

# **CHAPTER 1: INTRODUCTION**

With the advance in time and technology there is a need for faster dissemination of information. The increasing advantages of automated system now are at highest position thus many manual processes are automated. Since the automated system is demanded now-a-days, educational infrastructures like colleges needed their manual system to function on mobile computing systems. Changes in Information Technology (IT) allow institutes to utilize databases and applications such as Student Information System thus, making the accessing of records centralized.

## **1.1 DESCRIPTION OF PROJECT**

The design and implementation of the system is to provide service in institute and colleges. The system is to provide comprehensive student information system and user interface is to replace the current paper records. College Staff uploads attendance, results and college notifications through cloud database using android application or directly in the cloud. All data is stored securely on Cloud database managed by the college Administrator. The system decreases paperwork and time needed to access student records. Previously, college relied heavily on paper records for this initiative which had its own disadvantages. This system provides a simple interface for the maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information are scattered, can be redundant and collecting relevant information may be very time consuming. Our proposed system ensures to overcome these limitations.

The project focuses on presenting information in an easy and intelligible manner which provides facilities like attendance monitoring, circular notifications, result viewing thus reducing paper work and automating the record generation process in an educational institution.

## **1.2 EXISTING SYSTEM**

The system which is used nowadays has some drawbacks which need to be improved for better performance. The system through which the feedback is taken is not good enough. The views of each and every student are not expressed through these systems. As the technology is developed day by day we need to use this technology so we can get an efficient result in adequate time. For attendance management in the present system all work is done on paper. The whole session attendance is stored in register and at the end of the session the reports are generated. We are not interested in generating report in the middle of the session or as per the requirement because it takes more time in calculation. In the present system the result is viewed on the notice board. It requires lot of paperwork and is time consuming. College cannot even provide urgent notifications to students in case of emergency. Every student and faculty who are using bus services face a problem that where the bus is coming at that particular time.

## **1.3 DRAWBACKS OF AN EXISTING SYSTEM**

- The existing system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently. The use of the some technology can be complicated and time consuming. These system need to handled by specialist for maintaining and update the system which can again be very costly.
- There is a possibility of human errors that costs huge misconception while entering internal marks or student attendance.
- No backups are maintained by the college management, if any damage occurs to the physical records or any system errors occur, the data lost cannot be recoverable.

## **1.4 PROPOSED SYSTEM**

Currently many android applications are available separately to favor the students with respect to their studies such as to assist them in taking lecture notes or videos, download the notes, calculator application which helps in doing complex engineering functions. There are many different applications which are dedicated to that particular college for result distribution, circulars and library management.

On the other hand there is no integrated application for student to have everything in one application. To overcome the above limitations this project develops an android application for students. One app to check available book details in Library, Internal Marks, Circulars, Time Tables, Attendance Details, Live Bus tracking, Hostel Details etc.

## **1.5 FEATURES OF PROPOSED SYSTEM**

- Circular Information**

Now describe a network for distributing campus information among lecturers and students. The concept of developing college information system is to ensure that student can access information at any time through smart phone. So this application updates the circular information such as holidays, news events, and last date for the payment of exam fees, enrollment of competitions and various academic notifications.

- Time Tables**

One of the major problem faced by the students are knowing and memorizing of time tables. Not everyone can remember the whole time table. So this application provides every department, every year and every section time tables. So every student can access the time tables anywhere and anytime.

- Internal marks**

Students can access their internal marks anytime. Students cannot remember their internal marks in every subject. Using this application they can view their internal marks whenever they want.

- Attendance**

When the attendance circular is passed to the students that information may not be reached to all students, some may absent on that day. They can view their attendance details whenever they need. It will help them to maintain their attendance.

- **Library data**

No one knows the details of the particular book available in the library or not. This project provides the count of available books in the library. User can search the book using the title.

- **Live Bus Tracking**

College busses information can also be known using this application. Location of the bus at that particular instance of the can be known using this application.

- **Hostel information**

If any student wants to join in the college hostel, one cannot know the details of vacancies in the hostel. Using this application the student can know how many vacancies are available in the hostel. The student can also register for bed using a register form online.

## **CHAPTER 2: ANALYSIS**

### **2.1 INTRODUCTION**

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

### **2.2 ANALYSIS MODEL**

The model that is basically being followed is the SPIRAL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study's done. Once that part is over the requirement analysis and project planning begins. If the system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are

- Requirement Analysis
- Project Planning
- System Design
- Coding
- Unit Testing

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model

with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

### **2.2.1 Spiral Model - Design:**

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

### **2.2.2 Identification:**

This phase starts with gathering the requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

The identified requirements in this project are digitalizing the circulars, internal marks, Attendance, Time Tables. Also getting the live location of the busses.

### **2.2.3 Design:**

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, and the final design in the subsequent spirals.

The design phase involves designing of data flow diagrams, UML diagrams and the system architecture.

### **2.2.4 Construct or Build:**

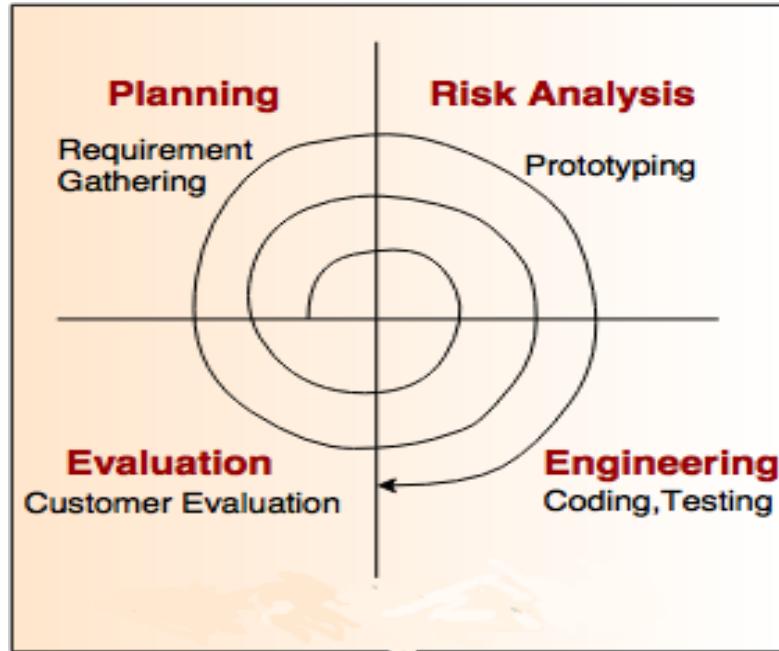
The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

The customer requirement may change every time like adding additional components like providing lecture notes via this project, online fee payment etc., these type of requirements will be added later spirals.

### **2.2.5 Evaluation and Risk Analysis:**

Risk Analysis includes identifying, estimating and monitoring the users limit in Firebase, invalid data entry to the Firebase etc. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

The following illustration is a representation of the Spiral Model, listing the activities in each phase.



**Fig.2.1: Spiral model**

Based on the customer evaluation, the software development process enters the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

## 2.3 SYSTEM REQUIREMENT SPECIFICATIONS

### 2.3.1 Introduction

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

### **2.3.2 Functional Requirements**

Following is a list of functionalities of the browsing enabled system.

- An Activity with a UI that allows you to interact with the modules provided in the application. Provide a second Activity that allows users to access the circulars in which they can view every circular provided through that application.
- This application allow a user to access time tables of particular branch, year and section at anytime, anywhere.
- The android studio provide a number of useful layout components, views, and tools that you may want to use to create modules.

### **2.3.3 Non-Functional Requirements**

The system should be support Android Studio. All the data provided to the student are stored in the cloud database called Firebase by Google. To store that data it needs a Gmail account and that credentials should be kept private.

These are the mainly following:

- Access of college information like marks, time tables, library management etc. 24 X 7 availability
- Better component design to get better performance at peak time.
- Flexible service based architecture will be highly desirable for future extension.

The existing system is completely dependent on the user to perform all the duties.

### **2.3.4 System Requirements**

- Operating System windows 10 / 7
- RAM 8Gb / 4Gb
- Processor intel core i3 or higher
- Hard-Disk At least 240Gb

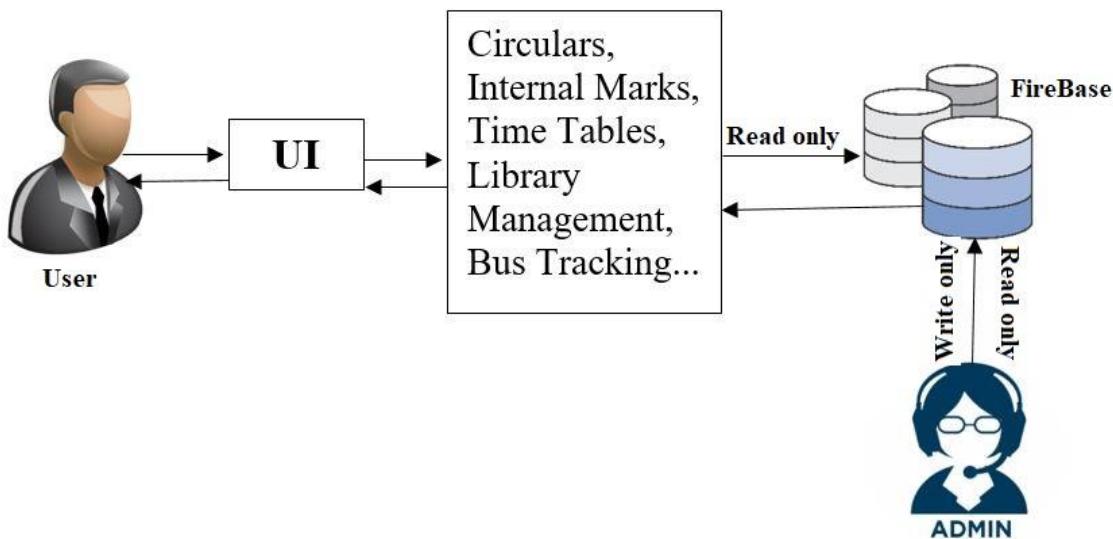
### **2.3.5 Software used**

- Android Studio

### 2.3.6 Database Used

- Firebase (Cloud database by Google).

### 2.4 CONTEXT DIAGRAM OF THE PROJECT



**Fig.2.2: System Architecture**

The proposed system of selective aggregation by the system analyst

The system has the following components

➤ **User:**

User is the person who enters into the application to access the features provided like accessing Circulars, Time Tables, Internal Marks, Bus Tracking, Attendance, Library Data.

➤ **Admin:**

Admin got the total control over the data he can modify anything he wants to , He can add the new data periodically provided by the faculty , at the same time he can also able to delete the data.

➤ **GUI:**

The user interface (UI) is everything designed into an information device with which a person may interact. This can include display screens, keyboards, touch screens and the appearance of a mobile screen. It is also the way through which a user interacts with an application

➤ **DATABASE:**

A database is a collection of information that is organized so that it can be easily accessed, managed and updated. In this application the database used is Firebase provided by Google.

# **CHAPTER 3: DESIGN**

## **3.1 INTRODUCION**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

## **3.2 DATA FLOW DIAGRAMS:**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations.

A full description of a system actually consists of a set of data flow diagrams.

Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow

diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels.

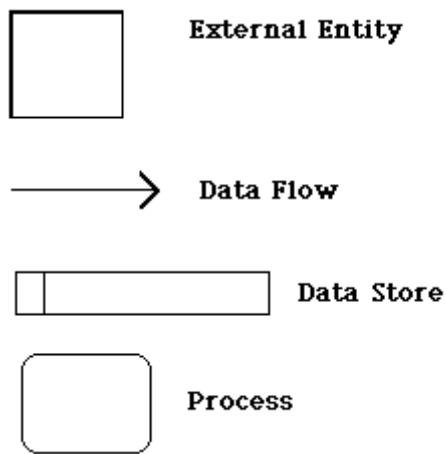
Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consist a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

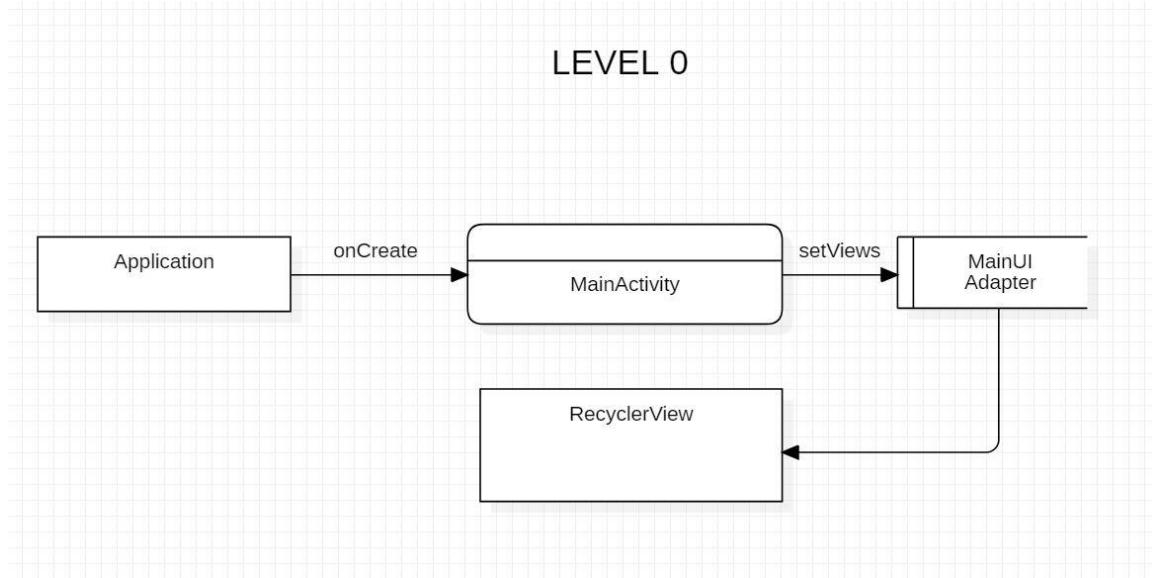
**DFD SYMBOLS:** In the DFD, there are four symbols

A square defines a source(originator) or destination of system data

- An arrow identifies data flow. It is the pipeline through which the information flows
- A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
- An open rectangle is a data store, data at rest or a temporary repository of data.

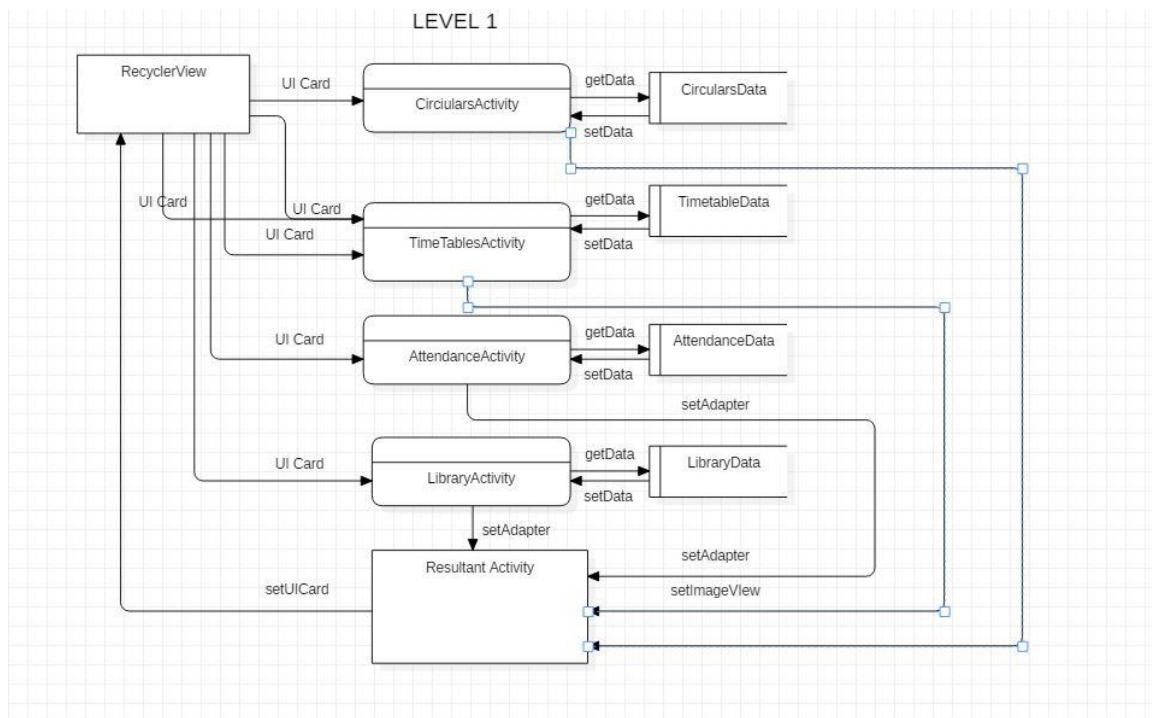


## Level 0:



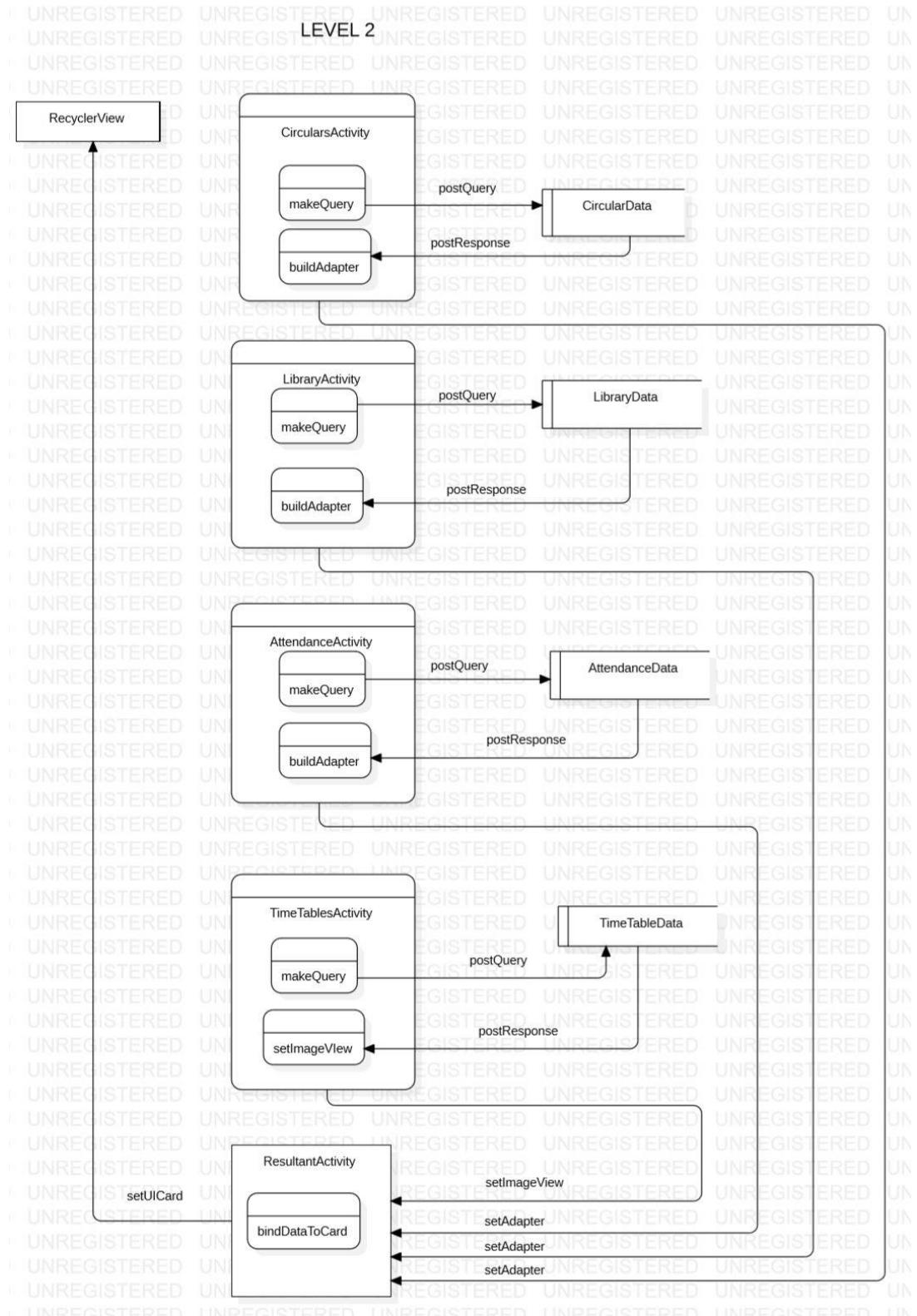
**Fig.3.1: DFD Level 0**

## Level 1:



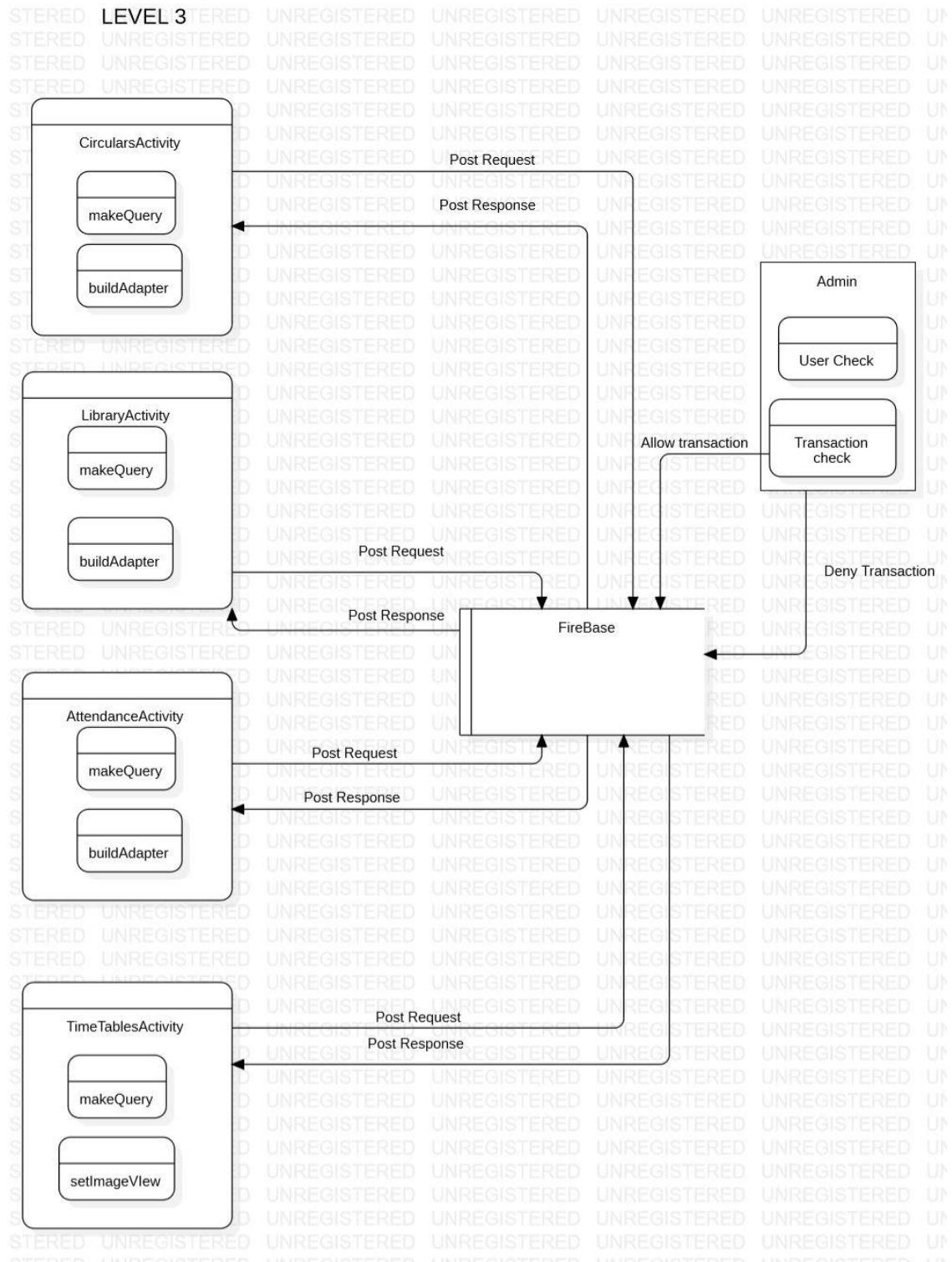
**Fig.3.2: DFD Level 1.**

## Level 2:



**Fig.3.3: DFD Level 2**

### Level 3:



**Fig.3.4:DFD Level 3**

### **3.3 UML DIAGRAMS:**

UML is the language used in the information technology industries versions of blue print. It is a method for describing the system architecture in details using this blue print. It becomes much easier to build or maintain a system and to ensure that the system we hold up to requirement changes.

A model is a simplification of reality. A model provides the blueprints of a system. A model may be structural, emphasizing the organization of the system, or it may be behavioral, emphasizing the dynamics of the system.

#### **3.3.1 Building Blocks of UML**

The three kinds of building blocks in UML are

- A. Things
- B. Relationships
- C. Diagrams

##### **A. Things:**

Things are the data abstractions that are first class citizens in a model. Things are of 4 types:

- Structural Things
- Behavioral Things
- Grouping Things
- Annotational Things.

##### **B. Relationships:**

Relationships tie the things together. Relationships in the UML are

- Dependency
- Association
- Generalization
- Specialization.

- **Dependency:** Dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing.
- **Generalization:** A generalization is a specialization / generalization relationship in which objects of the specialized element (child) are substitutable for objects of the generalized element (parent).
- **Association:** An association is a structural relationship that describes a set of links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts.

### **C. Diagrams:**

A diagram is graphical representation of a set of elements; most often rendered as a connected graph of vertices a thing. Diagrams in the UML are of 2 types

- Static diagrams
- Dynamic diagrams

#### **Static diagrams are**

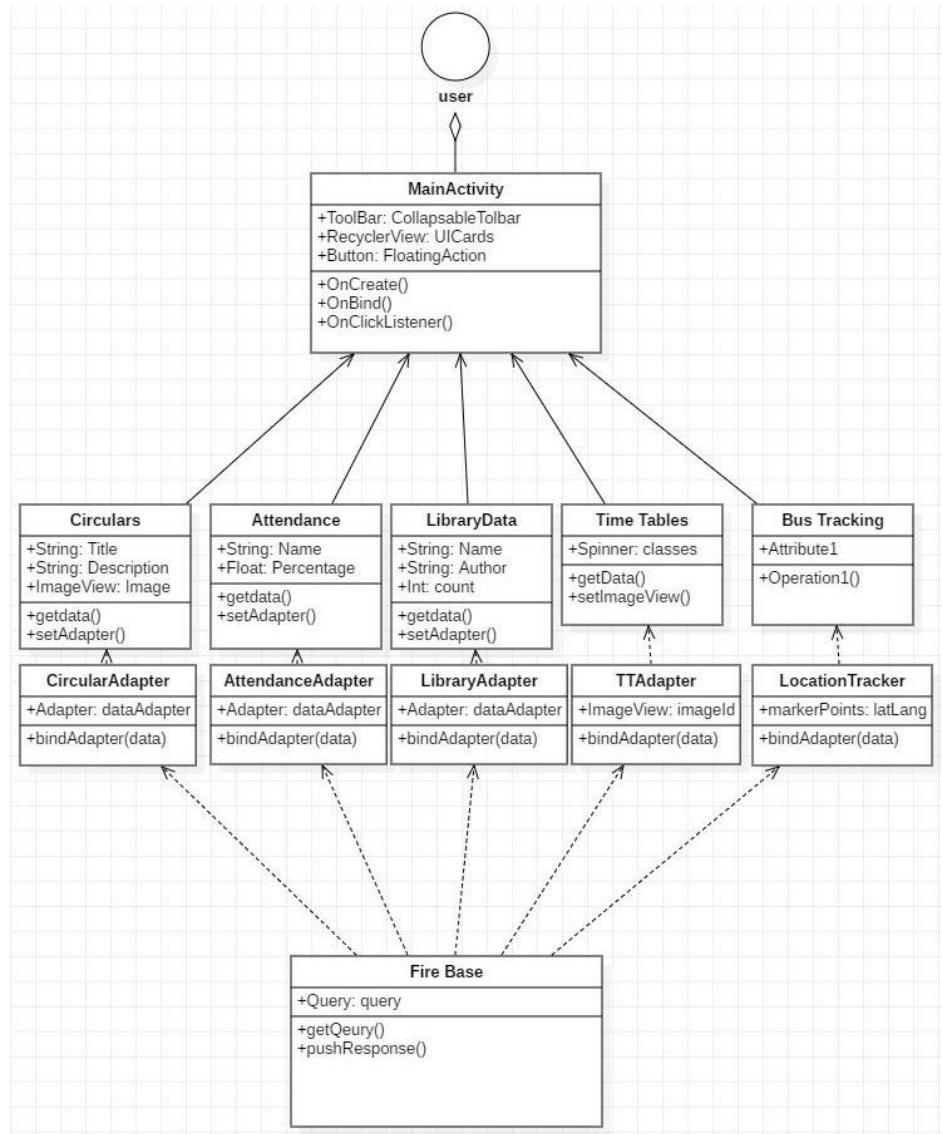
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

#### **Dynamic diagrams are**

- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram

### 3.3.2 Class Diagram for User:

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.



**Fig.3.5: User Class diagram for College Management System.**

The working of class diagram includes both the attributes of all class required in this project along with the operations with their required relationships and some multiplicities.

### **Attributes and operations:**

MainActivity:

The attributes of MainActivity are ToolBar, RecyclerView and Button.

The operations carried out in MainActivity class includes:

- OnCreate( ) - initializes the activity
- OnBind( ) - returns an IBinder object.
- OnClickListner () - click event handler for a button.

### **Circulars:**

The attributes of Circulars are String and imageView.

The operations carried out in Circulars include:

- getdata( ) - used to extract the data.
- setAdapter( ) - in order to display items in the list.

### **Attendance:**

The attributes of Attendance are String and Float.

The operations carried out in Attendance include:

- getdata( ) - used to extract the data.
- setAdapter( ) - in order to display items in the list.

### **LibraryData:**

The attributes of LibraryData are String and int.

The operations carried out in LibraryData include:

- getdata( ) - used to extract the data.
- setAdapter( ) - in order to display items in the list.

### **Time tables:**

The attributes of Time Tables are Spinner.

The operations carried out in Time Tables are:

getdata( ) - used to extract the data.

setImageview( ) -to display image on the screen.

**Firebase:**

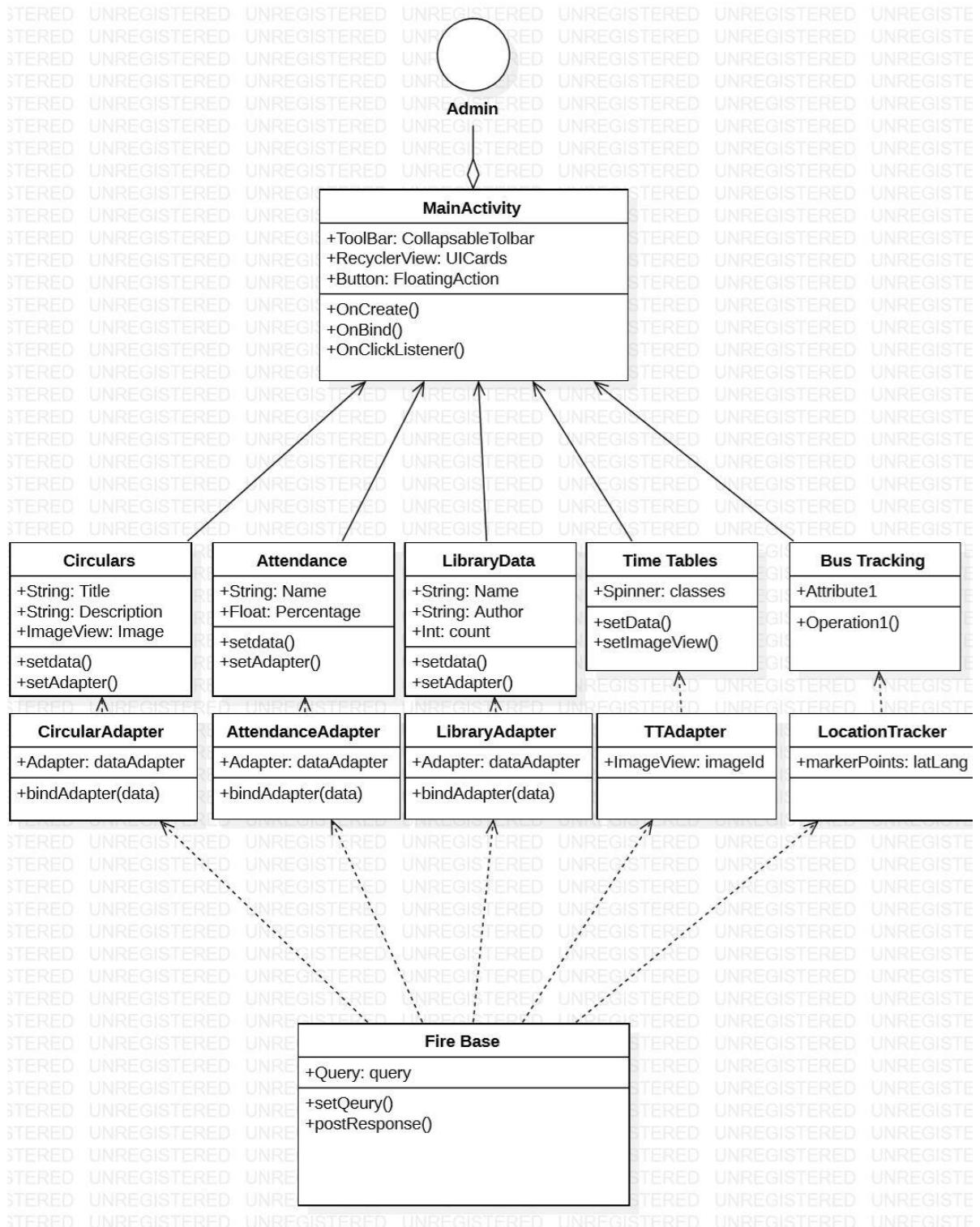
The attributes of Firebase are Query.

The operations carried out in Firebase are:

getQuery( ) - used for reading data.

pushResponse( ) - sends response to the request.

### 3.3.3 Class Diagram for Admin:



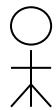
**Fig.3.6 Admin Class Diagram for Smart College Management**

### **3.3.4 Use Case Diagram:**

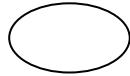
Use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagram address the static use case view of a system.

These diagrams are especially important in organizing and modeling the behavior of a system. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So we can say use cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

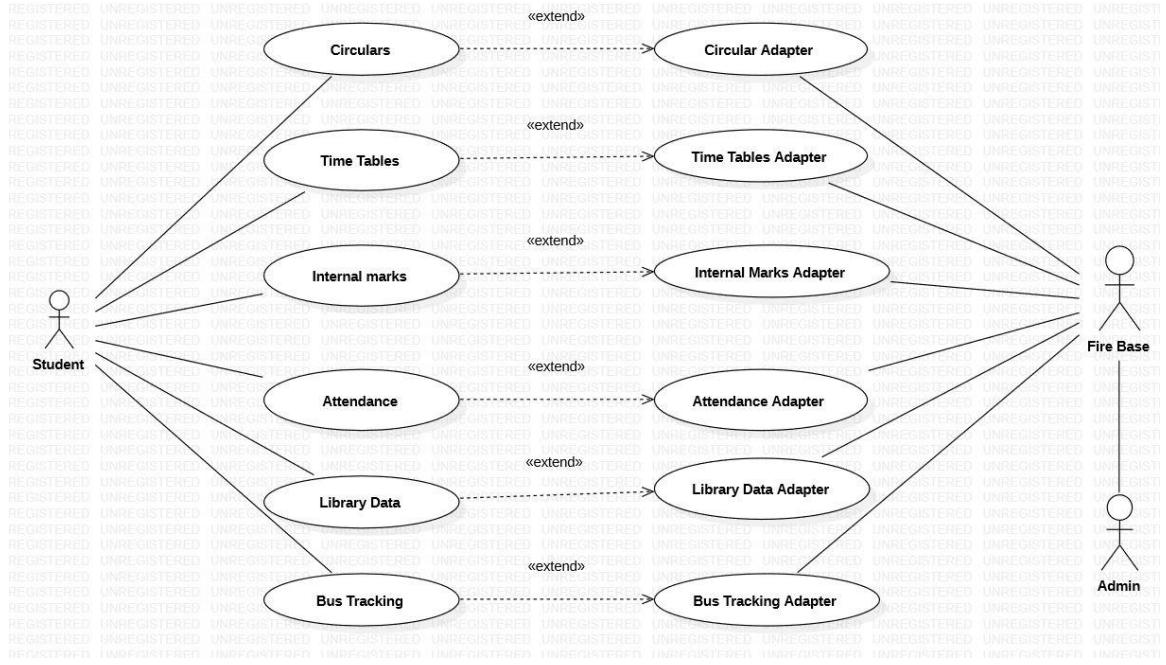
- **Use Cases :** A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- **Actors:** An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.



Actor



Action



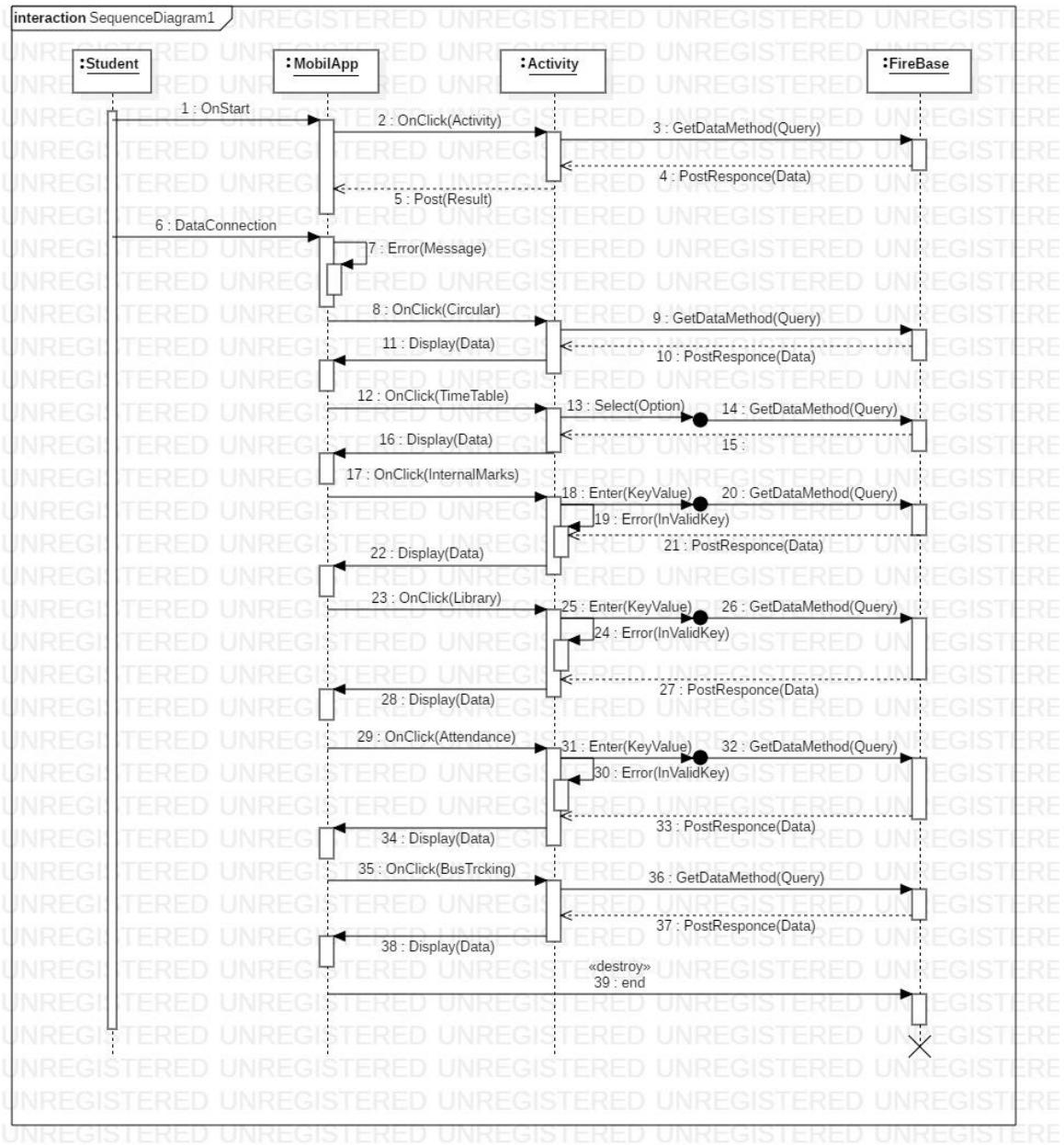
**Fig.3.7: use case diagram for smart college management system**

The actors included in this are student. The use cases required for designing are :

- Circulars for getting information about circulars.
- Time Tables for knowing the Timetables.
- Internal Marks for knowing marks anytime.
- Attendance for getting their attendance every month.
- Library Data for knowing availability of books.
- Bus Tracking for live status of the bus.

### 3.3.5 Sequence Diagram:

The sequence diagram is an interaction diagram that emphasizes the time ordering of messages for modeling a real time system. Graphically, a sequence diagram is a table that shows objects arranged along the X axis and messages, ordered in increasing time, along the Y axis. Sequence Diagram consists of objects, links, lifeline, focus of control, and messages.



**Fig.3.8: sequence diagram for smart college management system.**

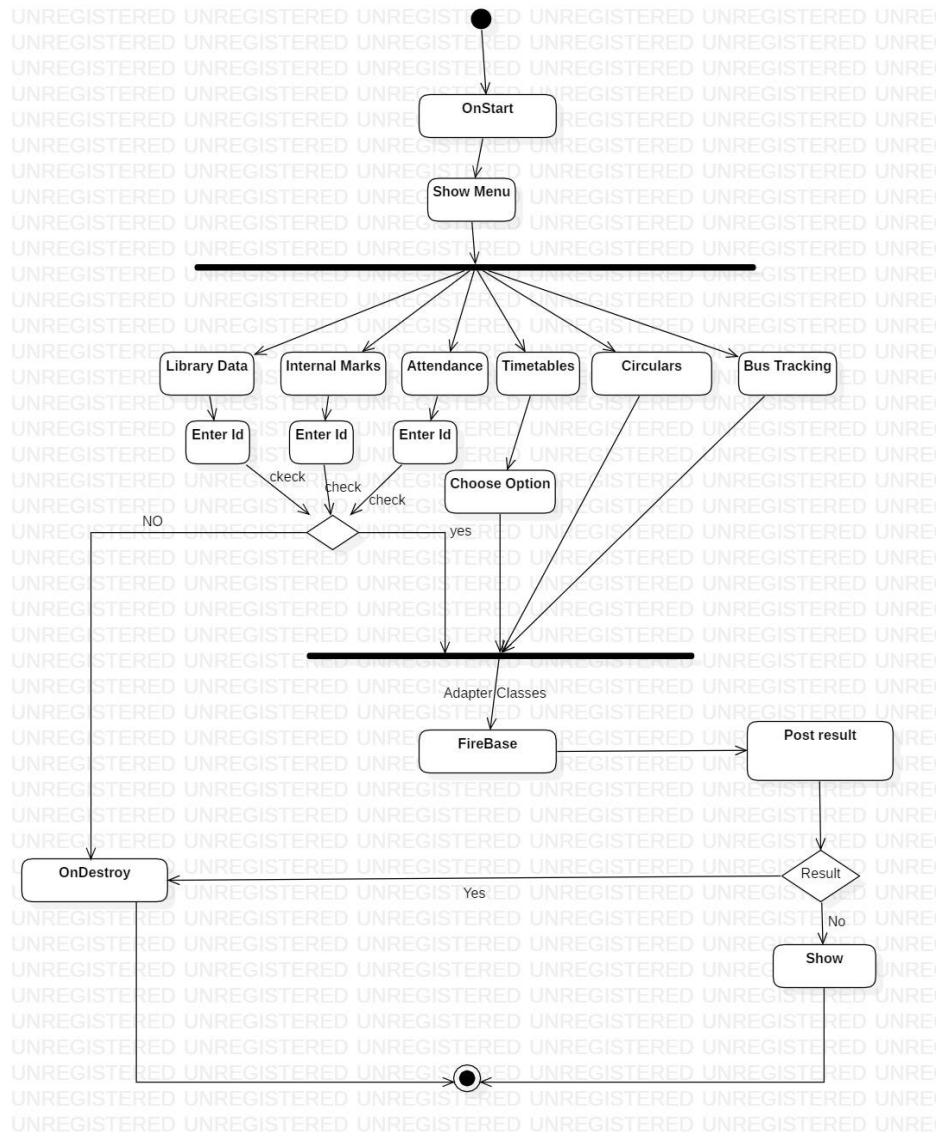
There are four objects in the sequence diagram. They are Student, Mobile app, Activity, Firebase.

- Student have access to all the modules provide in the application.
- Mobile app is an interface between user and the data.
- Firebase is a cloud database which allows admin to store the data and can be retrieved when the user requests for data.

### 3.3.6 Activity Diagram for User:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

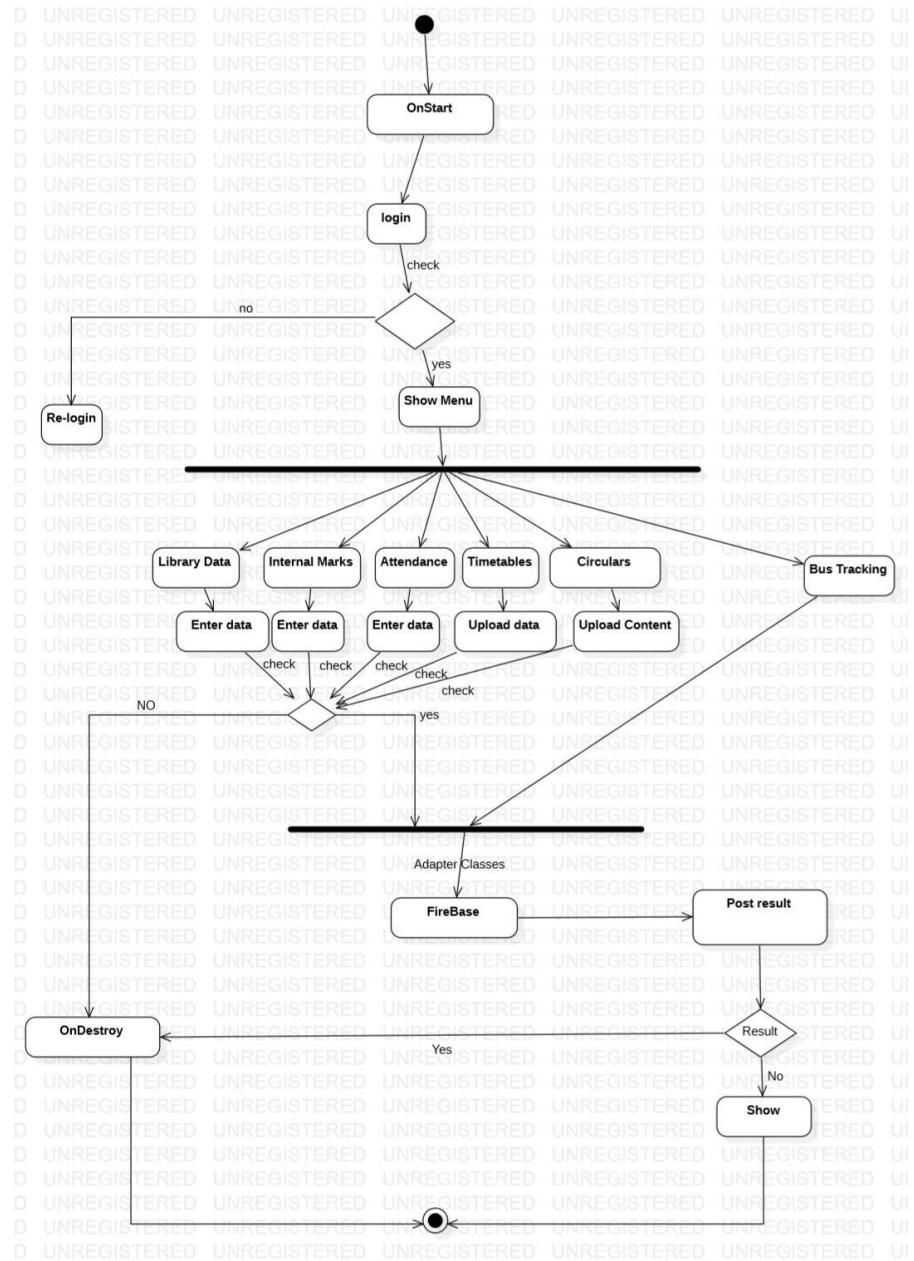


**Fig.3.9: User Activity diagram for Smart College Management System**

OnStart( ) calls whenever the activity starts, it displays the menu which has many UIcards which represents different activities.

On clicking those activities a new event starts in the new window. Whenever the user sends request to the database, it processes the request and gives response the user.

### 3.3.7 Activity Diagram for Admin:



**Fig.3.10 Admin Activity Diagram for Smart College Management System**

### 3.3.8 Object Diagram:

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

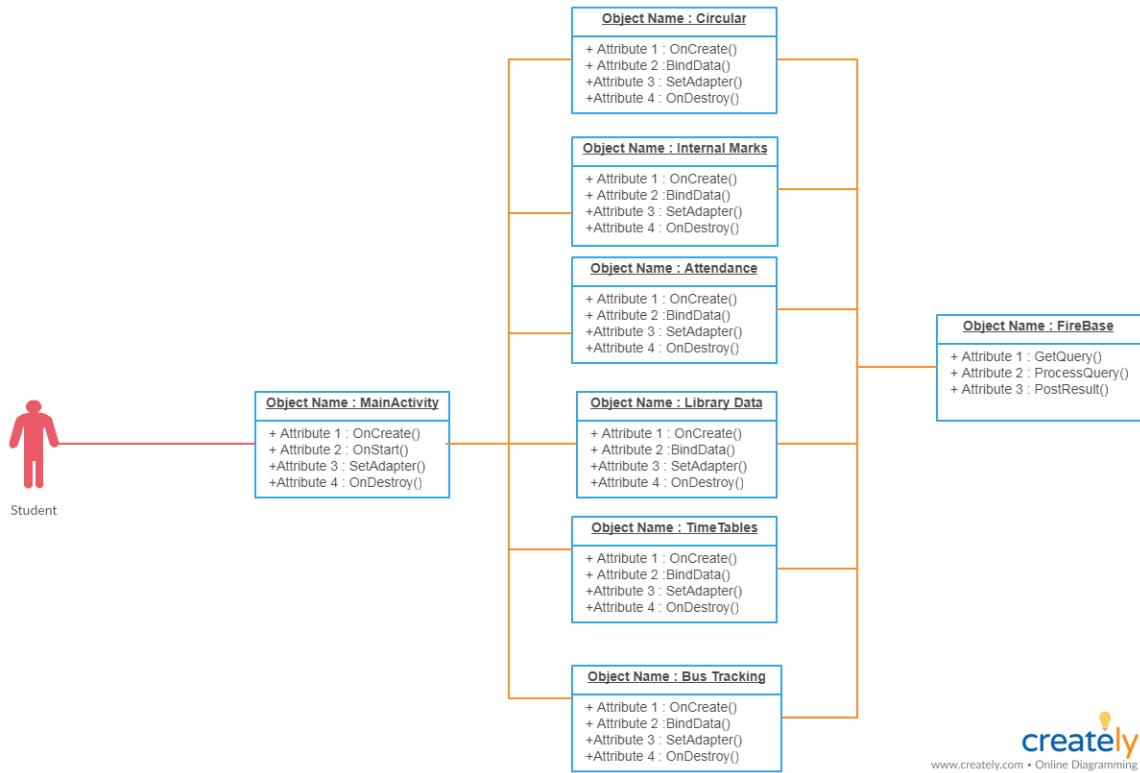
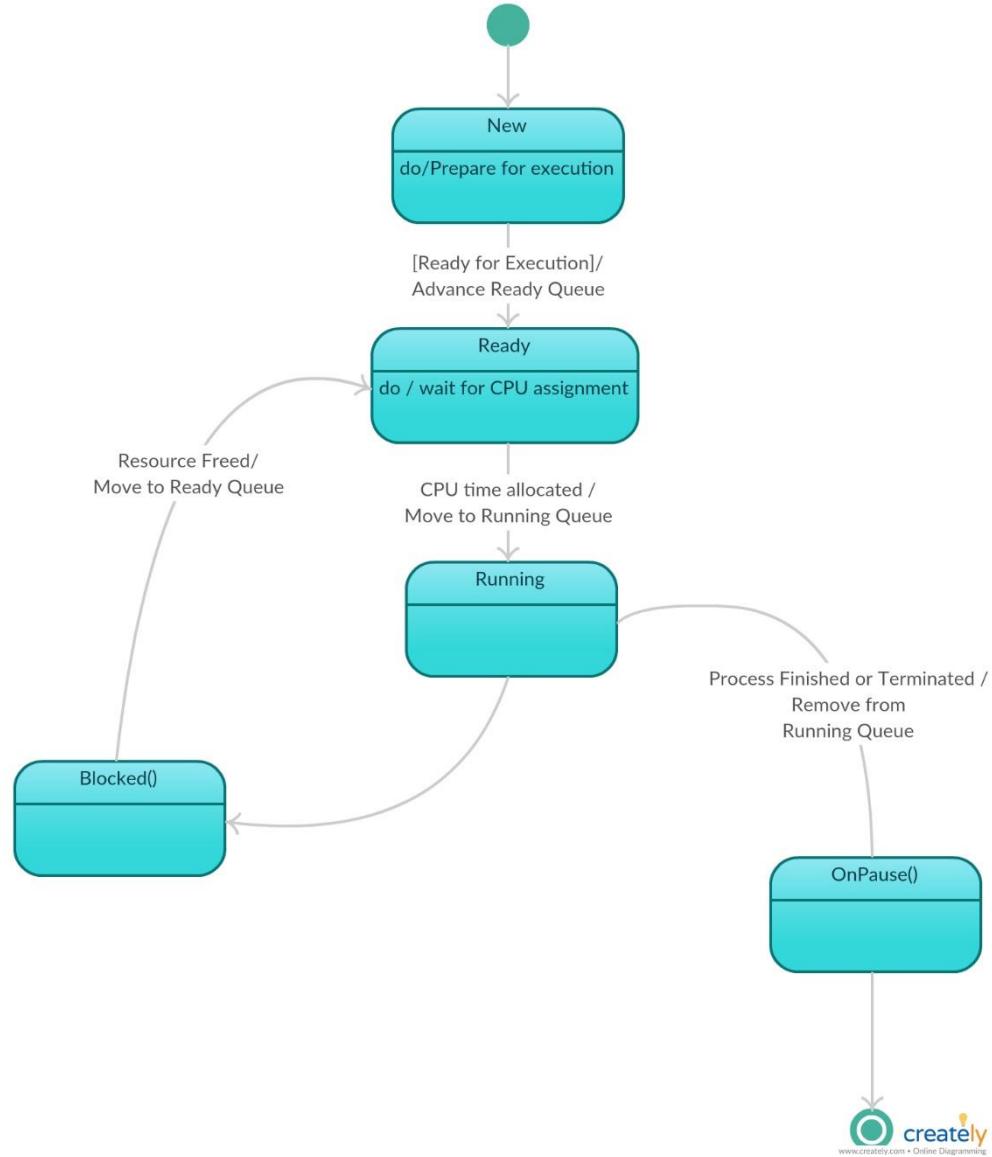


Fig.3.11: Object diagram for smart college management system

### 3.3.9 State Chart Diagram:

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

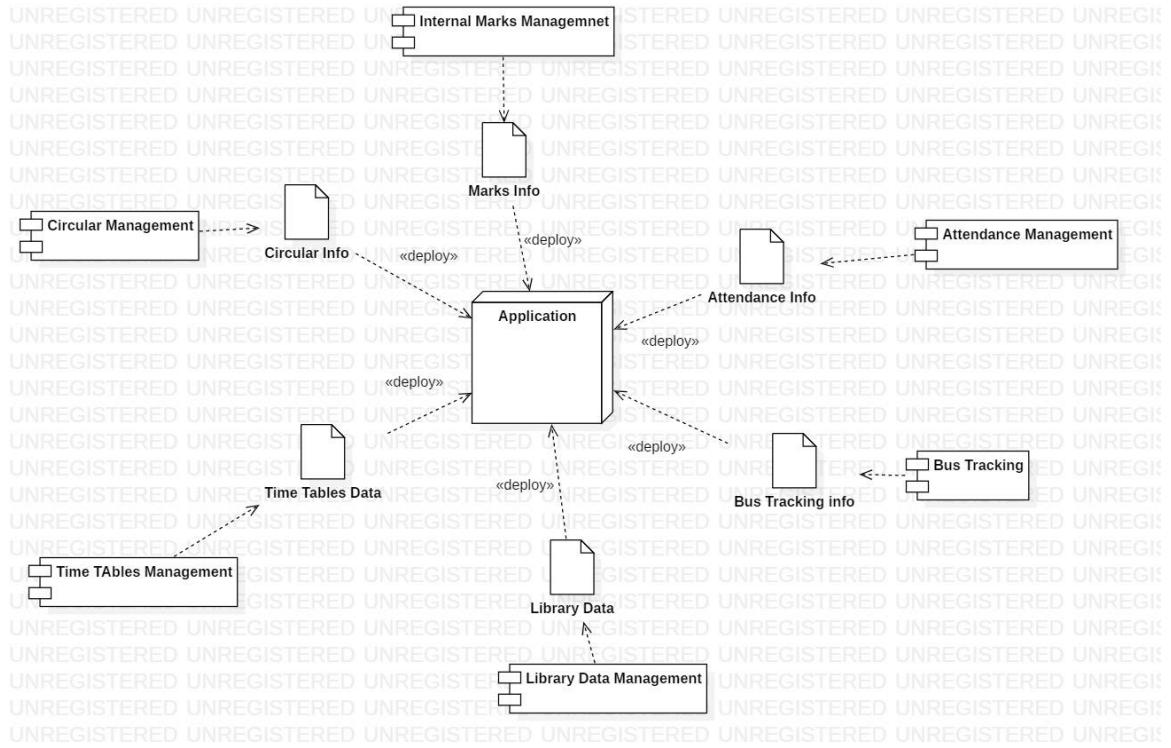


**Fig.3.12 State chart diagram for Smart College Management System**

### 3.3.10 Component Diagram :

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse

engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.



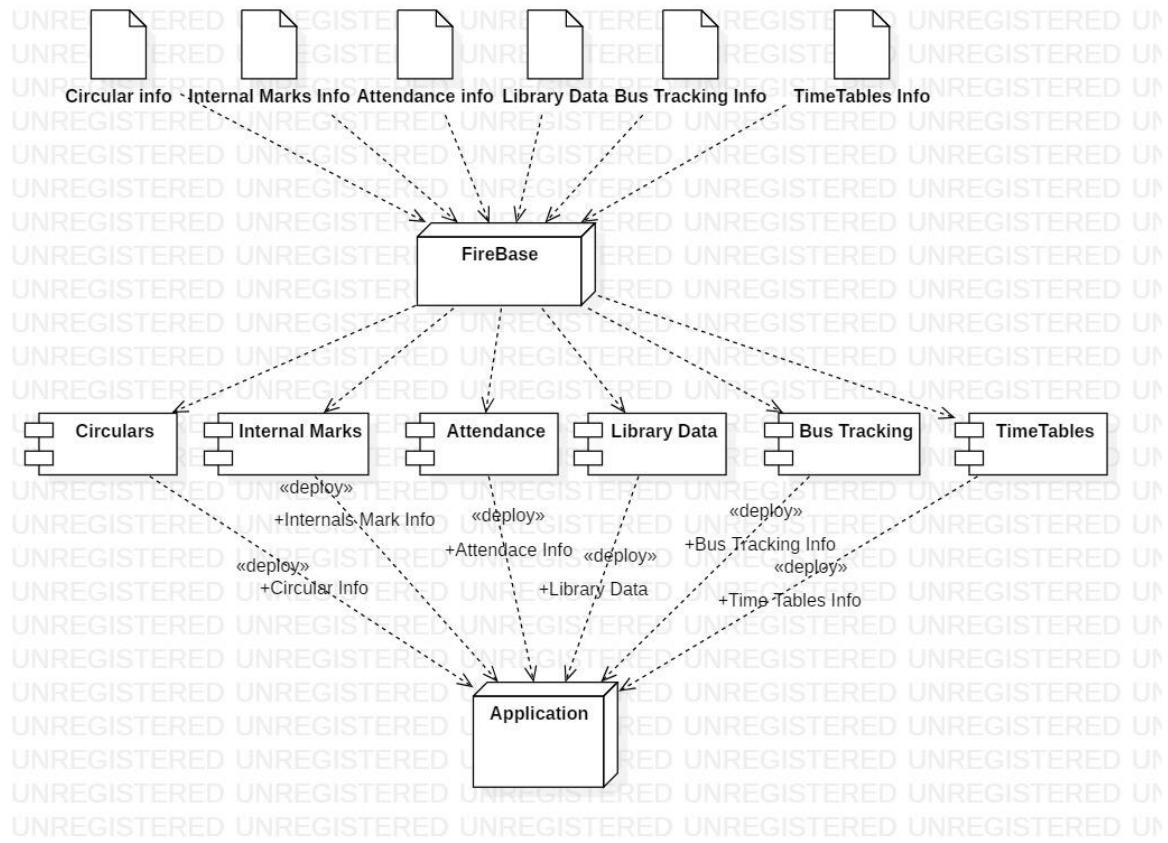
**Fig.3.13. Component Diagram for Smart College Management System**

### 3.3.11 Deployment Diagram for User:

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

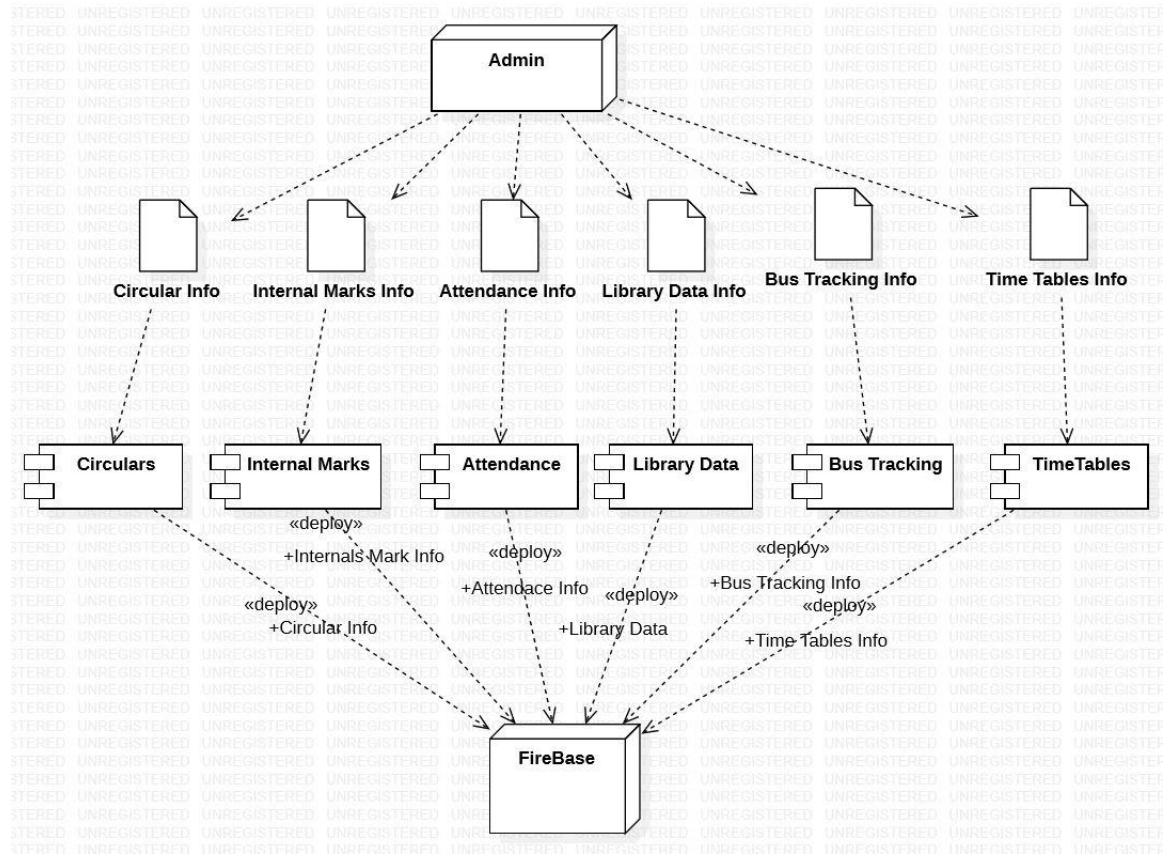
Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.



**Fig.3.14. Deployment Diagram for Smart College Management**

### 3.3.12 Deployment Diagram for Admin:



**Fig.3.15 Admin Deployment Diagram for Smart College Management System**

# **CHAPTER 4: IMPLEMENTATION AND RESULTS**

## **4.1 INTRODUCTION**

There are different modules present in student application. For each module they have different functionalities. They are:

### **➤ Circular Information**

In this circular information the student can know the details of every circular given by the particular department. For accessing the circulars student has to select the circular option provided in the application. The contents in the circular will open in the next window where there are different types of circulars which are sorted based on date modified. When the student clicks on the particular circular he can view the circular circulated by the admin. He can view that circular in jpg format where he can zoom in and zoom out.

### **➤ Time Tables**

In this Time Tables the user can view their semester time tables whenever they wish to see. The timetables are separated according to the branches. In this there are different drop down menus where we can select year branch and section after selecting that the timetable of that particular class will be displayed there itself in the jpg format. The user can see that timetable he can zoom in and zoom out for better experience.

### **➤ Internal Marks**

Internal marks are another module in which the user uses to know their internal marks at any time. The main problem when knowing the internal marks by traditional method is, it may consist of misconception while creating the form and the day it circulates all the students may not come to the college and also parents cannot know the internal marks of their students. Using this application everyone can view their internal marks anytime and anywhere. When they press the internal marks option provided in the application he/she see a search bar where the user searches using their college id number and they can view all the internal marks and grand total in the same window.

## ➤ **Attendance**

Knowing attendance is one of the biggest problem of the student. One should maintain minimum attendance for that semester to write the exams. Also the day the faculty circulates the attendance information students may absent and may not know the information about their attendance for that semester. In this application there is an option where the students can see their attendance. By clicking that option there is a search bar that will open in the next page. In the search bar the user search based on their roll number provided by the college. By clicking search button the data regarding that roll number will be displayed on the screen.

## ➤ **Library Data**

Library information like the availability of the book will be known only when the user personally goes to the library and searches for the particular book. Using this application those work will be reduced. The user can search the required book by this application. On clicking the option provided for the Library data there is a page displayed in the next window where the user find a list of books with author name and the available books in the library. The user may find difficulty for finding the book in that list so we provide a search option for better interaction. The user can search for the book in that search option he can find easily whether the book is available in the library or not.

## ➤ **Bus Tracking**

All the faculty and students who are using the bus facility in the college faces a problem like where the bus is currently locating. To reduce this problem we include this facility in our project. There is an option for live bus tracking in which we find the exact live status of that particular bus. To use this service the person who is in the bus share the location of the bus and it will broadcast the information to all the users using the bus facility.

## ➤ **Hostel Information**

To join in the college hostel one cannot know how many vacancies present in the hostel at the particular time. By using this application one can see the available vacancy information. There is also another option provided in the application that the students

can apply to hold a bed for that student. All he/she needs is to fill the form which are provided in the application. On clicking new application a form will open in the new window. After filling the form click on submit, the request will be sent to the admin and admin will contact with the student.

When it comes to the Admin application the admin has the permission to modify the content that is visible in the student application. In the Admin application all the services provided in the student application will be provided with write permissions. Admin application links to the Firebase that he/she can upload or delete the content of any service. The services like Attendance, Internal Marks, Circulars, Library Data, Time Tables etc.

#### **4.1.1 About Android Studio:**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

#### **4.1.2 Project Structure:**

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules

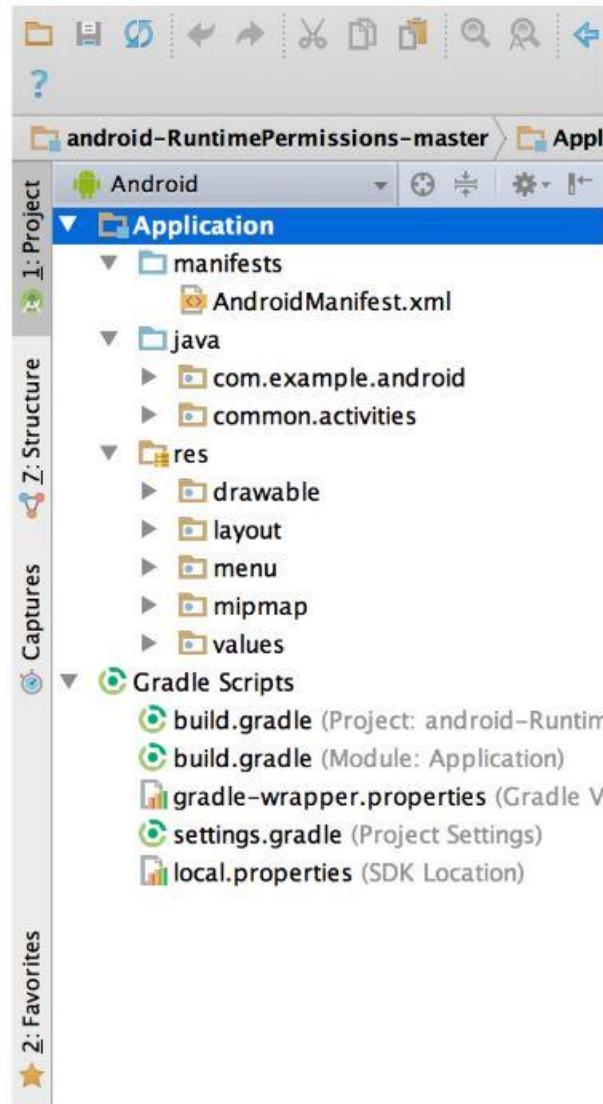
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

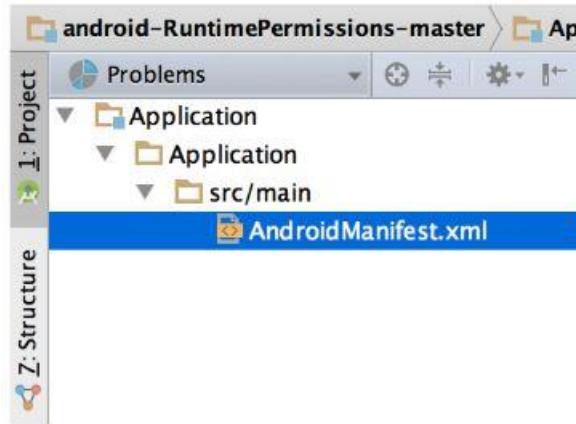
- **Manifests**: Contains the `AndroidManifest.xml` file.
- **Java**: Contains the Java source code files, including JUnit test code.
- **Res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in figure 1, it's showing as **Android**).



**Fig.4.1: The project files in Android view.**

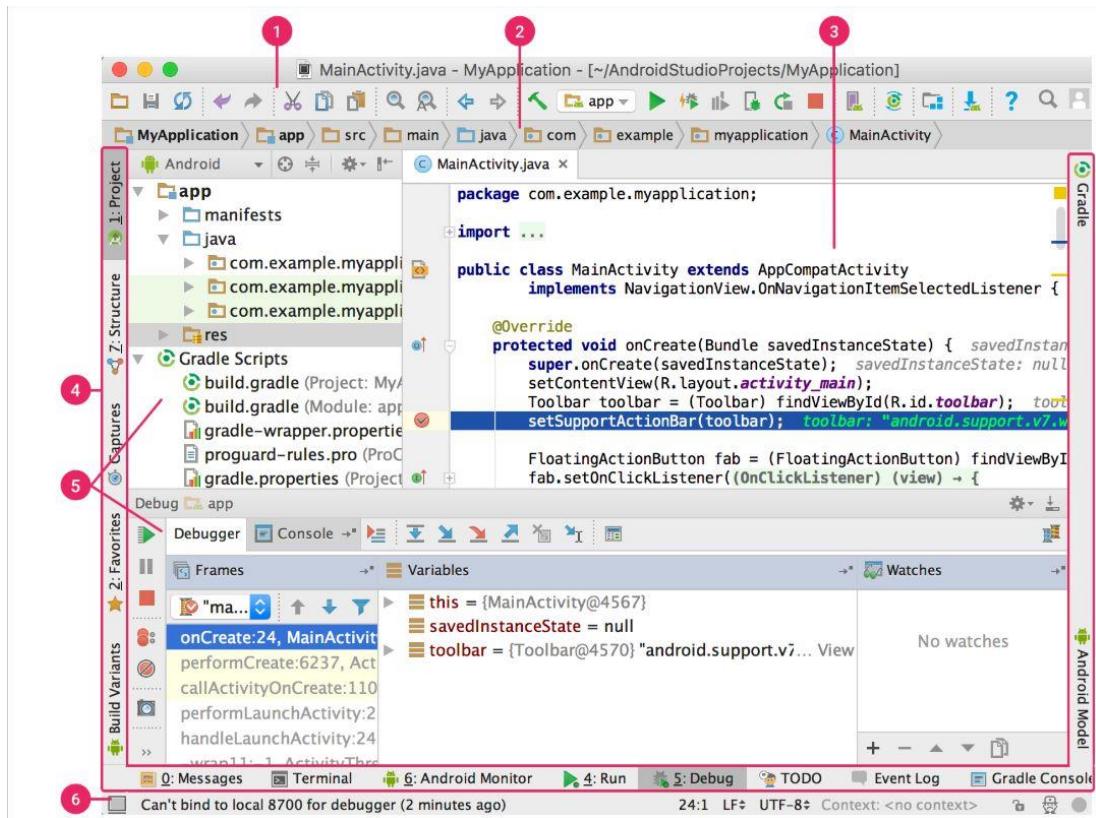
You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the **Problems** view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.



**Fig.4.2:** The project files in Problems view, showing a layout file with a problem.

#### 4.1.3 The User Interface:

The Android Studio main window is made up of several logical areas identified in figure 3.



**Fig.4.3:** The Android Studio main window.

1. The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.

2. The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
3. The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
5. The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
6. The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

#### **4.1.4 Gradle Build System:**

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.
- Reuse code and resources across source sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named `build.gradle`. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

## 4.2 EXPLANATION OF KEY FUNCTIONS

### 4.2.1 Activities:

Activities are one of the fundamental building blocks of apps on the Android platform. They serve as the entry point for a user's interaction with an app, and are also central to how a user navigates within an app (as with the Back button) or between apps (as with the Recents button).

Skillfully managing activities allows you to ensure that, for example:

- Orientation changes take place smoothly without disrupting the user experience.
- User data is not lost during activity transitions.
- The system kills processes when it's appropriate to do so.

This section begins by providing an introduction to the concept of activities. It goes on to describe the activity lifecycle in detail. Next, it discusses state changes and how to accommodate them. After that, this section talks about the relationship between activities and intra- and inter-app navigation. Last, this section explains the relationship between activities and the processes that host them.

### 4.2.2 Intent and Intent Filters:

An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases:

- **Starting an activity**

An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an Intent to `startActivity()`. The Intent describes the activity to start and carries any necessary data.

If you want to receive a result from the activity when it finishes, call `startActivityForResult()`. Your activity receives the result as a separate Intent object in your activity's `onActivityResult()` callback. For more information, see the Activities guide.

- **Starting a service**

A Service is a component that performs operations in the background without a user interface. With Android 5.0 (API level 21) and later, you can start a service with JobScheduler. For more information about JobScheduler, see its API-reference documentation.

For versions earlier than Android 5.0 (API level 21), you can start a service by using methods of the Service class. You can start a service to perform a one-time operation (such as downloading a file) by passing an Intent to startService(). The Intent describes the service to start and carries any necessary data.

If the service is designed with a client-server interface, you can bind to the service from another component by passing an Intent to bindService(). For more information, see the Services guide.

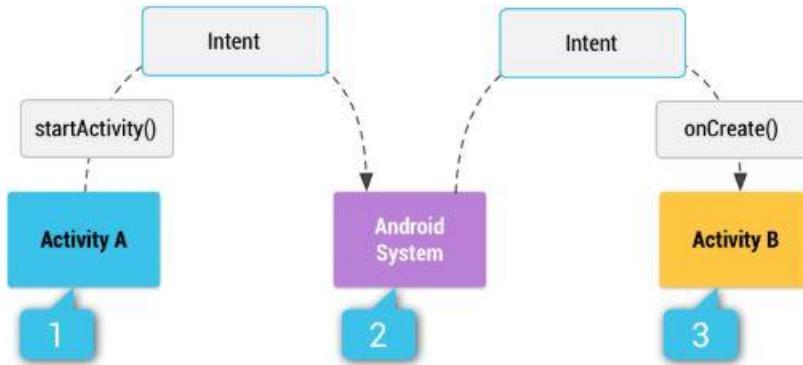
- **Delivering a broadcast**

A broadcast is a message that any app can receive. The system delivers various broadcasts for system events, such as when the system boots up or the device starts charging. You can deliver a broadcast to other apps by passing an Intent to sendBroadcast() or sendOrderedBroadcast().

#### 4.2.3 Intent types:

There are two types of intents:

- **Explicit intents** specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start. For example, you might start a new activity within your app in response to a user action, or start a service to download a file in the background.
- **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.



**Fig.4.4: Intent diagram**

How an implicit intent is delivered through the system to start another activity:

1. *Activity A* creates an Intent with an action description and passes it to `startActivity()`.
2. The Android System searches all apps for an intent filter that matches the intent. When a match is found.
3. The system starts the matching activity (*Activity B*) by invoking its `onCreate()` method and passing it the Intent.

When you use an implicit intent, the Android system finds the appropriate component to start by comparing the contents of the intent to the *intent filters* declared in the manifest file of other apps on the device. If the intent matches an intent filter, the system starts that component and delivers it the Intent object. If multiple intent filters are compatible, the system displays a dialog so the user can pick which app to use.

An intent filter is an expression in an app's manifest file that specifies the type of intents that the component would like to receive. For instance, by declaring an intent filter for an activity, you make it possible for other apps to directly start your activity with a certain kind of intent. Likewise, if you do *not* declare any intent filters for an activity, then it can be started only with an explicit intent.

#### 4.2.4 Background Tasks:

Every Android app has a main thread which is in charge of handling UI (including measuring and drawing views), coordinating user interactions, and receiving lifecycle events. If there is too much work happening on this thread, the app appears to hang or slow down, leading to an undesirable user experience. Any long-running computations and operations such as decoding a bitmap, accessing the disk, or performing network requests should be done on a separate background thread. In general, anything that takes more than a few milliseconds should

be delegated to a background thread. Some of these tasks may be required to be performed while the user is actively interacting with the app. To learn how you can run tasks on background threads and off the main UI thread while the app is actively being used, please take a look at the threading solutions guide.

Applications may also require some tasks to run even when the user is not actively using the app such as syncing periodically with a backend server or fetching new content within an app on a periodic basis. Applications may also require services to run immediately to completion even after the user has completed interacting with the app. This guide will help you learn which solution best meets your needs for these use cases.

#### 4.2.5 Event Listeners:

An event listener is an interface in the `View` class that contains a single callback method. These methods will be called by the Android framework when the `View` to which the listener has been registered is triggered by user interaction with the item in the UI.

Included in the event listener interfaces are the following callback methods:

- **onClick()**

From `View.OnClickListener`. This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.

- **onLongClick()**

From `View.OnLongClickListener`. This is called when the user either touches and holds the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses and holds the suitable "enter" key or presses and holds down on the trackball (for one second).

- **onFocusChange()**

From `View.OnFocusChangeListener`. This is called when the user navigates onto or away from the item, using the navigation-keys or trackball.

- **onKey()**

From `View.OnKeyListener`. This is called when the user is focused on the item and presses or releases a hardware key on the device.

- `onTouch()`

From `View.OnTouchListener`. This is called when the user performs an action qualified as a touch event, including a press, a release, or any movement gesture on the screen (within the bounds of the item).

- `onCreateContextMenu()`

From `View.OnCreateContextMenuListener`. This is called when a Context Menu is being built (as the result of a sustained "long click"). See the discussion on context menus in the Menus developer guide.

These methods are the sole inhabitants of their respective interface. To define one of these methods and handle your events, implement the nested interface in your Activity or define it as an anonymous class. Then, pass an instance of your implementation to the respective `View.set...Listener()` method. (E.g., call `setOnClickListener()` and pass it your implementation of the `OnClickListener`.)

#### 4.2.6 Event Handlers:

If you're building a custom component from `View`, then you'll be able to define several callback methods used as default event handlers. In the document about `Custom_View Components`, you'll learn some of the common callbacks used for event handling, including:

- `onKeyDown(int, KeyEvent)` - Called when a new key event occurs.
- `onKeyUp(int, KeyEvent)` - Called when a key up event occurs.
- `onTrackballEvent(MotionEvent)` - Called when a trackball motion event occurs.
- `onTouchEvent(MotionEvent)` - Called when a touch screen motion event occurs.
- `onFocusChanged(boolean, int, Rect)` - Called when the view gains or loses focus.

There are some other methods that you should be aware of, which are not part of the `View` class, but can directly impact the way you're able to handle events. So, when managing more complex events inside a layout, consider these other methods:

- `Activity.dispatchTouchEvent(MotionEvent)` - This allows your Activity to intercept all touch events before they are dispatched to the window.
- `ViewGroup.onInterceptTouchEvent(MotionEvent)` - This allows a ViewGroup to watch events as they are dispatched to child Views.
- `ViewParent.requestDisallowInterceptTouchEvent(boolean)` - Call this upon a parent View to indicate that it should not intercept touch events with `onInterceptTouchEvent(MotionEvent)`.

## 4.3 FIREBASE:

### 4.3.1 Introduction

Cloud Storage for Firebase lets you upload and share user generated content, such as images and video, which allows you to build rich media content into your apps. Your data is stored in a Google Cloud Storage bucket, an exabyte scale object storage solution with high availability and global redundancy. Cloud Storage lets you securely upload these files directly from mobile devices and web browsers, handling spotty networks with ease.

### 4.3.2 How to connect Firebase to your Application:

Adding Firebase to your app involves tasks both in the Firebase console and in your open Android project (for example, you download Firebase config files from the console, then move them into your Android project).

#### ➤ Step 1: Create a Firebase project

Before you can add Firebase to your Android app, you need to create a Firebase project to connect to your Android app. Visit [Understand Firebase Projects](#) to learn more about Firebase projects.

Create a Firebase project

#### ➤ Step 2: Register your app with Firebase

After you have a Firebase project, you can add your Android app to it.

1. In the center of the Firebase console's project overview page, click the **Android** icon to launch the setup workflow.

If you've already added an app to your Firebase project, click **Add app** to display the platform options.

- An *application ID* is sometimes referred to as a *package name*.

- Find this application ID in your module (app-level) Gradle file, usually app/build.gradle (example ID: com.yourcompany.yourproject).
2. Enter your app's application ID in the **Android package name** field.  
*(Optional)* Enter other app information as prompted by the setup workflow.
  3. The nickname is an internal, convenience identifier and is only visible to you in the Firebase console.
  4. Click **Register app**.
- **Step 3: Add a Firebase configuration file**
1. Add the Firebase Android configuration file to your app:
    - a. Click **Download google-services.json** to obtain your Firebase Android config file (google-services.json).
    - b. You can download your Firebase Android config file again at any time.
    - c. Move your config file into the module (app-level) directory of your app.
  2. To enable Firebase products in your app, add the google-services plugin to your Gradle files.
    - a. In your root-level (project-level) Gradle file (build.gradle), add rules to include the Google Services plugin. Check that you have Google's Maven repository, as well.

```

buildscript {
// ...
dependencies {
// ...
// Add the following line:
classpath 'com.google.gms:google-services:4.2.0' // Google Services plugin
}
}

allprojects {
// ...

```

```
repositories {  
    // Check that you have the following line (if not, add it):  
    google() // Google's Maven repository  
    // ...  
}  
}
```

- b. In your module (app-level) Gradle file (usually app/build.gradle), add a line to the *bottom of the file*.

```
apply plugin: 'com.android.application'  
  
android {  
    // ...  
}  
  
// Add the following line to the bottom of the file:  
apply plugin: 'com.google.gms.google-services' // Google Play services Gradle plugin
```

#### ➤ Step 4: Add Firebase SDKs to your app

You can add any of the supported Firebase products to your Android app. We recommend starting with the core Firebase SDK (com.google.firebaseio.firebaseio-core), which provides Google Analytics for Firebase functionality.

1. In your module (app-level) Gradle file (usually app/build.gradle), add the dependency for the core Firebase SDK:

```
dependencies {  
    // ...  
    implementation 'com.google.firebaseio.firebaseio-core:16.0.8'  
  
    // Getting a "Could not find" error? Make sure that you've added  
    // Google's Maven repository to your root-level build.gradle file  
}
```

2. (*Optional*) Add the dependencies for the other Firebase products that you want to use.
3. Sync your app to ensure that all dependencies have the necessary versions.

- Run your app to send verification to Firebase that you've successfully integrated Firebase.

Your device logs will display the Firebase verification that initialization is complete. If you ran your app on an emulator that has network access, the Firebase console notifies you that your app connection is complete.

#### **4.4 DATABASE TABLES:**

##### **Attendance Table:**

<b>name</b>	<b>roll</b>	<b>status</b>
Naveen	15ak1a0544	Present
Nanda Kishore Reddy	15ak1a0542	Absent
Kiran	15ak1a0536	Present

**Table 4.4.1: Attendance Table.**

##### **Circulars Table:**

<b>Title</b>	<b>Description</b>	<b>Image_id</b>
Circular	National Science Day	1
Circular_21/03/2019	UGC committee inspection	2
Placement Circular	Summer internship program	3

**Table 4.4.2: Circulars Table.**

##### **Library Data Table:**

<b>Title</b>	<b>Author</b>	<b>Count</b>
Agile project management	Jim Highsmith	16
Hadoop the definitive guide	Tom White	4

**Table 4.4.3: Library Data Table.**

## 4.5 OUTPUT SCREENS

Showing the application on the mobile screen



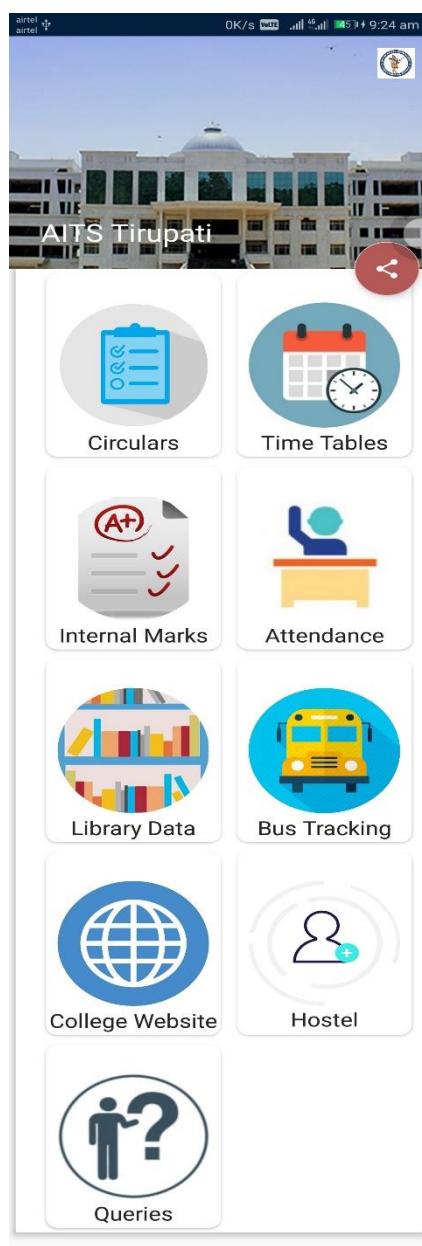
**Fig.4.5: Showing Applications on Mobile Screen**

#### 4.5.1 Output Screens of User Application

##### Home Screen

The home screen contains many UI cards which are dedicated to different tasks like Circulars, Time Tables, Internal Marks, Attendance, Library data, Bus Tracking etc. whenever you clicked the above shown buttons you will be lead to their respective screens

The interface of home screen be like,

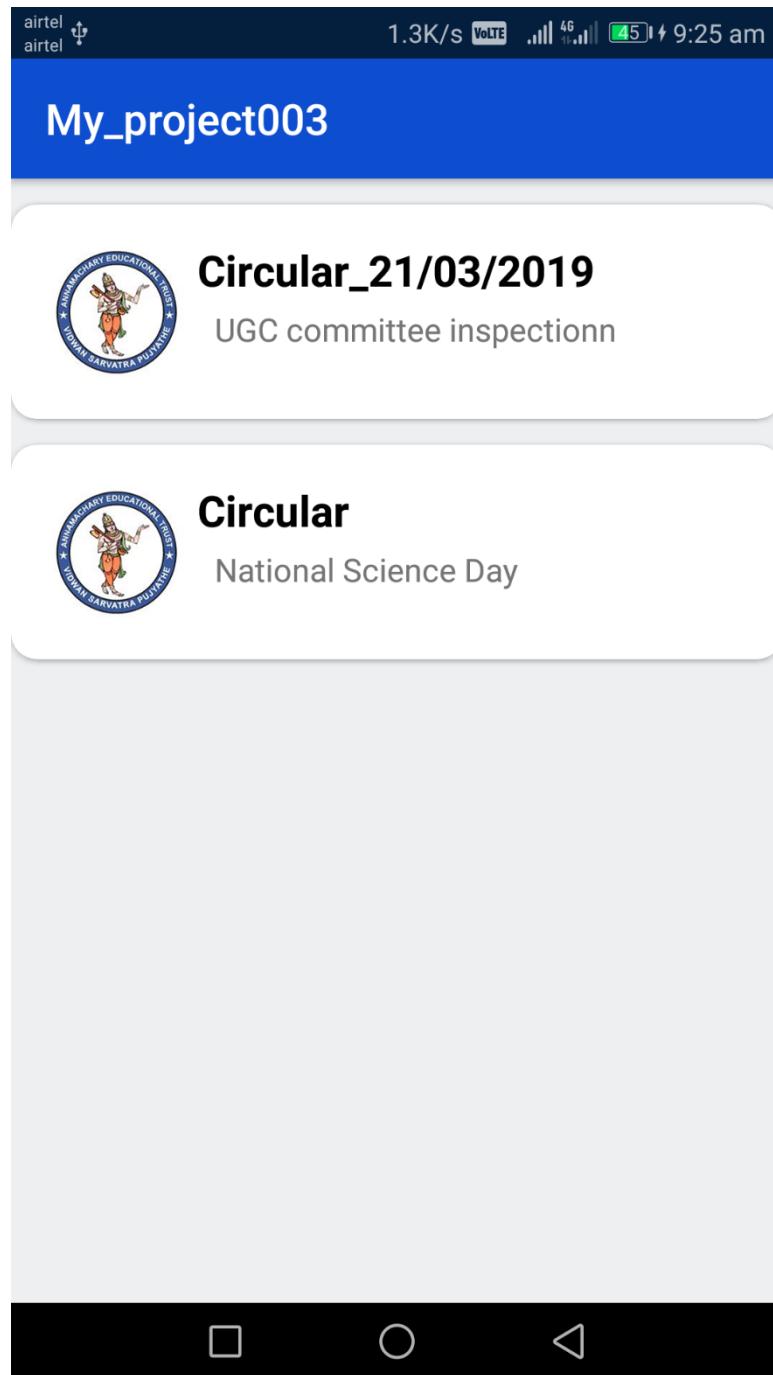


**Fig.4.6: Home Screen of the Project**

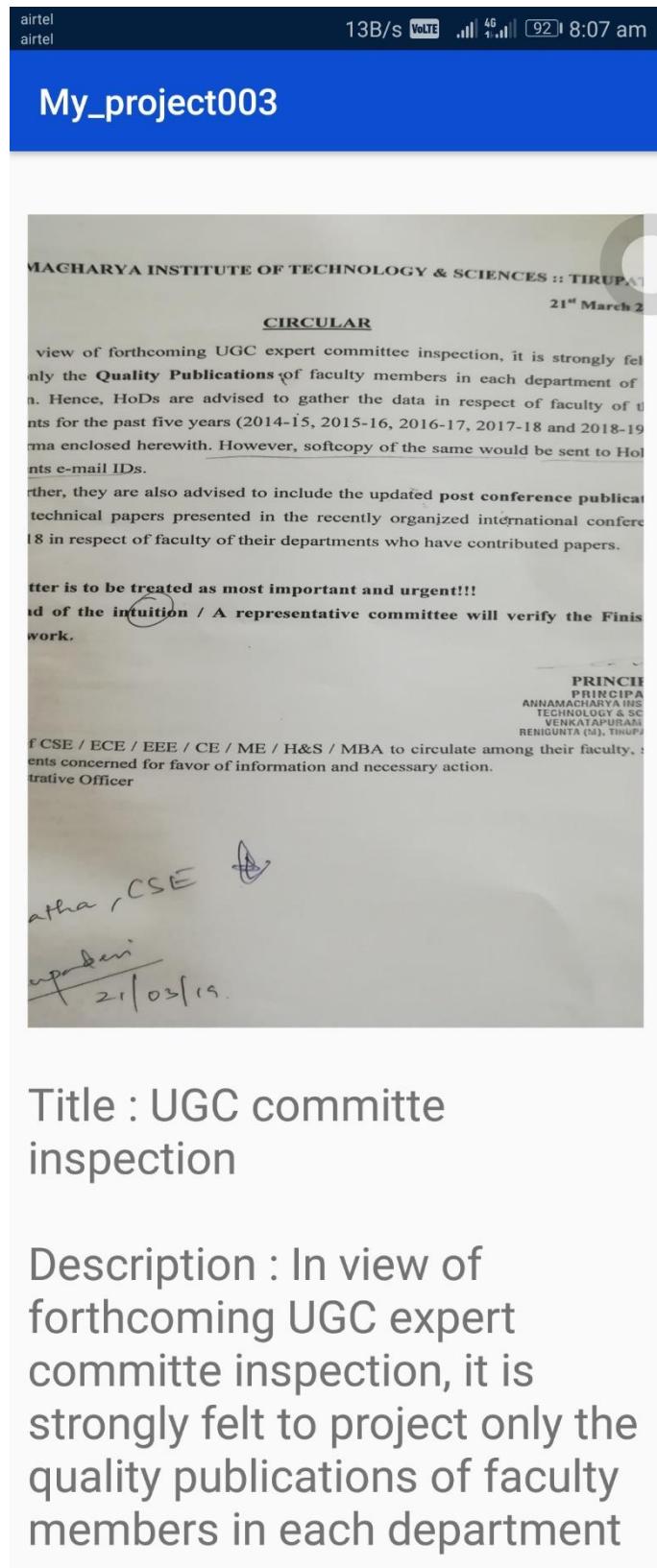
### **Circulars page:**

In this page the student can access all the circulars which are provided by the college.

The Circulars that are available in that page are up to date and are listed based on the date modified.



**Fig.4.7: User Circulars page of the project**



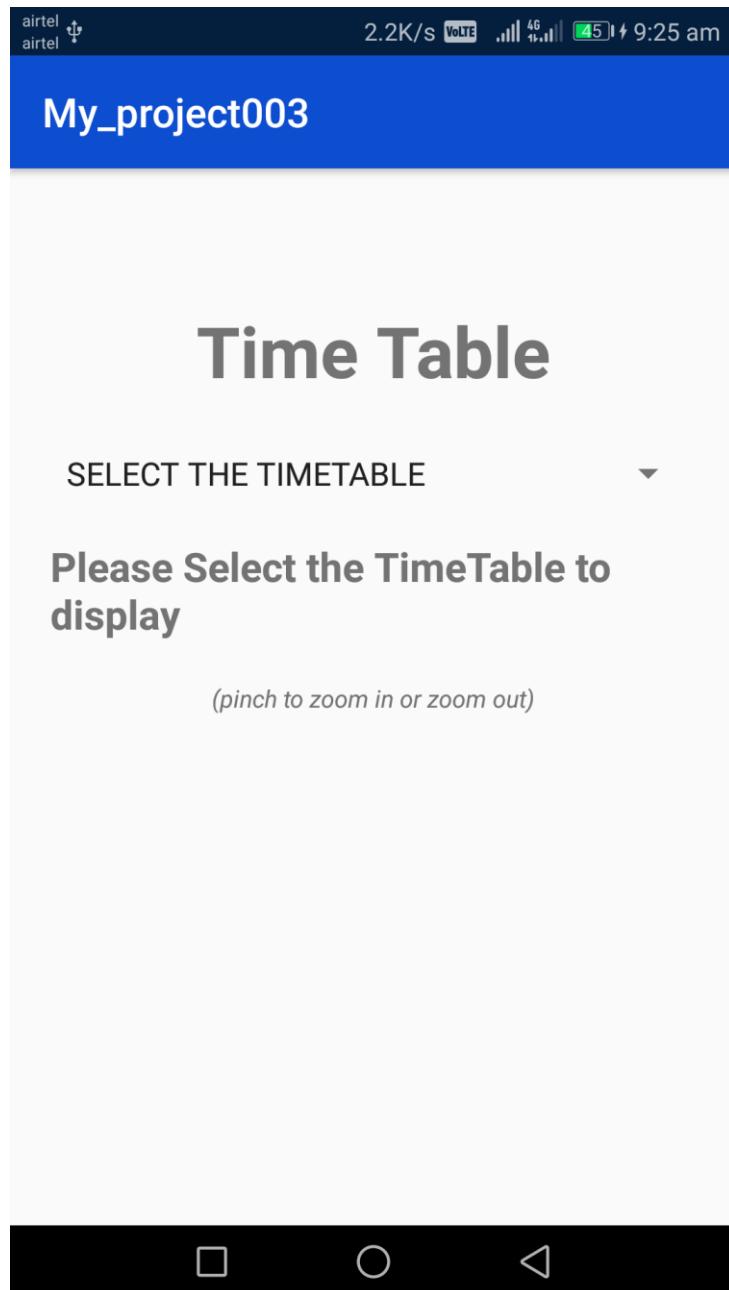
## Title : UGC committe inspection

Description : In view of forthcoming UGC expert committe inspection, it is strongly felt to project only the quality publications of faculty members in each department

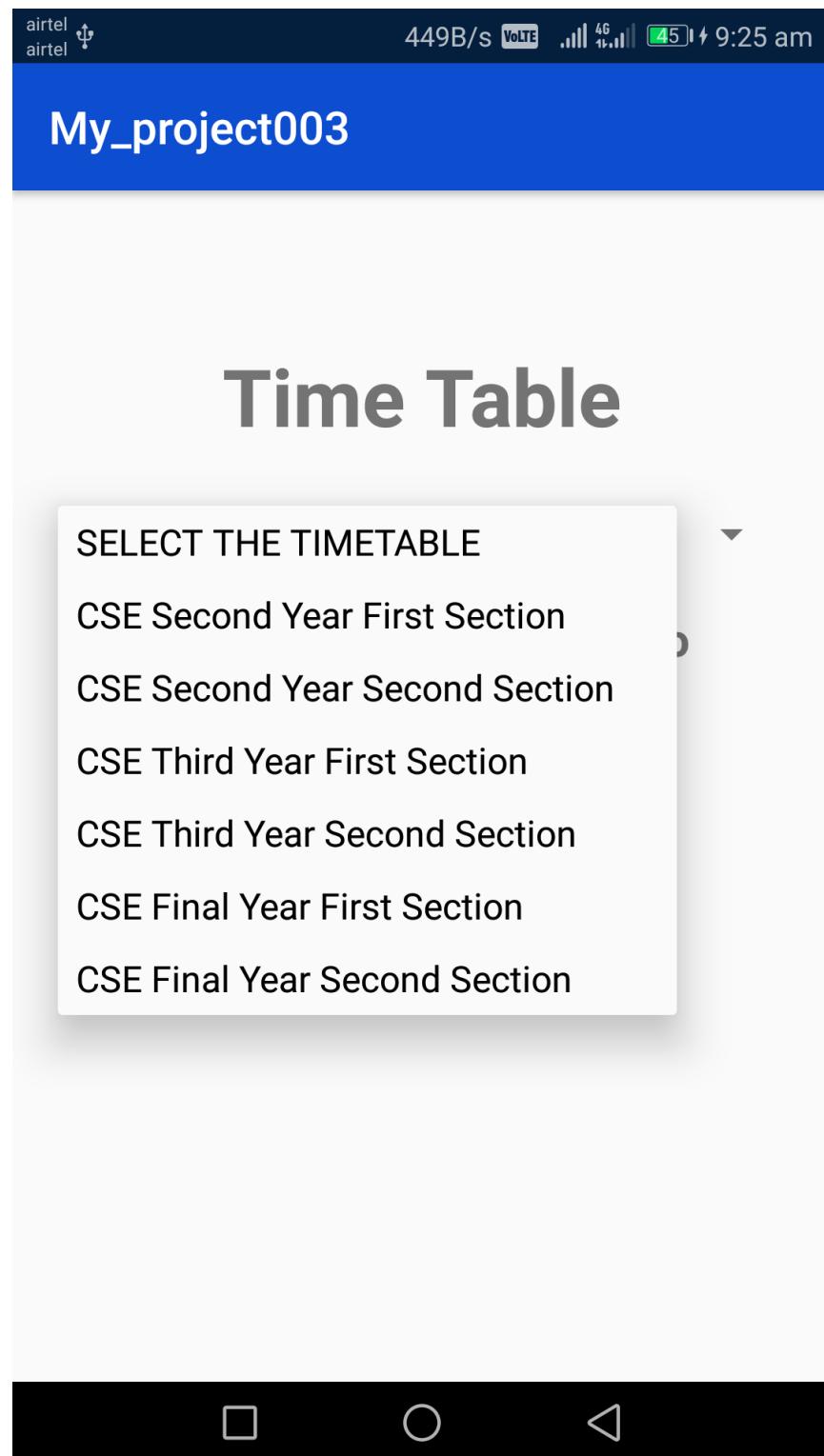
Fig.4.8.Displaying Circulars on the Screen.

### **Time Tables page:**

In this page student can access the time tables of particular year and section. There is a drop down menu in which students can select their time tables. On clicking the timetable, the image of time table will display on the screen in which we can zoom in and zoom out the image.



**Fig.4.9: User Time Tables screen**



**Fig.4.10: Selecting the time table**

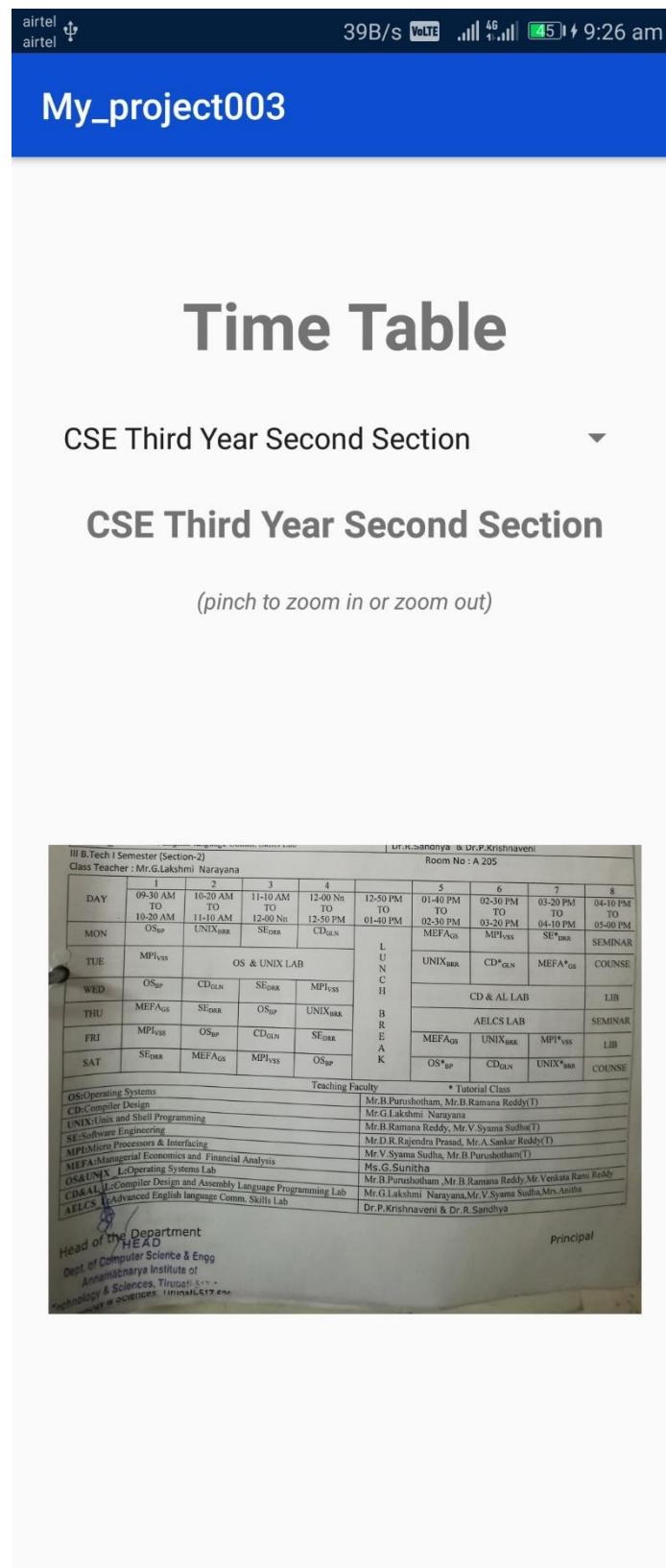
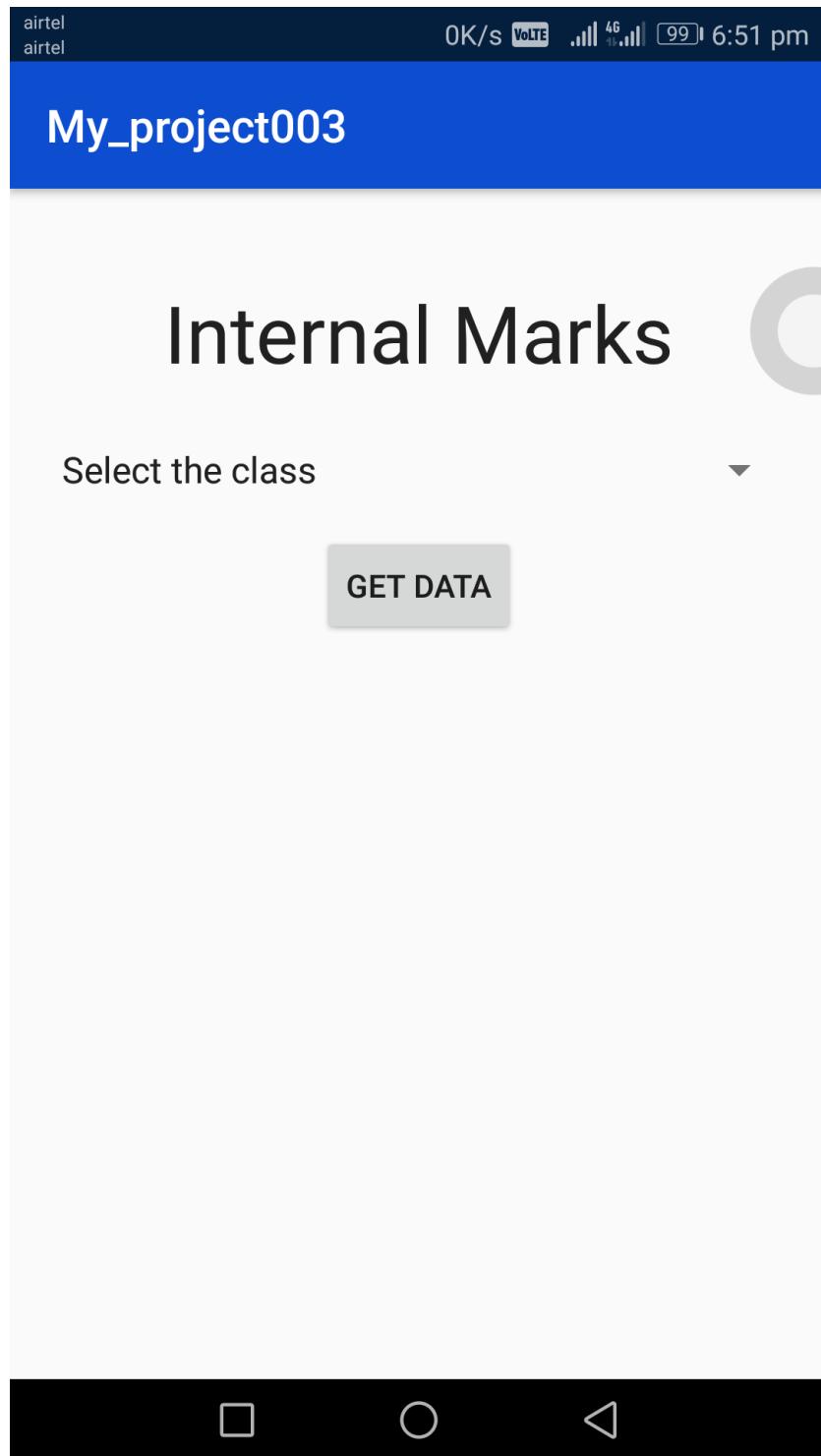


Fig.4.11: Displaying the Time Table on the Screen

**Internal marks:**

In this page students can be able to search their internal marks, view their internal marks whenever they want to know their marks.



**Fig.4.12:** User Internal marks page

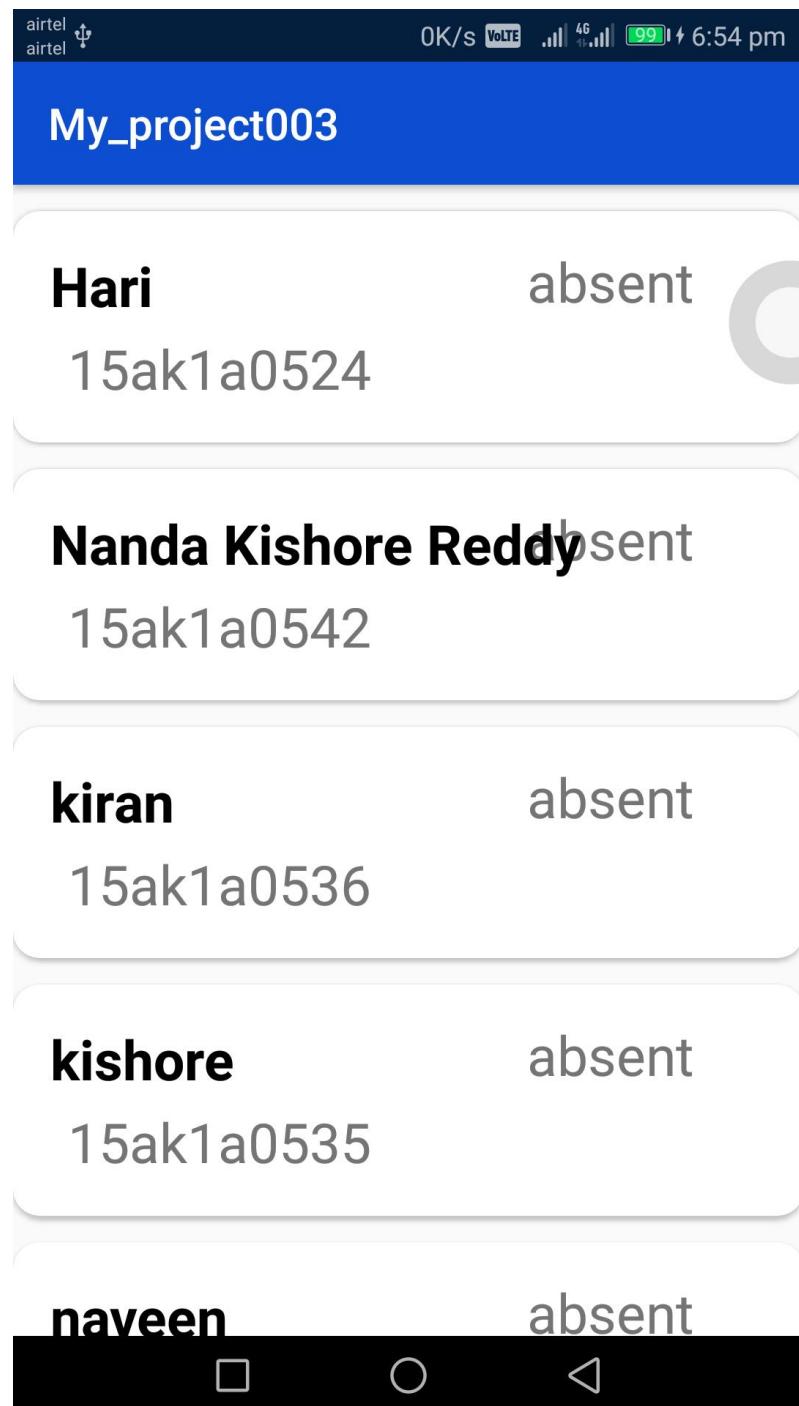
The screenshot shows a mobile application interface. At the top, there's a header bar with the text "airtel" on the left and "OK/s VoLTE 4G 92% 8:06 am" on the right. Below the header is a blue navigation bar with the text "My\_project003". The main content area has a large title "Internal Marks" in bold black font. Underneath it, there's a sub-section title "FINAL CSE B". In the center, there's a button labeled "GET DATA". The main data area is a table with 90 rows and 11 columns. The columns represent various subjects and skills, with some column headers being partially cut off. The rows contain numerical values representing student marks.

SNO.	HallTicketNumber	15A0560 2####DA 1####C MPILER DESIGN	15A0560 3####DE WAREH OUSING & MINING	15A0560 4####DE SIGN & ANALYSI S OF ALGORI THMS	15A0560 5####W EB & INTERN ET TECHNO LOGIES	15A0560 6####AR TIFICIAL INTELLI GENCE	15A0560 9####W EB & INTERN ET TECHNO LOGIES LAB	15A0561 0####DA WAREH OUSING & MINING LAB	15A5260 2####AD VANCED ENGLIS H LANGUA GE COMMU N SKILLS LAB	
48	15AK1A0559	30	30	28	28	25	30	25	29	28
49	15AK1A0560	17	13	15	15	11	14	23	27	17
50	15AK1A0561	23	15	18	16	12	11	23	27	24
51	15AK1A0562	30	29	28	28	27	25	25	29	29
52	15AK1A0563	26	27	24	26	22	29	21	28	29
53	15AK1A0564	29	22	26	28	28	28	27	30	29
54	15AK1A0566	16	15	18	18	16	19	21	28	16
55	15AK1A0567	18	15	18	15	13	17	21	27	27
56	15AK1A0568	29	26	26	28	25	29	24	29	29
57	15AK1A0569	30	28	30	29	29	30	24	30	29
58	15AK1A0570	29	27	30	27	29	29	24	30	30
59	15AK1A0571	28	25	21	26	24	26	24	28	29
60	15AK1A0573	19	18	19	21	20	21	24	28	28
61	15AK1A0576	28	25	22	24	23	30	22	29	29
62	15AK1A0577	30	29	30	28	27	30	24	30	29
63	15AK1A0578	30	27	27	27	27	27	25	30	29
64	15AK1A0579	29	30	28	23	26	29	25	29	29
65	15AK1A0580	18	23	25	20	26	23	21	27	26
66	15AK1A0581	26	25	24	24	21	28	25	27	29
67	15AK1A0584	26	24	25	24	22	28	25	29	30
68	15AK1A0585	26	25	29	21	24	26	24	29	29
69	15AK1A0586	26	25	24	21	18	25	24	29	27
70	15AK1A0587	21	18	23	20	17	25	24	28	20
71	15AK1A0588	30	27	28	26	24	29	24	30	30
72	15AK1A0589	24	21	23	23	19	25	24	28	29
73	15AK1A0590	30	26	27	24	25	29	21	29	29
74	15AK1A0591	27	26	22	18	16	23	22	27	29
75	15AK1A0592	15	12	18	15	17	18	21	27	22
76	15AK1A0593	30	25	29	28	23	27	24	29	26
77	15AK1A0594	27	23	24	25	21	24	24	27	29
78	15AK1A0595	30	14	23	23	18	21	24	28	29
79	15AK1A0596	26	23	23	24	25	28	27	29	28
80	15AK1A0597	25	18	23	17	19	22	23	28	26
81	15AK1A0598	29	30	26	27	27	26	25	29	30
82	15AK1A0599	22	16	16	16	0	0	24	27	29
83	15AK1A05A0	16	18	15	15	14	21	21	27	16
84	15AK1A05A1	29	22	26	23	23	28	24	28	29
85	15AK1A05A2	13	11	11	16	16	16	23	27	16
86	15AK1A05A4	29	27	25	27	28	29	21	30	30
87	15AK1A05A5	30	26	28	28	29	30	25	30	30
88	15AK1A05A6	28	23	24	24	23	29	21	30	30
89	15AK1A05A7	30	30	30	27	29	30	22	30	30
90	14AK1A0518	30	27	29	24	25	29	22	30	27

Fig.4.13: Displaying internal marks

### **Attendance page:**

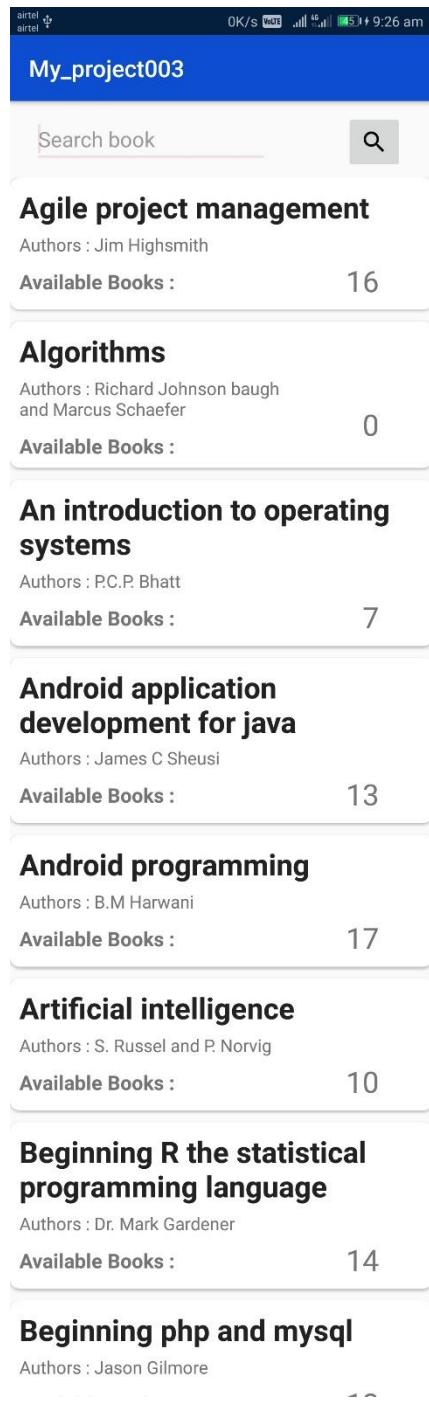
In this page student can view their attendance whenever they need. This attendance is updated every month. This feature will help the students to maintain good attendance and to attend classes everyday.



**Fig.4.14: User Attendance page**

## **Library Data:**

In this student can find the available books of a particular book can be known without going to the library. Student can even search the books by book name to find the availability of that book. The author details of that book also provided for better recognition.



**Fig.4.15: User Library data page**

### Bus Tracking page:

This feature provides the user to find the college busses live location status. This is the optional feature provided in the application. The location status of the bus will be shared in peer to peer manner.

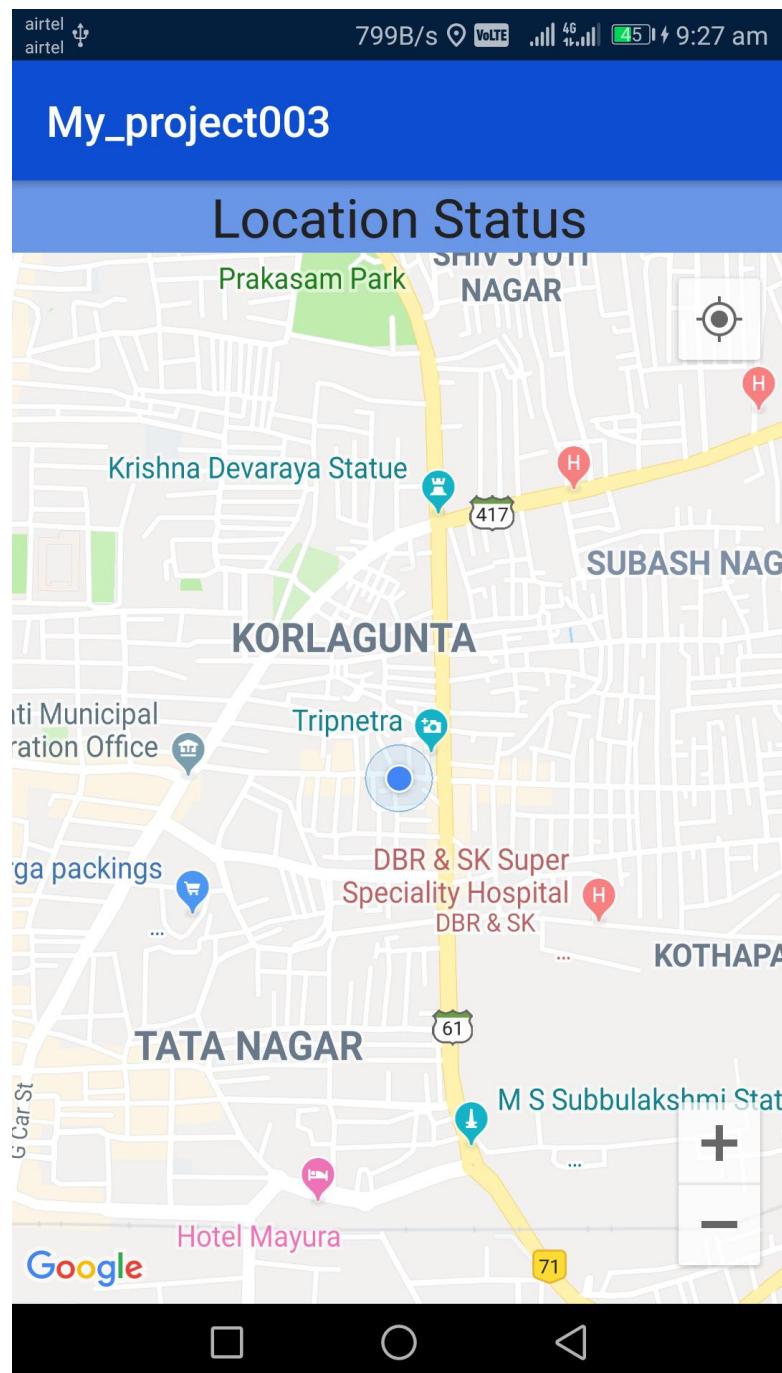


Fig.4.16: Bus Tracking page.

### **College Website:**

In this page the user can know the details about college like founder, mission & vision, faculty details, gallery etc.



## **Engineering College in Tirupati**

Annamacharya Institute of Technology & Sciences, Tirupati (AITST) was established in the year 2007, with a belief that education can influence and mould young minds.

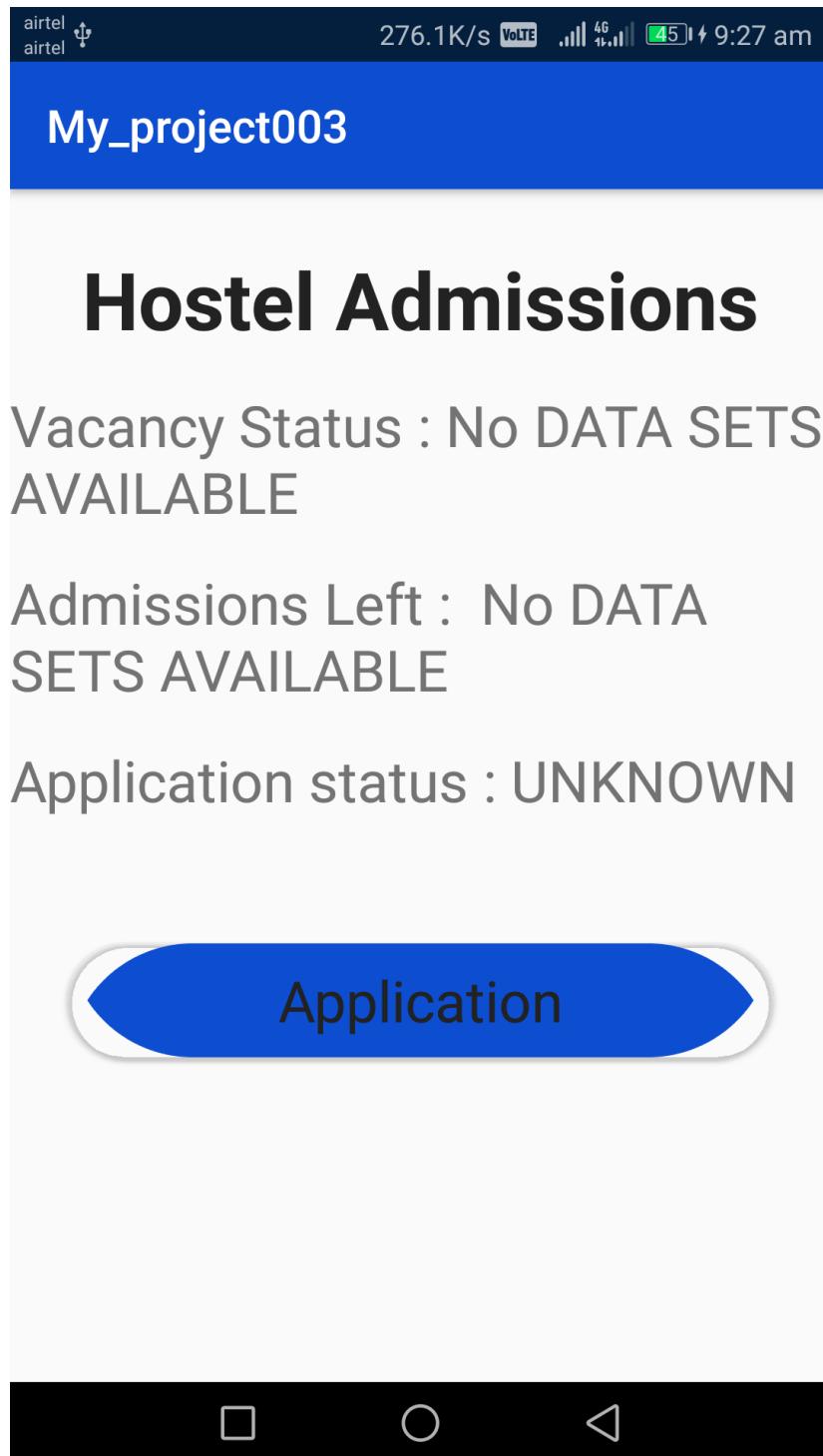
Within a decade, AITS, Tirupati received the



**Fig.4.17: College Website page.**

**Hostel Information:**

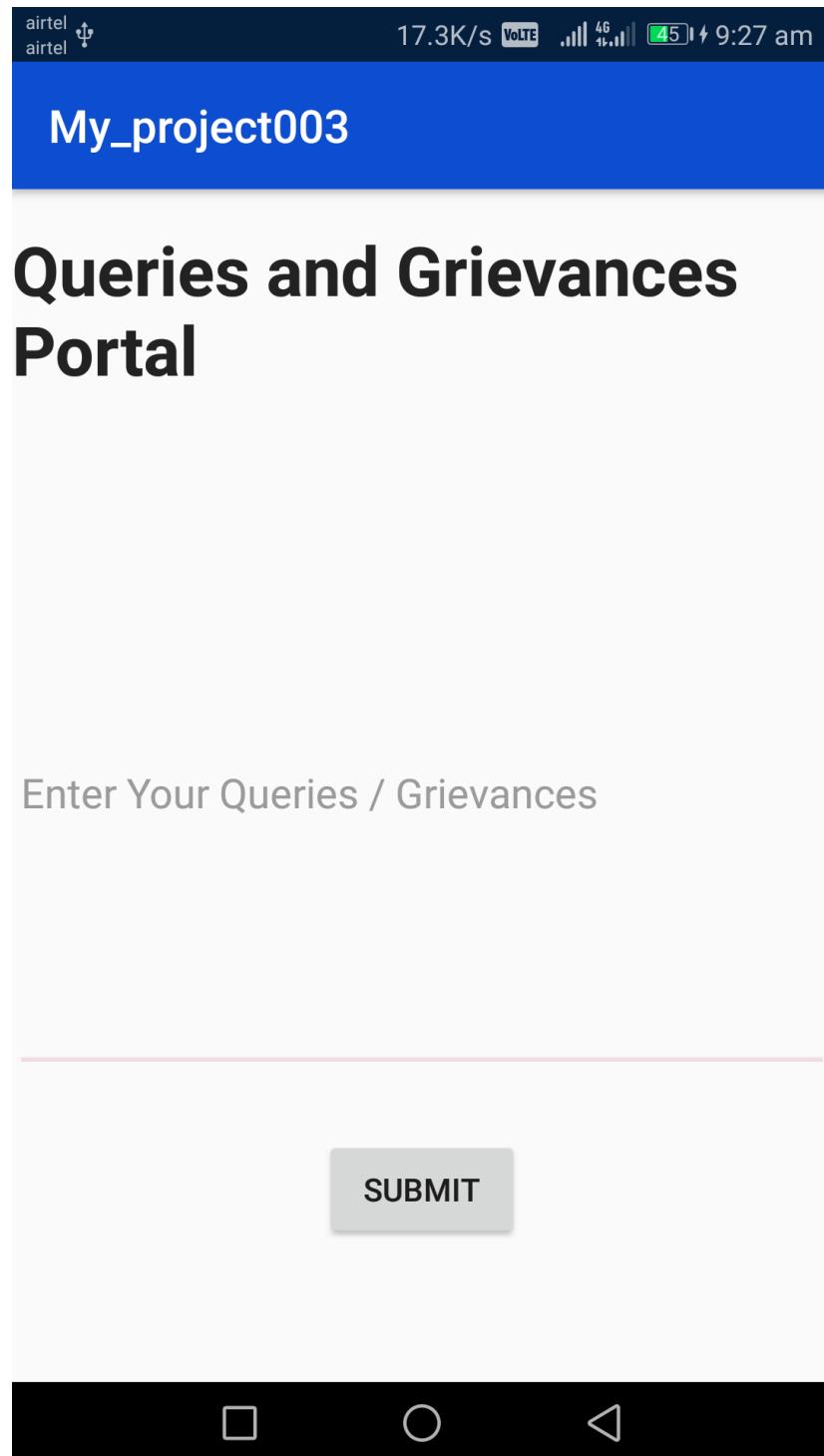
In this page the user can know the vacancy status, Admissions left and the applications left. Students who are willing to join in the hostel can also register for the room.



**Fig.4.18: Hostel Management.**

### **Queries page:**

In this page students can give feedback and also can give complaints to the management.



**Fig.4.19: Queries page.**

#### 4.5.2 Output Screens of Admin Application:

Login page:

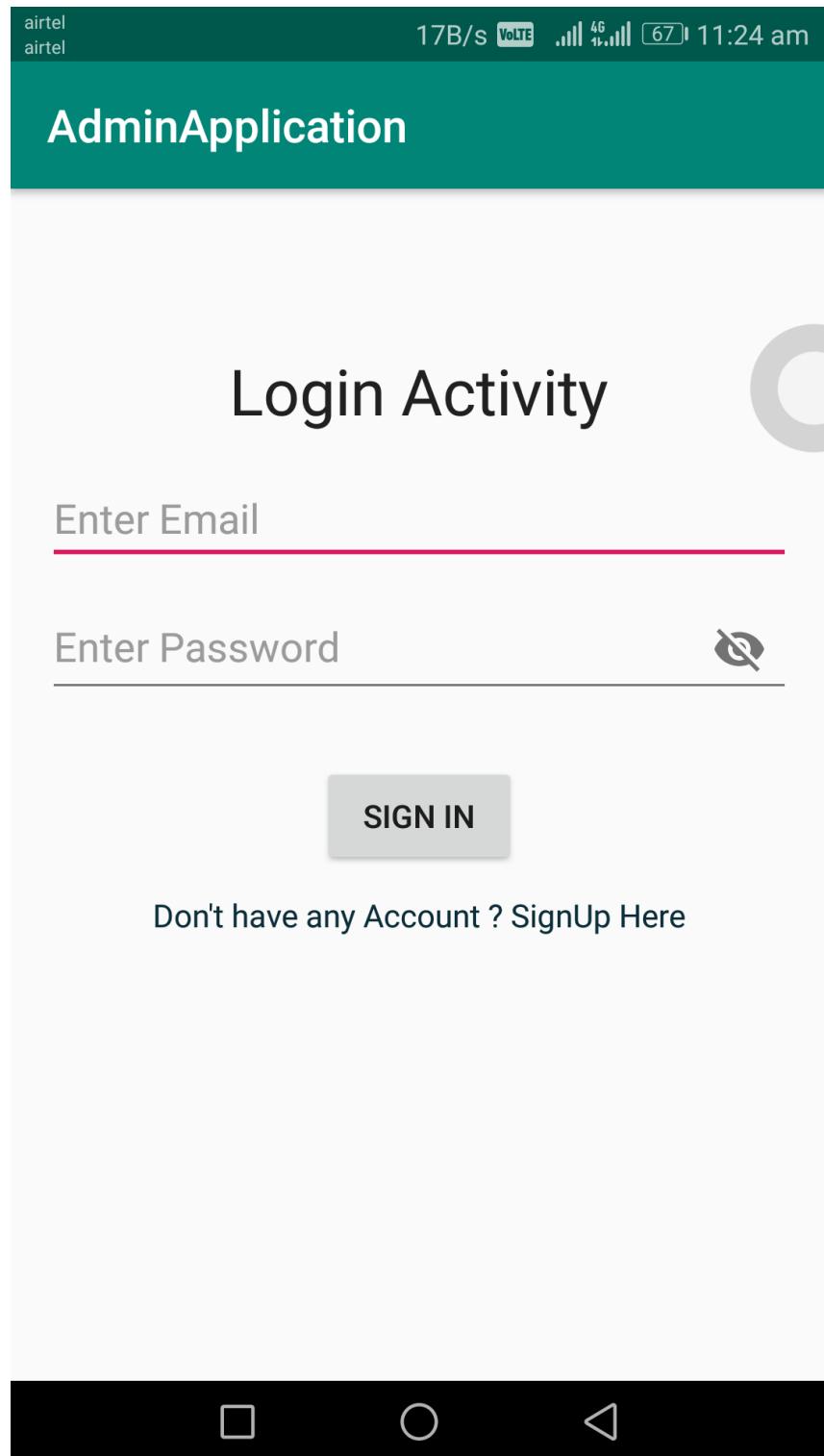
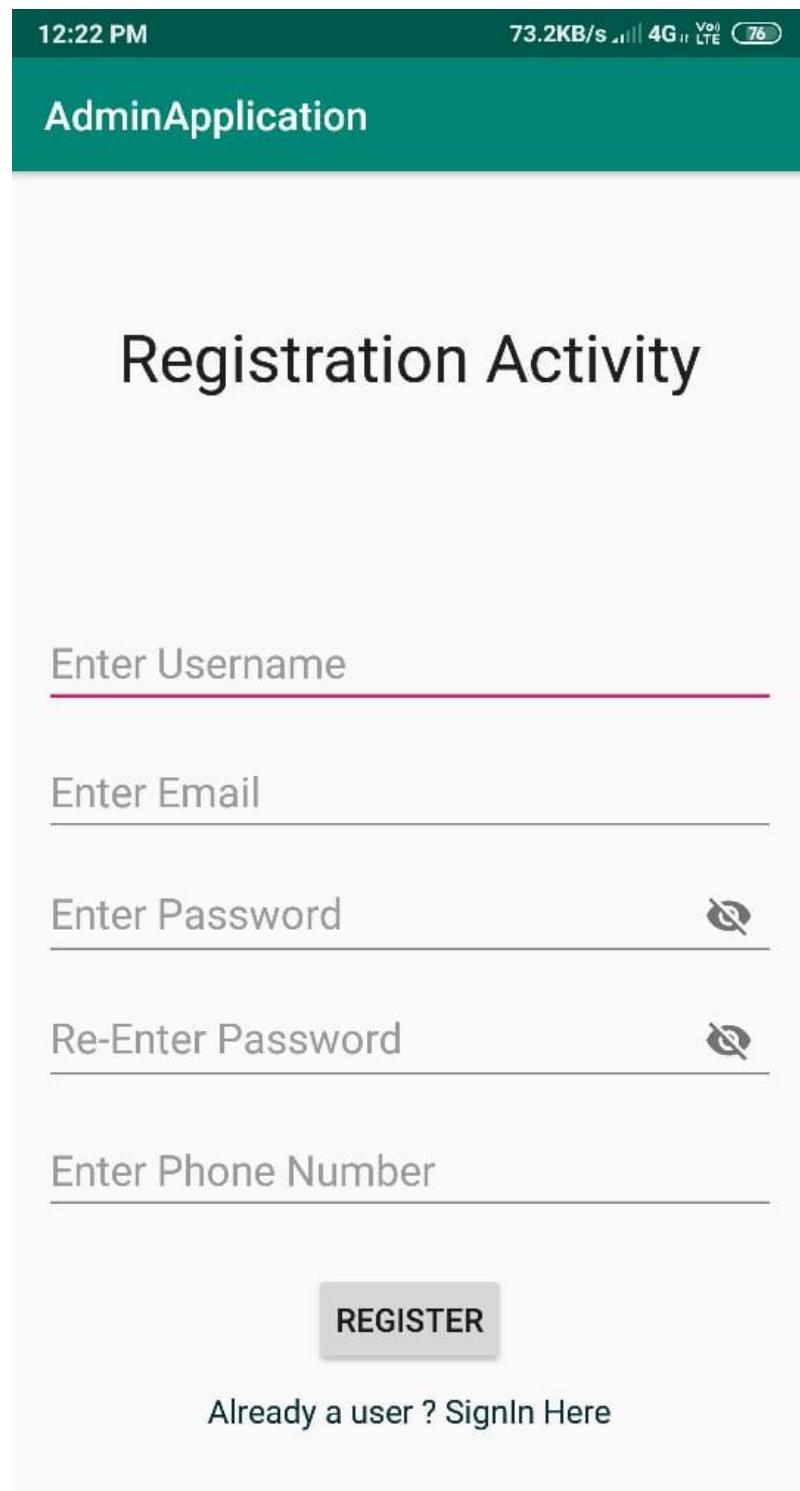


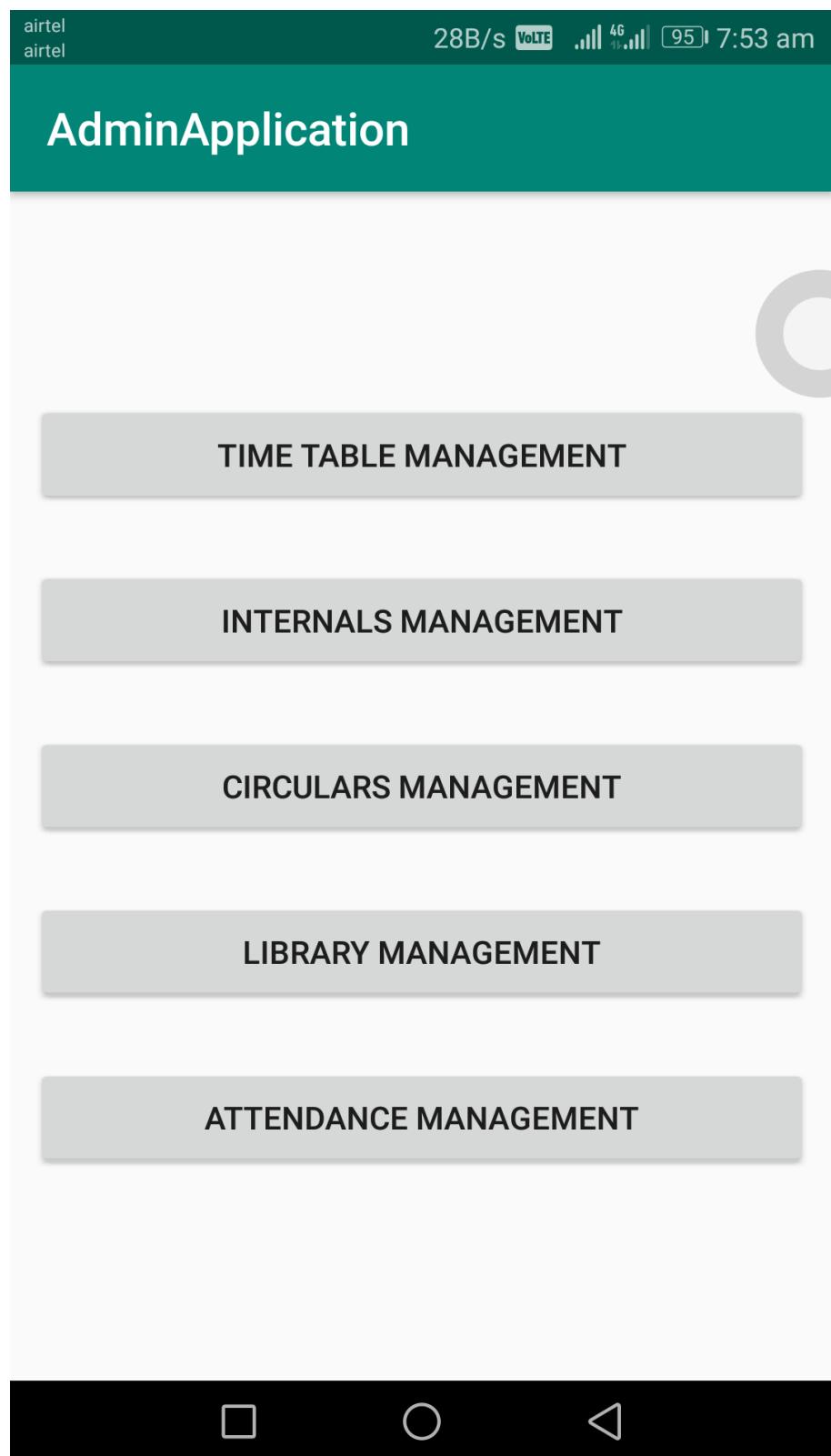
Fig.4.20: Login Screen.

**Registration Form:**



**Fig.4.21: Registration Screen**

**Home Screen:**



**Fig.4.22: Home Screen of Admin Application**

### Time Table Management page:

In this page admin can update and delete the contents by selecting the branch, year, section.

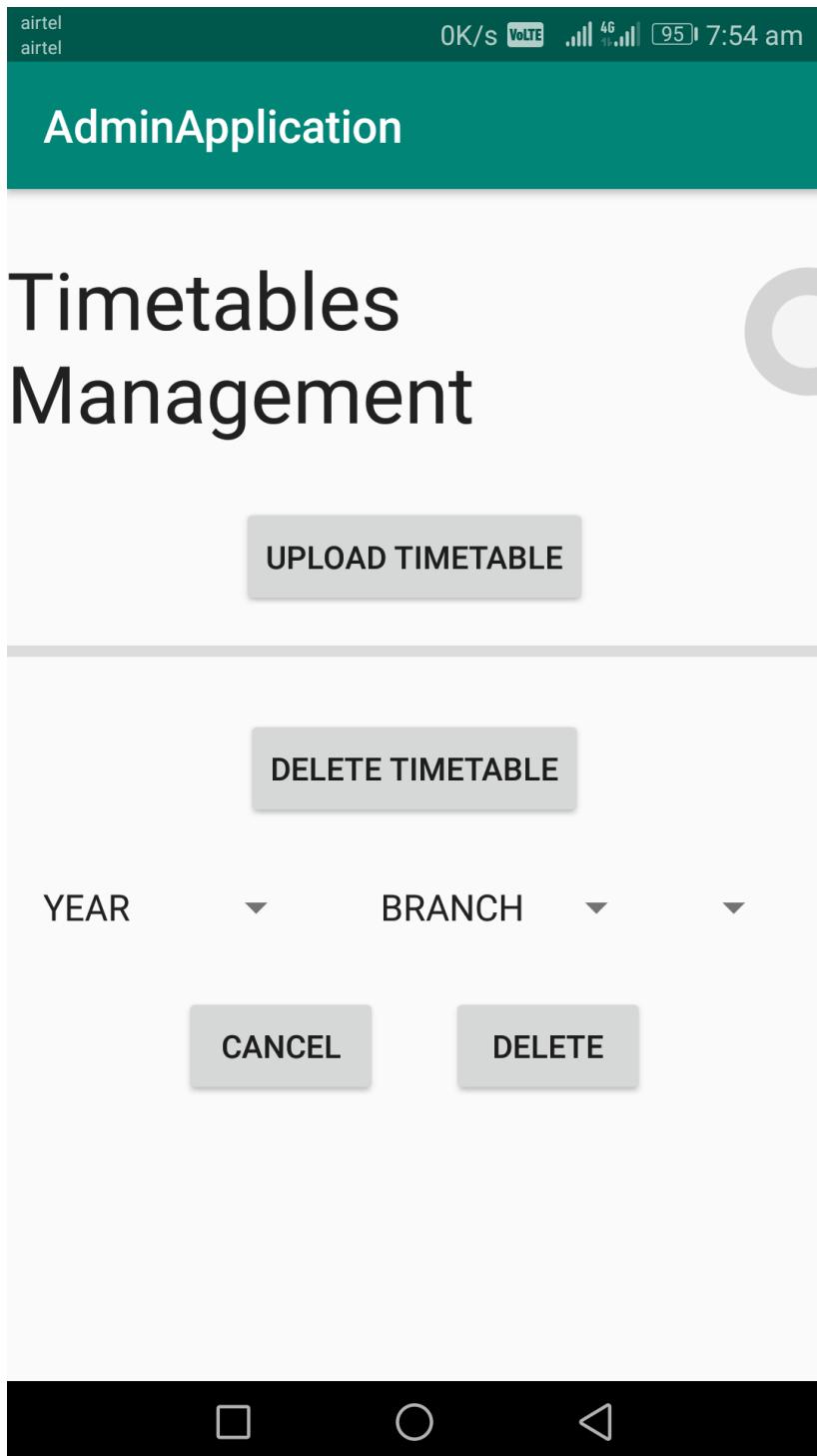
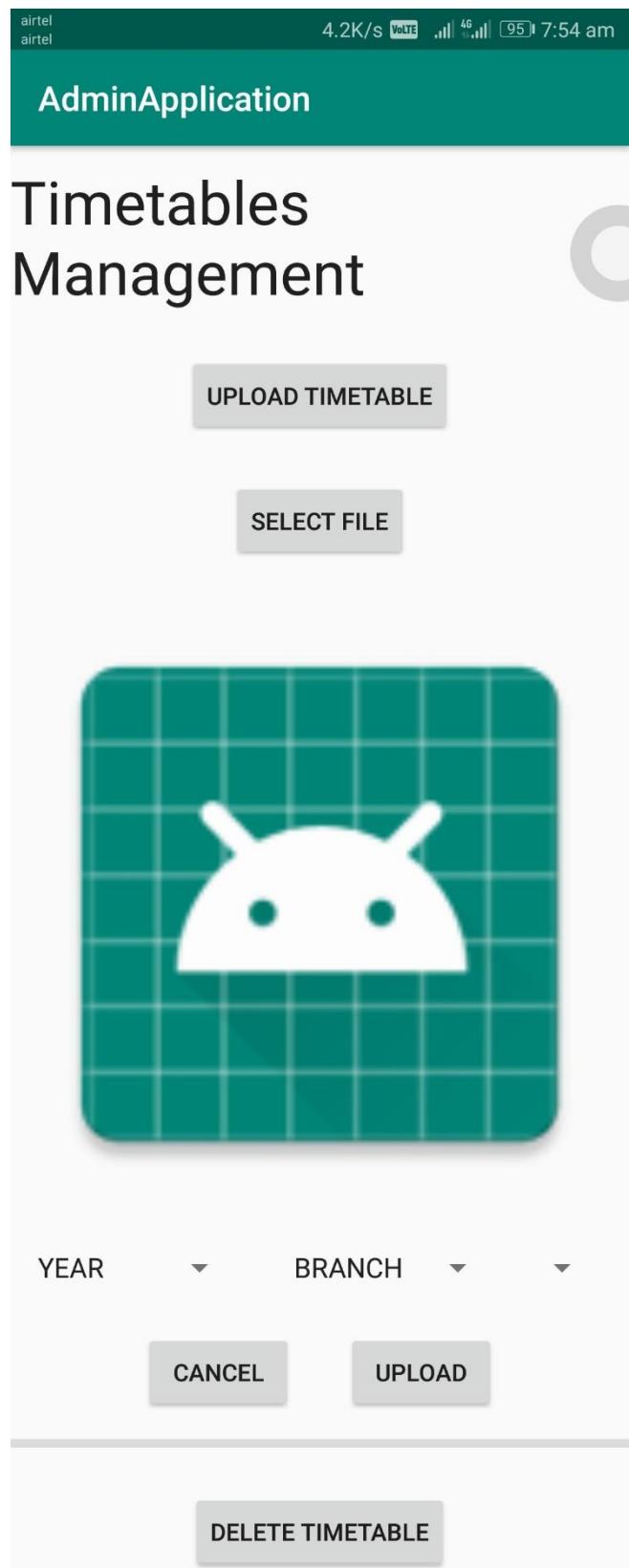
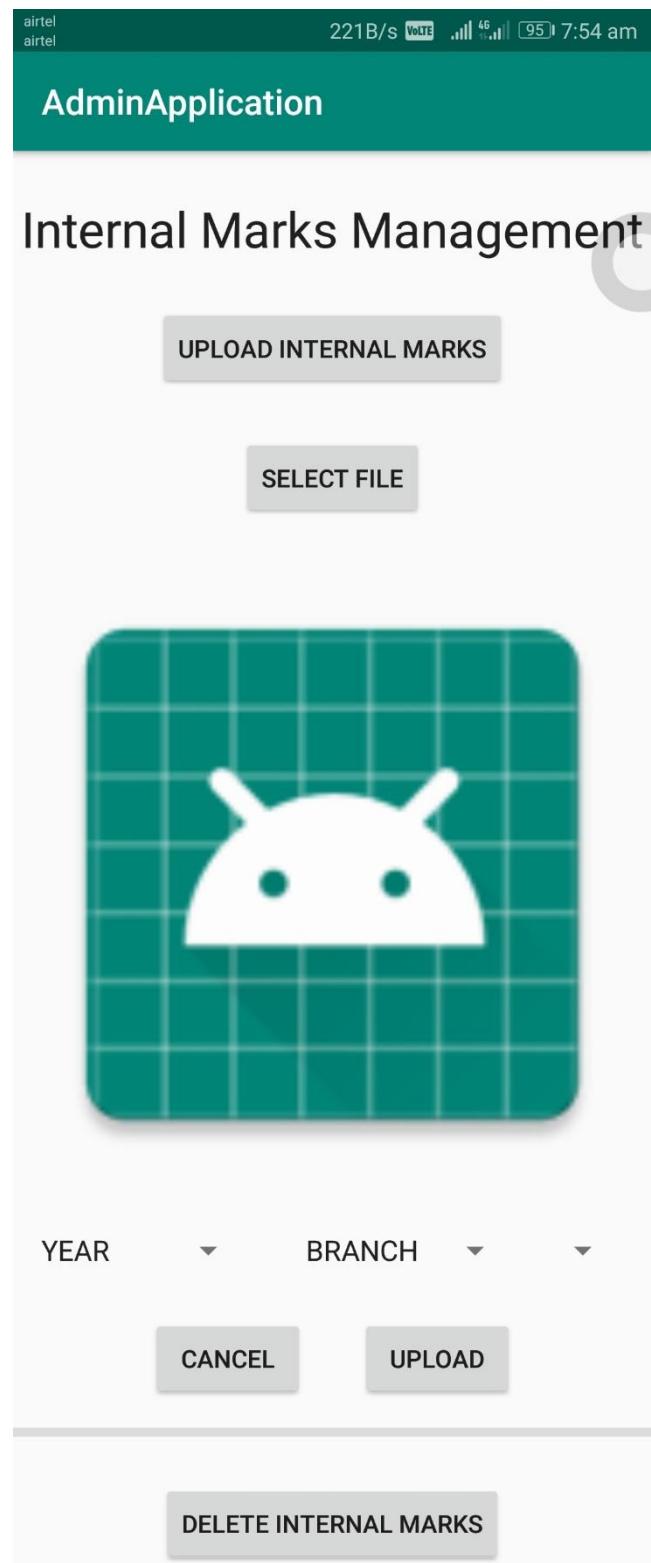


Fig.4.23: Admin Time Table Management Page.



**Fig.4.24:** Time Table Uploading Page.

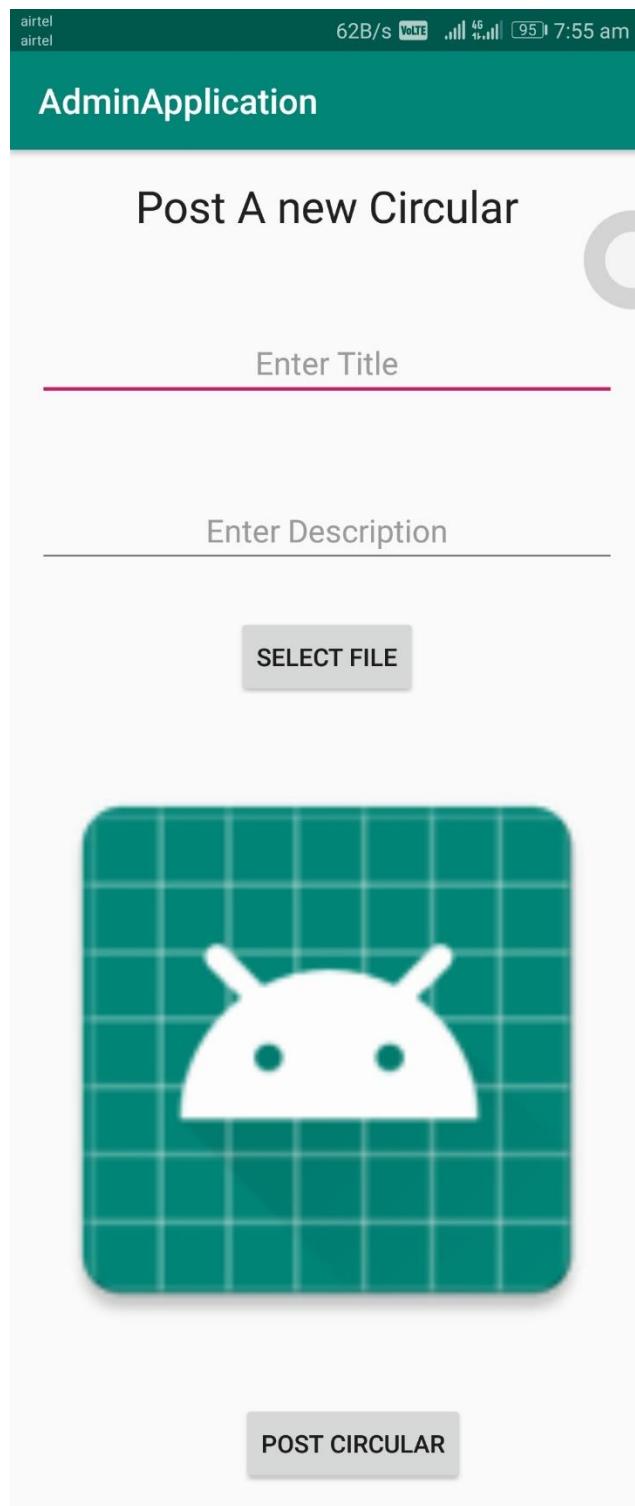
### **Internal Marks Management:**



**Fig.4.25: Internal Marks Management Page.**

### **Circulars Management page:**

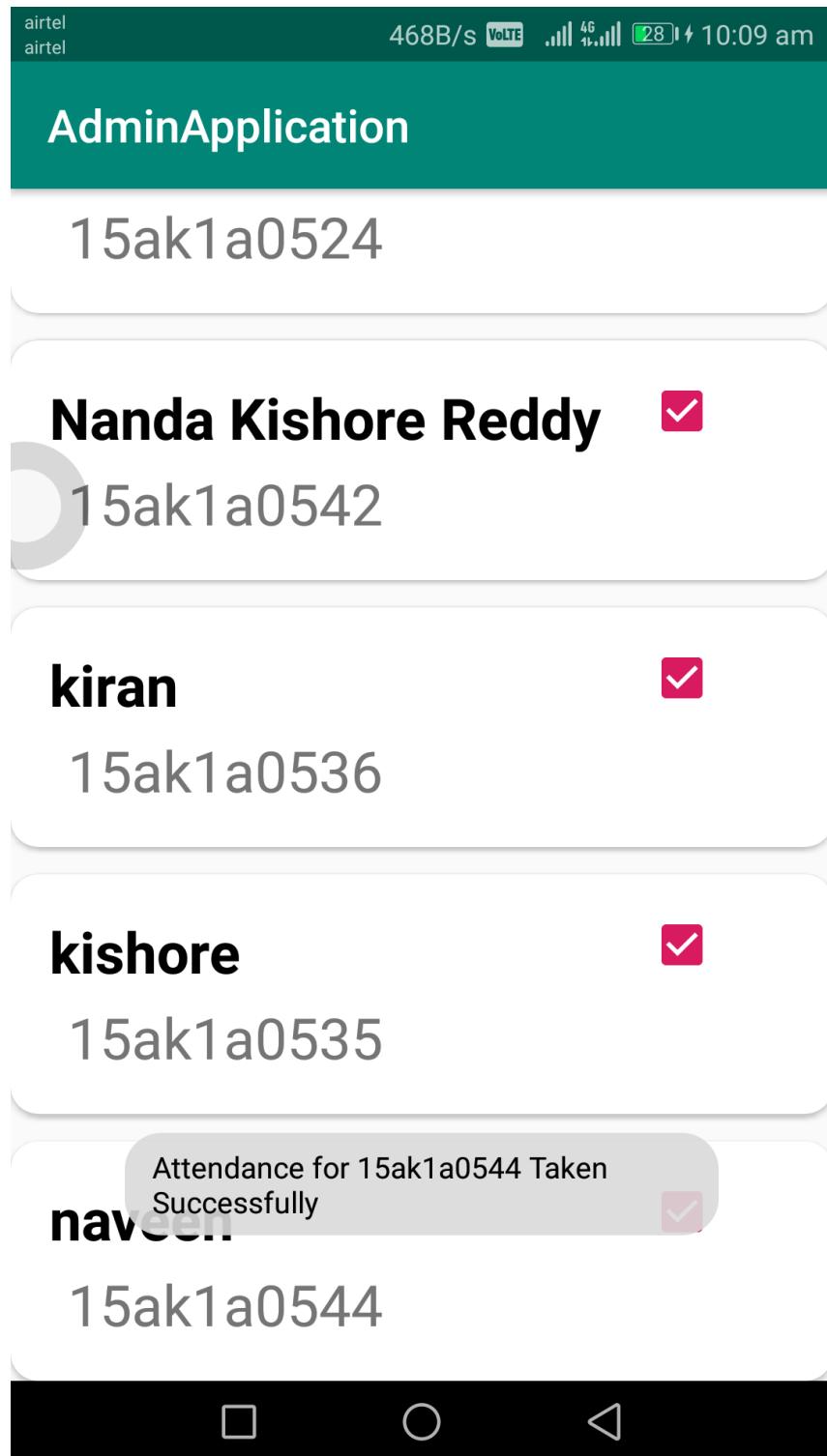
In this the Admin add a circular image along with the title, description of that circular.



**Fig.4.26: Admin Circular Management Page.**

### **Attendance Management:**

In this the faculty can take attendance directly in the application.



**Fig.4.27: Admin Attendance Page.**

## CHAPTER 5: TESTING AND VALIDATION

### 5.1 INTRODUCTION TO TESTING

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.

### 5.2 TEST CASES:

**User Application Testing**

Serial No.	Test Description	Type of Testing	Expected Result	Actual result	Pass/Fail
<b>Overall Application Testing</b>					
1.	On clicking the application, launch as designed	Black Box Testing	Opens the applications Successfully	Opens the application Successfully	pass
2.	Main menu is displayed in grid view as designed	Black Box Testing	The menu buttons are displayed in grid view Successfully	The menu buttons are displayed in grid view Successfully	pass
3.	Icons are displayed as kept	Black Box Testing	Icons are displayed Successfully	Icons are displayed Successfully	pass
4.	The Resolution of the application is fits for different mobile sizes	Black Box Testing	The resolution fits for all mobiles Successfully	The resolution fits for all	pass

				mobiles Successfully	
5.	Icon sizes are fitted to the screen	Black Box Testing	Icons are fitted to the screen Successfully	Icons are fitted to the screen Successfully	pass
<b>Testing on Circulars module</b>					
6.	Test whether the Circular button works properly	Black Box Testing	Circular button works Successfully	Circular button works Successfully	pass
7.	Test whether the Circulars are in list view	Black Box Testing	Circulars are listed in list view	Circulars are listed in list view	Pass
8.	Test whether the Circulars are ordered based on last modified	Black Box Testing	Circulars are listed based on last modified	Circulars are listed based on last modified	Pass
9.	On clicking the particular circular whether it shows the circular or not	Black Box Testing	Circulars are shown Successfully	Circulars are shown Successfully	pass
<b>Testing on Time Tables Module</b>					
10.	Test whether the Time Tables button works properly	Black Box Testing	Time Table button works Successfully	Time Table button works Successfully	Pass
11.	Drop down button in the Time Tables is working properly	Black Box Testing	Drop down button works Successfully	Drop down button works Successfully	Pass

12.	On clicking required time table whether it shows the timetable or not	Black Box Testing	On click button works successfully	On click button works successfully	Pass
13.	Whether Time Tables are showing in .jpg or not	Black Box Testing	Time Tables are showing in .jpg format Successfully	Time Tables are showing in .jpg format Successfully	Pass
14.	Whether the provided zoom in and zoom out options are working properly or not	Black Box Testing	Zoom in and zoom out options are working Successfully	Zoom in and zoom out options are working Successfully	Pass

#### **Testing on Internal Marks Module**

15.	Check whether the internal marks button is working properly	Black Box Testing	Internal marks button is working Successfully	Internal marks button is working Successfully	Pass
16.	On clicking the button the content is displayed in the next window or not	Black Box Testing	The content is displayed in the next window Successfully	The content is displayed in the next window Successfully	Pass
17.	Check whether the keyboard is appeared on the screen when user click on search bar	Black Box Testing	Keyboard appears on the screen Successfully	Keyboard appears on the screen Successfully	Pass
18.	Check whether the search button is working	Black Box Testing	Search Button is Working Successfully	Search Button is	Pass

				Working Successfully	
--	--	--	--	-------------------------	--

### **Testing on Attendance Module**

19.	Check whether the Attendance button is working properly	Black Box Testing	Attendance button is working Successfully	Attendance button is working Successfully	Pass
20.	On clicking the button the content is displayed in the next window or not	Black Box Testing	The content is displayed in the next window Successfully	The content is displayed in the next window Successfully	Pass
21.	Check whether the keyboard is appeared on the screen when user click on search bar	Black Box Testing	Keyboard appears on the screen Successfully	Keyboard appears on the screen Successfully	Pass
22.	Check whether the search button is working	Black Box Testing	Search Button is Working Successfully	Search Button is Working Successfully	Pass

### **Testing on Library Module**

23.	Check whether the Library Data button is working properly	Black Box Testing	Library Data button is working Successfully	Library Data button is working Successfully	Pass
24.	On clicking the button the content is displayed in the next window or not	Black Box Testing	The content is displayed in the next window Successfully	The content is displayed in the next window Successfully	Pass

				window Successfully	
25.	Check whether the displayed library data is sorted alphabetically	Black Box Testing	Data is sorted alphabetically	Data is sorted alphabetically	Pass
26.	The data entered is displayed as predicted	Black Box Testing	Data Displayed Successfully	Data Displayed Successfully	Pass
<b>Testing on Bus Tracking Module</b>					
27.	Check whether the Bus Tracking button is working properly	Black Box Testing	Bus Tracking button is working Successfully	Bus Tracking button is working Successfully	Pass
28.	Check whether the maps are displayed on the screen when clicks the Bus Tracking button	Black Box Testing	Maps displayed Successfully	Maps displayed Successfully	Pass
<b>Testing on Hostel Management</b>					
29.	Check whether the Hostel Management button is working properly	Black Box Testing	Hostel Management button is working Successfully	Hostel Management button is working Successfully	Pass
30.	Check whether the application form opens when clicking on new Application	Black Box Testing	New application opens Successfully	New application opens Successfully	Pass

31.	Check whether the keyboard opens on clicking the text field	Black Box Testing	Key Board opens Successfully	Key Board opens Successfully	Pass
32.	Check whether the toast message id displayed on clicking the submit button	Black Box Testing	Toast message displayed Successfully	Toast message displayed Successfully	Pass
<b>Testing on Queries</b>					
33.	Check whether the Queries button is working properly	Black Box Testing	Queries button is working Successfully	Queries button is working Successfully	Pass
34.	Check whether the keyboard is displayed on the Screen on clicking the text field	Black Box Testing	Keyboard is displayed Successfully	Keyboard is displayed Successfully	Pass
35.	Check whether the toast message is displayed on the screen when clicking submit button	Black Box Testing	Toast message is displayed Successfully	Toast message is displayed Successfully	Pass

**Table: 5.2.1 Test case for User Application**

### Admin Application Testing

Serial No.	Test Description	Type of Testing	Expected Result	Actual result	Pass/Fail
36.	On clicking the application, launch as designed	Black Box Testing	Opens Application Successfully	Opens Application Successfully	Pass

37.	Verify if User will be able to login with a valid username and valid password	Black Box Testing	Login Successful	Login Successful	Pass
38.	Verify if a User can login with a valid username and invalid password	Black Box Testing	Login Failed	Login Failed	Pass
39.	Verify if a User can login with invalid username	Black Box Testing	Login Failed	Login Failed	Pass
40.	Verify if the data in password field is visible as asterisk	Black Box Testing	Password is visible as asterisk	Password is visible as asterisk	Pass

#### **Testing on Circular management**

41.	Test whether the Circular button works properly	Black Box Testing	Button works Successfully	Button works Successfully	Pass
42.	Test whether text fields are working properly	Black Box Testing	Text fields are working properly	Text fields are working properly	Pass
43.	When click on select file, check whether it shows the file manager.	Black Box Testing	Opens File manager Successfully	Opens File manager Successfully	Pass
44.	When click on post circular, check whether the circular is updated in Firebase.	Black Box Testing	Circular is Updated Successfully	Circular is Updated Successfully	Pass

#### **Testing on Timetable management**

45.	Check whether upload timetable and delete timetable buttons are working properly	Black Box Testing	Buttons are working properly	Buttons are working properly	Pass
46.	When click on select file check whether it show file manager.	Black Box Testing	Opens File manager Successfully	Opens File manager Successfully	Pass
47.	Check whether provided dropdown menus are working properly	Black Box Testing	Drop down menu are working properly	Drop down menu are working properly	Pass
48.	Check whether upload and cancel buttons are working properly	Black Box Testing	Upload and Cancel buttons are Working properly	Upload and Cancel buttons are Working properly	Pass

#### **Testing on Internal Marks Management**

49.	Check whether upload Marks and delete Marks buttons are working properly	Black Box Testing	Buttons are working properly	Buttons are working properly	Pass
50.	When click on select file check whether it show file manager	Black Box Testing	Opens File manager Successfully	Opens File manager Successfully	Pass
51.	Check whether provided dropdown menus are working properly	Black Box Testing	Drop down menu are working properly	Drop down menu are working properly	Pass

52.	Check whether upload and cancel buttons are working properly	Black Box Testing	Upload and Cancel buttons are Working properly	Upload and Cancel buttons are Working properly	Pass
<b>Testing on Attendance management</b>					
53.	Check whether the name and roll number are displayed as designed	Black Box Testing	Details are displayed successfully	Details are displayed successfully	Pass
54.	Check whether the check boxes are working properly	Black Box Testing	Check boxes are working Properly	Check boxes are working Properly	Pass
55.	Check whether toast message is displaying on the screen	Black Box Testing	Toast message is displayed Successfully	Toast message is displayed Successfully	Pass
56.	Check whether the status of the student is changed when clicked check box	Black Box Testing	Status is changed Successfully	Status is changed Successfully	Pass

**Table: 5.2.2 Test case for Admin Application**

# **CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT**

## **6.1 CONCLUSION**

An Android based mobile application for College Management System is presented. The application offers reliability, time savings and easy control. It can be used as a base for creating and enhancing applications for viewing results, tracking attendance for college. Students and their parents will also view results, attendance and curriculum details using this application. Also students can view details, notifications anywhere and anytime. The application will greatly simplify and speed up the result preparation and management process. It includes advertisement of the college thus satisfying the marketing criteria to showcase its facilities to the HSC and diploma students. It provides high security and a system that reduces the work and resources required in traditional process. The proposed system provides the new way of computing and displaying operations with responsive and attractive user-interface.

## **6.2 FUTURE ENHANCEMENT**

By implementing this project struggle to find marks, books in the library, attendance, bus info etc., will be reduced. There is a scope in the future that social networking can also be added where students can communicate with each other. Also online class functionality can be added and online exam functionality can be added. Online fee payment like College fee, Examination fee, any fine payments can also be added.

## CHAPTER 7: REFERENCES

- [1] Martin Hellebrandt and Rudolf Mathar “Location Tracking of Mobiles in Cellular Radio Networks”, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, SEPTEMBER 1999.
- [2] M.A. Norasiah and A. Norhayati. “Intelligent student information system”. 4th International conference on telecommunication technology proceedings, Shah Alam, Malaysia, 0-7803-7773-7/03 2003 IEEE.
- [3] Sandeep Kumar, Mohammed Abdul Qadeer, Archana Gupta, “Location Based Services using Android”, IEEE- 2009.
- [4] TANG Yu-fang, ZHANG Yong-sheng, ,“Design and implementation of college student information management system based on the web services”. Natural Science Foundation of Shandong Province(Y2008G22), 978-1-4244-3930-0/09 2009 IEEE.
- [5] Bal’azs Beny’ o, B’alint S’odor , Tibor Doktor and Gergely Fordos “Student attendance monitoring at the university using NFC” IEEE on computer techonolgy- June 2010.
- [6] Zhibing Liu, Huixia Wang,Hui Zan “Design and implementation of online college management system.” 2010 International symposium on intelligence information processing and trusted computing. 978-0-7695- 4196-9/10 IEEE.
- [7] Zhi-gang YUE, You-wei JIN, “The development and design of the student management system based on the network environment”, 2010 International Conference on Multimedia Communications, 978-0-7695-4136- 5/10 2010 IEEE.
- [8] Ramesh Shrestha, Yao alihong “Design of Secure Location and Message Sharing System for Android Platform”, IEEE-2012 on computer technology.
- [9] Fei Shao, Zinan Chang, Yi Zhang, “AES Encryption Algorithm Based on the High Performance Computing of GPU”, Second International Conference on Communication Software and Networks, IEEE -ICCSN.2010.124.
- [10] Jin Mei-shan1 Qiu Chang-li 2 Li Jing 3. “The Designment of student information management system based on B/S architecture”. 978-1-4577-1415-3/12 2012 IEEE.
- [11] S.R.Bharamagoudar, Geeta R.B., S.G.Totad “Web Based Student Information Management System”, International Journal of Advanced Research in Computer and Communication Engineering -June 2013, ISSN : 2319-5940.

- [12] Penghui Li1, Yan Chen, Taoying Li, Renyuan Wang, Junxiong Sun “Implementation of Cloud Messaging System Based on GCM Service”, IEEE-2013.
- [13] Suriyani Ariffin, Ramlan Mahmood, “SMS Encryption using 3D-AES Block Cipher on Android Message Application ”, International Conference on Advanced Computer Science Applications and Technologies May-2013-IEEE 978-1-4799-275.
- [14] Yang Shulin, Hu jipieng “Research and Implementation of Web Services in Android Network Communication Framework Volley” IEEE- ICCSN.2014.20.
- [15] Suchita Tayde, Asst. Prof. Seema Siledar “File Encryption, Decryption Using AES Algorithm in Android Phone”, International Journal of Advanced Research in Computer Science and Software Engineering, May-2015, ISSN: 2277 128X.
- [16] Siddhant Gokule, Rohit Jadhav, Mruganayani Mane, Sanchita Chhajed “An Application to Enhance the Admission Process and Communication between Members of an Institute”, International Journal on Recent and Innovation Trends in Computing and Communication- ISSN: 2321-8169.
- [17] Miss. Namrata N. Shahade, Miss. Priya A. Kawade, Mr. Satish L. Thombare “Student Attendance Tracker System in Android”, INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY- ISSN: 2321-8134.

**Website Links:**

- <https://developer.android.com/docs>
- <https://firebase.google.com/docs>
- <https://github.com/googlesamples/android-CardView/#readme>
- <https://github.com/googlesamples/android-RecyclerView/#readme>
- <https://developer.android.com/studio/write/sample-code>
- <https://developer.android.com/training/basics/firstapp>
- <https://developer.android.com/guide>