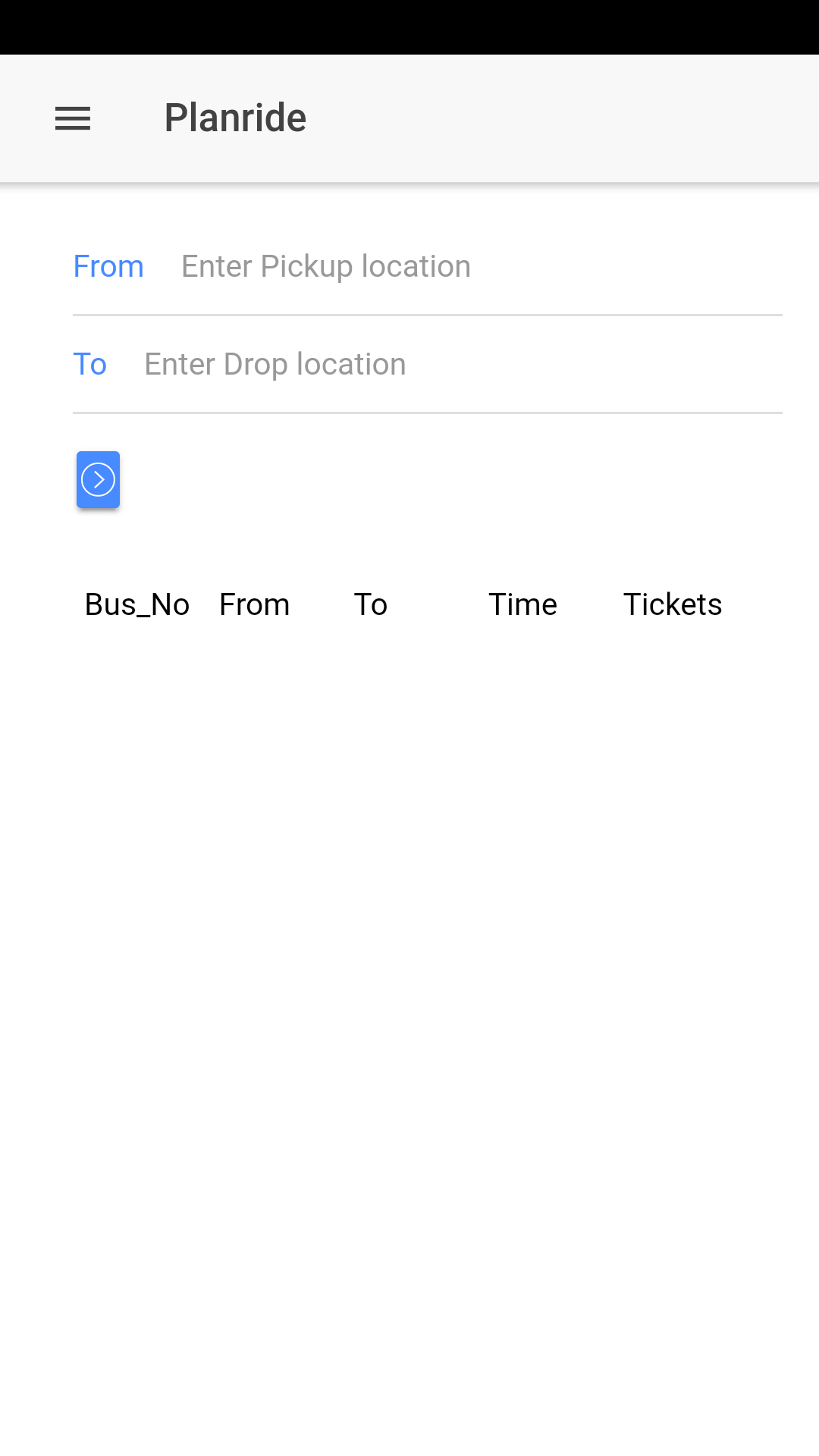
**Screenshots of signup page**

* Users have an access to registration module where they can able to register as a new user.
* On Successful registration user receives an unique username.

**4.1.3 USER MODULE:**



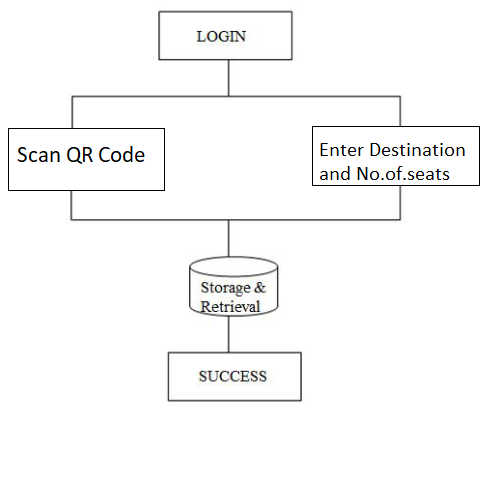
**Figure 4.1.3 Overview of User Module**

****

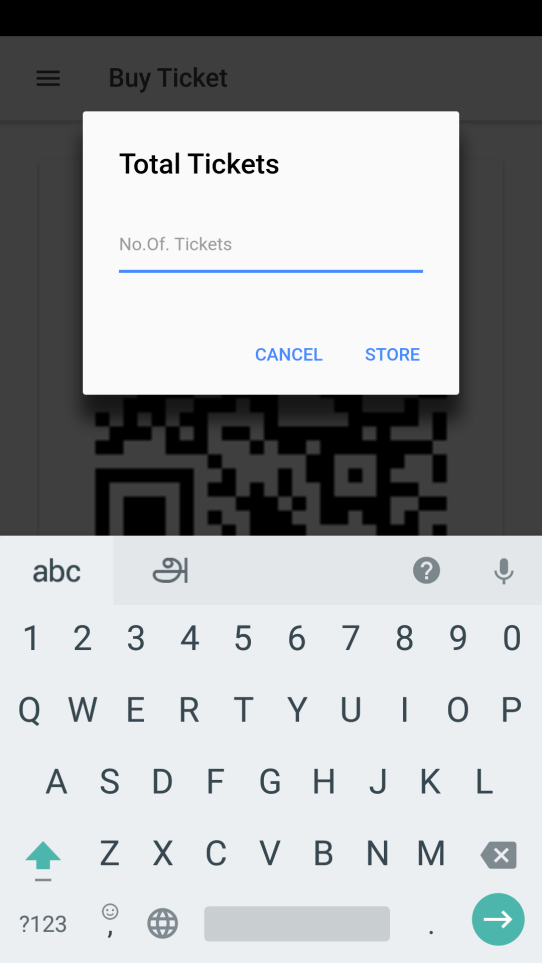
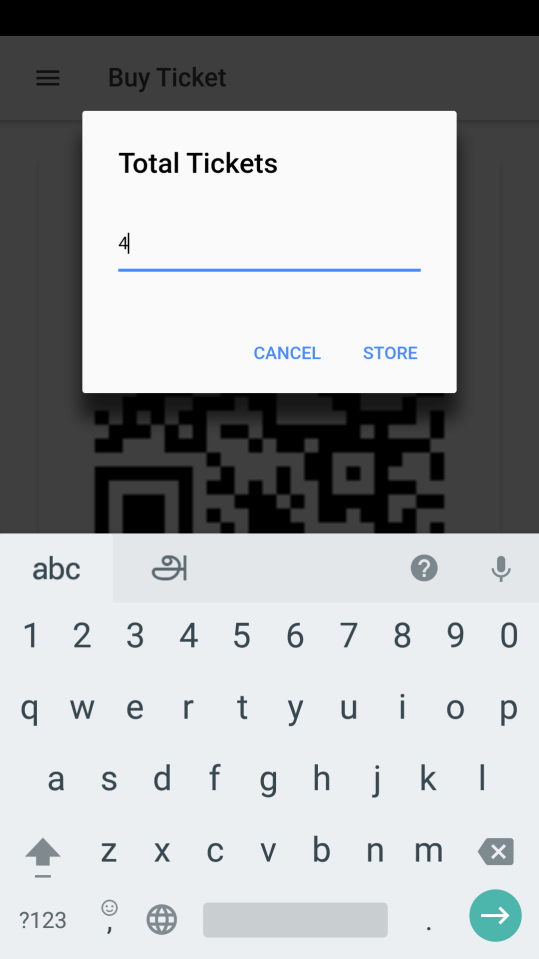
**User Home Page**

* Users can view the available buses based on their travel requirements
* User can keep track of their travel details.
* Users can plan their trips based on the availability of buses.

**4.1.4 TICKET PURCHASE MODULE:**



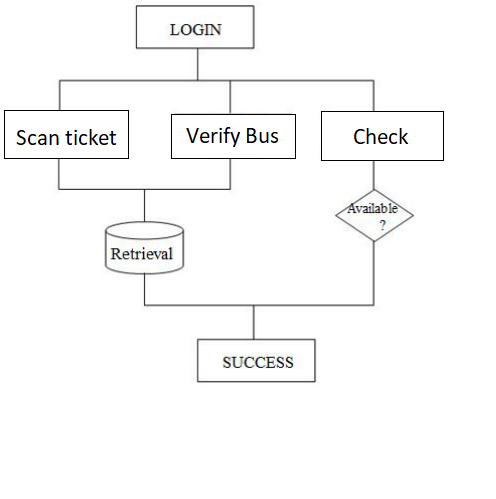
**Figure 4.1.4 Overview of Ticket Purchase Module**

****

**Ticket processing Pages**

* This module is used to purchase E-tickets.
* The module uses QR scanner to fetch the bus details and generates a
* E-tickets in the form of a QR code.

**4.1.5 TICKET VERIFICATION MODULE:**



**Figure 4.1.5 Overview of Ticket Verification Module**

* Ticket verification module is a admin access module.
* This module is used to ensure the purchase of tickets by users before they complete their journey.

**4.2 TESTING**

Since the error in the software can be injured at any stage. So, we have carry out the testing process at different levels during the development. The basic levels of testing are,

* Unit Testing
* Integration Testing
* Validation Testing
* Functional Testing
* Structural Testing

**4.2.1 Unit Testing**

Unit testing was used to test individual units in the system and ensure that they operate correctly. Alternate logic analysis and screen validations were tested in this to ensure optimum efficiency in the system. The procedures and functions used and their association with data were tested.

* + 1. **Integration Testing**

This testing process focuses on identifying the interfaces between components and their functionality. The bottom up approach was adopted during this testing. Low-level modules are integrated and combined as a cluster before testing. This allowed identifying any wrong linkages or parameters passing early in the development process as it just can be passed in the set of data and checked if the result returned is an accepted one.

* + 1. **Validation Testing**

Software testing and validation is achieved through a series of block box tests that demonstrate conformity with requirements. A test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both, the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After each validation test case has been conducted, one of the two possible conditions exists.

* + 1. **Functional Testing**

Functional testing, also known as block box or closed box testing, is normally applied to HDL (High-Level Data Link) code that operates concurrently and concentrates on checking the interaction between modules, blocks or functional boundaries. The objective here is to ensure that `correct results” are obtained when `good inputs” are applied operates in a predictable manner. Functional testing can therefore be considered as concentrating on checking that the data paths operate correctly. The coverage measurements that fall into this category are toggle, triggering, and signal trace coverage.

* + 1. **Structural Testing**

Structural testing, are known as white box or open box testing, is normally applied to sequential HDL (High-Level Data Link) code and concentrates on checking that all executable statements within each module have been exercised and the corresponding branches and paths through that module have been covered. If there is a section of HDL code that has never been exercised then there is a high possibility that it could contain an error that will remain undetected.

* 1. **DRIVING TEST CASES**

A test case is a set of conditions or variables under which a tester will determine if a requirement upon an application is partially or fully satisfied. The types of testing that are to be carried out on the system is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case no** | **Description** | **Pre-conditions** | **Pass/Fail** | **Expected results** |
| PTS\_001 | Validate user Registration | New user only allowed | Pass | Registered Successfully |
| PTS\_002 | Validate user login | Registered user only allowed | Pass | Login successfully |
| PTS \_003 | Admin upload Bus route | Registered user only allowed | Pass | Added successfully |
| PTS\_004 | User search for preferable bus route | Registered user only allowed | Pass | Fetch successfully |
| PTS\_005 | User Book Ticket | Destination and no.of seats | Pass | Ticket Booked Successfully |
| PTS\_006 | Ticket Generation | Book Ticket | Pass | Ticket Generated Successfully |
| PTS\_007 | User View Bus details | Registered user only allowed | Pass | Display the available Buses |
| PTS\_008 | Admin Verify E-Ticket | Only send Acknowledgement to valid user | Pass | Send key successfully |
| PTS\_009 | User get the response | Get the response from admin | Pass | Get the key successfully |

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

**5.1 CONCLUSION**

Conventional method of managing and generating tickets in public transport system is a more hectic to maintain and manage the functionality of the very large network of public transport system. Various issues are identified in the conventional system of management. To overcome some of the major issues like miscalculation and data maintenance of records an assessor tool is proposed.

Entity based data model is used in the form of a modern mobile application. It covers various aspects of passenger need and serves as a platform to maintain their travel history and keep track of their expenditure made on the travels. This application also serves as a complete tool to maintain the public transport system in digital form.

The application helps to maintain the public transport database effectively without any human intervention. This application helps to reduce the human effort to maintain a very large set of data being collected on a daily basis.

**5.2 FUTURE WORK**

The application developed is simple prototype to explain the basic functionalities of the upcoming application. In the upcoming release following features will be added

* **Tracking the status of Transits[8]**
* **Remainder based on user profile[1]**
* **Provides a suggestion list for the users based on travel history[1]**
* **Integration of all public transports**
* **Includes the Air transport and Railways**

**REFERENCES**

1. Bus Information System: A smart partner for commuters. DOI: [10.1109/ICCIT.2010.5711021](http://dx.doi.org/10.1109/ICCIT.2010.5711021)
2. <https://www.ionicframework.com/docs>
3. https://www.firebase.com/docs
4. Paul RN,https://ieeexplore.ieee.org/abstract/document/6953110/
5. <https://www.researchgate.net/publication/286586037_Smart_public_transport_system>
6. Mobile app : Boston Transit: MBTA Bus, Subway & Rail Transport Tracker by Anil Vasani
7. Easy.GO - For bus, train & Co. The free app for buses, trains and trams in the Mitteldeutschen Verkehrsverbund (MDV), the city of Bonn (SWB)
8. Gaurav Chheda, Niket Gajra, Manal Chhaya, Jitesh Deshpande, Saylee Gharge, ”REAL TIME BUS MONITORING AND PASSENGER INFORMATION SYSTEM”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-6, January 2012.
9. Juan Zambada, Ricardo Quintero, Ramon Isijara, Ricardo Galeana, Luis Santillan, “An IoT based scholar bus monitoring system”, Smart Cities Conference (ISC2), 2015 IEEE First International, IEEE Xplore: 28 December 2015, DOI: 10.1109/ISC2.2015.7366202 .
10. Lin Kang, Stefan Poslad, Weidong Wang, Xiuhua Li, Yinghai Zhang, Chaowei Wang, “A Public Transport Bus as a Flexible Mobile Smart Environment Sensing Platform for IoT”, Intelligent Environment (IE), 2016 12th International conference, IEEE Xplore : 27 October 2010, DOI: 10.1109/IE.2016.10 [5] Nomusa Dlodlo, “The internet of things in transport management in South Africa”, Emerging Trends in Networks and Computer Communications (ETNCC), 2015 International Conference, IEEE Xplore: 13 August 2015, DOI: 10.1109/ETNCC.2015.7184802.
11. R. Meier, ―Professional Android 4 Application Development,‖ John Wiley & Sons, 2012.

**URL**

1. Source :https://www.w3school.com
2. Source :https://www.ionicframework.com/docs
3. Source :https://www.firebase.com/docs