Alpesh patil,  ATTENDANCE MONITORING APPLICATION

**ATTENDANCE MONITORING APPLICATION**



**Roll No/Exam Seat No.**:229014



**“Education through self help is our motto”-KARMAVEER**

**Rayat Shikshan Sanstha’s**

**Mahatma Phule Arts, Science &**

**Commerce College, Panvel. Dist: Raigad**

Re-accredited by NAAC at ‘A’ Grade with CGPA of 3.18

“Best College Award” by University of Mumbai

Recipient of Grant under DBT Star College Scheme

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that Mr. **Alpesh Dhananjay Patil** has satisfactorily carried out required practical work for the subject **Software Project Management,** prescribed by the **University of Mumbai** **for the T. Y. B. Sc course in Information Technology Semester–V** and this journal represent his/her bonafide work in the year **2022-2023.**

**Signature of Teacher In-Charge**

**Date:**

**Signature of Examiner**

**Date: Head**

**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PRN **No.: 2020016401584591** Roll no**: 229014**

1. **Name of the Student**

Alpesh Dhananjay Patil

1. **Title of the Project**

ATTENDANCE MONITORING APPLICATION

1. **Name of the Guide**

Miss. Pooja Thakur.

1. Teaching experience of the Guide 5 Month
2. Is this your first submission?  **Yes** No

Signature of the Student Signature of the Guide

Date: ………………… Date: …………………….

Signature of the Coordinator

Date: …………

**ABSTRACT**

Attendance management is important to every single organization; it can decide whether or not an organization such as educational institutions, public or private sectors will be successful in the future. Organizations will have to keep a track of people within the organization such as employees and students to maximize their performance. Managing student attendance during lecture periods has become a difficult challenge. The ability to compute the attendance percentage becomes a major task as manual computation produces errors, and wastes a lot of time. For the stated reason, an efficient Web-based application for attendance management system is designed to track student's activity in the class. This application takes attendance electronically and the records of the attendance are storing in a database. The system design using the Model, View, and Controller (MVC) architecture, and implemented using the power of Laravel Framework. JavaScript is adding to the application to improve the use of the system.

MySQL used for the Application Database. The system designed in a way that can differentiate the hours of theoretical and practical lessons since the rate of them is different for calculating the percentages of the students' absence. Insertions, deletions, and changes of data in the system can do straightforward via the designed GUI without interacting with the tables. Different presentation of information is obtainable from the system. The test case of the system exposed that the system is working enormously and is ready to use to manage to attend students for any department of the University. INTRODUCTION Due to student's interest in classrooms, and whose is the largest union in the study environment of university or institution, so recording absence at a department having a large number of students in a classroom is a difficult task and time-consuming. Moreover, the process takes much time, and many efforts are spent by the staff of the department to complete the attendance rates for each student. So in many institutions and academic organizations, attendance is a very important criterion which is used for various purposes. These purposes include record keeping, assessment of students, and promotion of optimal and consistent attendance in class. As long as in many developing countries, a minimum percentage of class attendance is required in most institutions and this policy has not been adhered to, because of the various challenges the present method of taking attendance presents. The process of recording attendances for students was in the form of hardcopy papers and the system was manually done.

**ACKNOWLEDGEMENT**

Is a great pleasure that we present our first venture in real life application of computing in the form of project work? It has proved very helpful with respect to program developing techniques. We acknowledge our sincere gratitude to all those who helped us to make this project a success.

I am happy to present this project work on **ATTENDANCE MONITORING SYSTEM.** The satisfactory completion of this project is not merely due to my own efforts but also due to the Valuable guidance rendered by my guide and others and also the requisite infrastructure and facilities made available by my college.

First, we extended our sincere thanks to the Principal **Dr. GANESH A. THAKUR** and our guide **MS**. **POOJA THAKUR** and Head of Department **MS. NANDINI GAIKWAD** for their co-operation and guidance throughput the course and briefed us from time to time a normal support and overcoming our anxiety to perform in this project.

We express our gratitude to all the faculty members of IT Department who turned our knowledge in the field of IT in the real world. We also give thanks to those who helped us directly or indirectly during our project. Last but not least we would like to thank my friends for their constant moral support.

**DECLARATION**

I hereby declare that the project entitled, “Attendance Monitoring System” done at my college is done, has not been any case duplicated to any other university for the ward of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**Patil Alpesh D.**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **TITLE** | **Page No** |
| **1** | **Chapter 1 :- Introduction** | 7-10 |
| 1.1 Background |
| 1.2 Objective |
| 1.3 Purpose and Scope |
| 1.3.1 Purpose |
| 1.3.2 Scope |
|  |  |  |
| **2** | **Chapter 2 :- System Analysis** | 11-17 |
| 2.1 Existing System |
| 2.2 Proposed System |
| 2.3 Requirement Analysis |
| 2.3.1 Hardware Requirements |
| 2.3.2 Software Requirements |
| 2.4 Justification of selection of Technology |
|  | | |
| **3** | **Chapter 3 :- System Design** | 20-35 |
| 3.1 Module Division |
| 3.2 ER diagram |
| 3.3 Class Diagram |
| 3.4 Use case Diagram |
| 3.5 Activity Diagram |
| 3.6 Gantt Chart |
|  | |
| **4** | **Chapter 4 :- Implementation and Testing** | 36-49 |
| 4.1 Code |
| 4.2 Testing Approach |
| 4.2.1 Unit Testing |
| 4.2.2 Integrating System |
|  |  |
| **5** | **Chapter 5 :- Result and Discussion** | 50-55 |
| 5.1 Screenshots |
|  |  |
| **6** | **Chapter 6 :- Conclusion and Future Work** | 56-58 |
|  | | |
| **7** | **Chapter 7 :- References** | 59-60 |

**CHAPTER 1**

**Introduction**

Attendance Monitoring System is software developed for daily student attendance in schools, colleges and institutes. It facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class.

[**Attendance Monitoring System**](https://www.lawinsider.com/dictionary/attendance-monitoring-system) means a computerized software along with special devices to capture attendance data such as arrival time, duration of absence from Section, leave at credit and ailment and monthly aggregate of hours of duty and absence of employee.

In the proposed system the user records will be stored digitally thereby overcoming the limitations of existing system. Moreover, the lighting errors that may possibly occur in manual system will now be minimized.

**1.1 BACKGROUND**

Tracking, storing and managing attendance is a regular task in schools, colleges, enterprises and other establishments. In an educational institution, the teachers need to track their student’s attendance and similarly, in an enterprise, the employers have to track their employees’ attendance. Managing attendance by calling out names or marking it using a pen on a register is a tedious task and an outdated process. So, it is essential for an organization to opt for a smart attendance management system.

**1.2 Objectives**

**“Attendance Monitoring System”** is software developed for maintaining the attendance of the student on the daily basis in the collage. Here the staffs, who are handling the subjects, will be responsible to mark the attendance of the students. Each staff will be given with a separate username and password based on the subject they handle. An accurate report based on the student attendance is generated here. This system will also help in evaluating attendance eligibility criteria of a student. Report of the student’s attendance on weekly and monthly basis is generated.

**PURPOSE, SCOPE AND APPLICABILITY**

**1.3.1 Purpose:**

Attendance management is an important task within every school. The procedure helps to keep a track of students, teaching staff and non-teaching staff of a school. The purpose of the attendance management system is to help the administrators keep a track of students and staff. Automated attendance management software can help administrators save time and money. Such a system also reduces staff workload and increases efficiency. The system helps to manage attendance by analysing student absences, looking for solutions, real-time tracking, eliminating duplicate data entry etc. You will get a graphical representation or other forms of representation of everything related to student/staff presence and absence.

**1.3.2 Scope:**

Attendance management system **keeps track of daily attendance, working hours, breaks, login, and logout time**. It prevents staff's time theft. An attendance management system integrates all attendance devices such as smart cards, biometric, and facial recognition devices in real-time.

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular institute. But later on the project can be modified to operate it online.

**1.3.3 Applicability:**

Monitoring and managing attendance manually can be a time-consuming, laborious, and expensive affair. It takes time to process paper sheets and time cards, create schedules, [authorize leave](https://www.greythr.com/leave-management-software/) and overtime, and create payroll manually. Free up precious administration time with an automated system that does it all for you, from keeping track of employee hours to automatically importing information into your [payroll](https://www.greythr.com/payroll-software/) system. The time and effort saved combined with data accuracy helps in optimizing the use of resources which lead to increased productivity and improves profits.

**CHAPTER 2**

**SYSTEM ANALYSIS**

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skilful wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation. The various tasks in the system analysis include the following. Understanding application.

* Planning.
* Scheduling.
* Developing candidate solution. Performing trade studies.
* Performing cost benefit analysis.
* Recommending alternative solutions. Supervising,
* Installing and maintaining the system.

**2.1 Existing System:**

The existing system is manual where the users have to edit their photo details. Users edit the photo by using old app. Storing the data manually may cause loss of data if the phone are lost or damaged. Sometimes phone-based work is difficult to manage and stored. Because of storing the photo data in phone may cause effects errors.

**Disadvantages**:

* In the existing system Users maintain their photo related data using phone.
* Storing data manually may cause loss of data.
* To fill the photo related data User, have to go into the gallery.
* It calculates the bill manually so it may cause effect errors.

**2.2 Proposed system:**

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the student’s attendance. The system provides with the best user interface. The efficient reports can be generated by using this proposed system.

**Advantages of Proposed System**:-

* It is trouble-free to use. It is a relatively fast approach to enter attendance is highly reliable, approximate result from user Best user Interface and efficient reports.

It is trouble-free to use. It is a relatively fast approach to enter attendance is

highly reliable, approximate result from user Best user Interface and efficient

* It will help to reduce the phone-based work.
* All information will be stored in an efficient manner in a database.
* Low cost will be required to develop this system.
* It is useful to all the user.
* It would be easily accessible by the entire User.
* Filter error that may be occurring in manual system will be minimized.

**Planning and Scheduling:**

Gantt chart in Project Management is basically a visual view of the tasks that are scheduled over time. They are needed for planning of all projects of all sizes and they are quite a useful way of displaying which work is scheduled to be executed on a specific day. Gantt charts help you in viewing the start and end dates of the project in a single simple view.

### Benefits of Gantt chart in Project Management

Many people in today’s world are quite used to creating lists of tasks in Excel or different spreadsheet tools. They might also have created a simple Gantt Chart. It works fine whenever you are creating a list of things for anyone to view. However, when you add more people to that activity, at that time it becomes very much easier to create the Gantt Chart in an online manner.

The benefits of the Gantt Chart are many. We have listed the most important benefits down below:

* Planning & Scheduling of Projects
* The Planning & Scheduling of tasks
* Planning & Scheduling of tasks across multiple projects
* Viewing tasks Over Time
* Planning in Sprints
* Team Collaboration
* Scheduling Team’s work
* Determining Planned Timelines Vs Actual Timelines on Project

**2.3 Requirements Analysis: -**

**2.3.1 Hardware Requirement:**

|  |  |
| --- | --- |
| Processor: | Intel |
| RAM: | 8 GB or above cache memory |
| Hard Disk: | 60 GB or above |
| Input Device: | Keyboard, Mouse |
| Output Device: | Monitor |

**2.3.2 Software Requirement:**

|  |  |
| --- | --- |
| Operating System: | Android 3.0 and above |
| Programming Language: | Java, Xml |
| Front-End: | Android studio |
| Back-End: | MySql |
| Web Server: | IIS SERVER |
| Libraries: | Android Libraries |
| Integrated Development: | Android Studio platform |

**2.4 Justification of Technology:**

**Front End:**

### 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.

Features provide by android studio are:

* + - Flexible gradle – based system
    - Fast and feature rich emulator
    - Instant run to push changes to your running app without building new apk.
    - Extensive testing tools and frameworks.
    - C++ and NDK supports
    - Built in support for Google cloud platform, making it easy to integrate Google cloud messaging and app engine.

All the build files of projects are visible at top level under gradle script and each app module contains following folders:

* Manifest : Contains the Android manifest – xml file
* Java : Contains the java source code files including Junit test code.
* Res : Contains all non-code resources such as XML layouts, UJ strings and bitmap images.

User Interface

* Toolbar The toolbar lets you carry out a wide ranges of actions including running app and launching Android tools.
* The Navigation Bar helps you to through project and open files for editing. It provides a more compact view of structure visible in project window.
* Editor Window is where you create and modify code. Eg. When viewing a layout file, the editor displays layout editor.
* Tool window runs around outside of IDE window and contains buttons that allow you to expand or collapse individual tool windows
* Tool windows give you access to specific task like project management, search, version control and more.

**Back End:**

**Back end – Firebase Database**

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firebase platform has 19 products, which are used by more than 1.5 million apps.

**Firebase Realtime Database**

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in Realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js.The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company's server-side-enforced security rules.

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real-time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great real-time experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then structure it accordingly.

Realtime - Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.

Offline - Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is re-established, the client device receives any changes it missed, synchronizing it with the current server state.

Accessible from Client Devices - The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there’s no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

Scale across multiple databases - With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

**CHAPTER 3**

**SYSTEM DESIGN**

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architectsyste

I have used ER Diagram, Activity Diagram, Use case Diagram, Data Flow Diagram class diagram, sequence diagram, circuit diagram for design system.

**3.1 Module Division:**

The “Photo Filter App” contains 3 modules to work on it.

**Administrator:** An Administrator will login the form. They could be viewing the Use. Admin can provide the password to the teaching and non-teaching User. It will add or delete the stream and department as well as decide the exam rate. Administrator will view and approve the exam bill. They will also change the password and logout the system.

* Login
* Update the Filter
* Add/delete the filter and user
* View the User information

**User:** A User will be register and login on that system. User can apply the filters on photo and related information. User can change the password. After doing all the editing. User will able to logout the system.

* Login
* Register,
* Filter the photo
* Logout

**Data Dictionary:**

1. **USER REGISTRATION DATA:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| Input Type | Varchar(10) | Enter user Input Type(not null) |
| Phone Number | Int(10)(Primary key) | Enter Phone Number(not null) |
| User\_Id | Varchar(10) | Enter User Id(not null) |
| Name | Varchar(10) | Enter name(not null) |
| Password | Varchar(10) | Enter password(not null) |
| Conform Password | Varchar(10) | Conform the password(not null) |
| Gender | Varchar(5) | Enter gender(not null) |
| Address | Varchar(20) | Enter the address(not null) |
| Email Id | Varchar(15) | Enter the email id (not null) |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| Id | Varchar(10)(primary key) | Enter Id(not null) |
| User\_Id | Varchar(10) | Enter User Id(not null) |
| Photo\_url | Varchar(100) | Enter photo(not null) |

**3.2 E-R Diagrams:**

An Entity Relationship (ER) diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER diagrams are most often used to design or debug relational databases in the fields of software engineering business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

**ERD NOTATONS:**

Peter Chen developed ERDs in 1976. Since then Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

**ERD entity symbols:**

Entities are objects or concepts that represent important data. Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams.

* Strong Entity:

|  |
| --- |
|  |

These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity

* Weak Entity:

Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity.

* Associative Entity:

Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances.

**ERD Relationship Symbols:**

Within entity-relationship diagrams, relationships are used to document the interaction between two entities. Relationships are usually verbs such as assign, associate, or track and provide useful information that could not be discerned with just the entity types.

* Relationship

Relationships are associations between or among entities.

* Weak Relationship

Weak Relationships are connections between a weak entity and its owner.

**ERD Attribute Symbols:**

ERD attributes are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.

* Attribute

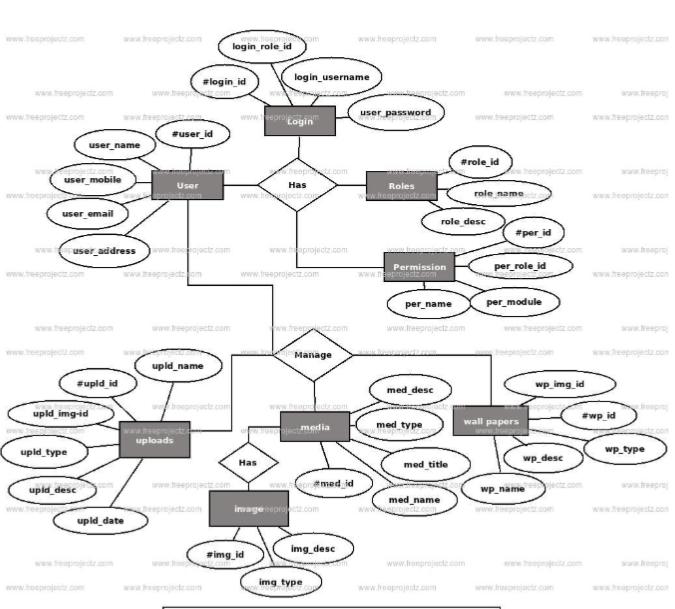
Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship.

* Multivalued Attribute

Multivalued attributes are those that are can take on more than one value.

* Derived Attribute

Derived attributes are attributes whose value can be calculated from related attribute values.



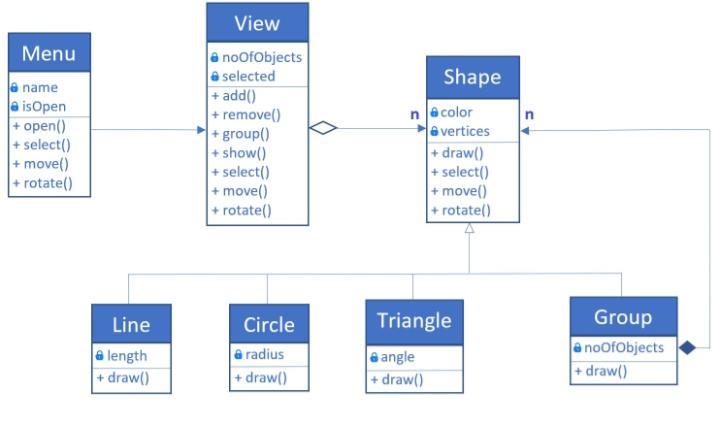
**3.3 Class Diagram:**

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a system by modelling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear. This guide will show you how to understand, plan, and create your own class diagrams.

**Basic components of a class diagram**

The standard class diagram is composed of three sections:

* **Upper section:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
* **Middle section:** Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
* **Bottom section:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.



**3.4 Activity Diagram:**

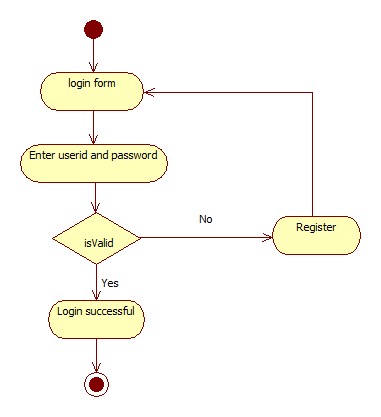
**INTRODUCTION:**

The focus of activity modelling is the sequence and conditions for coordinating lower level behaviours, rather than which classifiers own those behaviours. These are commonly called control flow and object flow models. The behaviours coordinated by these models can be initiated because other behaviours finish executing, because objects and data become available, or because events occur external to the flow.

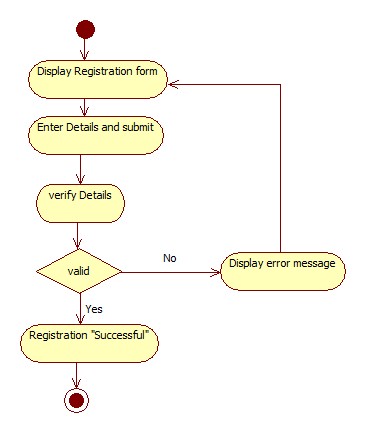
**SYMBOLS AND DESCRIPTION:**

|  |  |
| --- | --- |
| **Symbols** | **Description** |
| Initial State | Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point. |
| Action State | An action state represents the non-interruptible action of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners. |
| Transition | Transition, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line. |
| Decision | Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container. |
| Final State | Marks the end state of an activity and represents the completion of all flows of a process. |

1. **For Login form:**

****

1. **For Registration form:**

****

**3.5 Use case Diagram:**

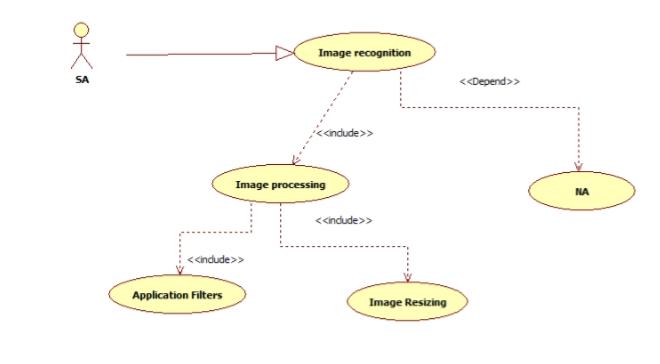
**Introduction:**

USE CASE DIAGRAM is an expression of relations between the use cases in a specific system or object and the external actors. Use Case expresses the functions of the system and how the system functions interact with the external actors.

**Symbols and Description:**

|  |  |
| --- | --- |
| **Symbols** | **Description** |
|  | **Actor** specifies a role played by a user or any other system that interacts with the subject. |
|  | **Use case** is a list of steps, typically defining interactions between an actor and a system, to achieve a goal. |
| System | Draw your **system's boundaries** using a rectangle that contains use cases. Place actors outside the system's boundaries. |
|  | An **association** is the relationship between an actor and a business use case. It indicates that an actor can use a certain functionality of the business system. |

Image Filter for use case diagram



**3.5.1 Sequence Diagram:**

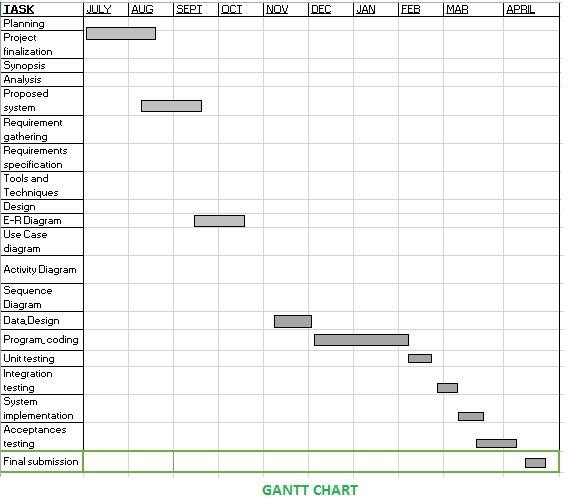
**INTRODUCTION:**

Sequence Diagram expresses the interactions of instances. It is a direct expression of the Interaction Instance Set, which is a set of the stimuli exchanged between the instances within a Collaboration Instance Set. While Sequence Role Diagram is a Classifier Role oriented expression, Sequence Diagram is an Instance-oriented expression.

**SYMBOLS AND DESCRIPTION:**

|  |  |
| --- | --- |
| **Symbols** | **Description** |
| Object1  Object | Objects are model elements that represent instances of a class or of classes. |
| Stimulus | Message is an element that defines a specific kind of  Communication between instances in an interaction. |

**3.6 GANTT CHART:**



**CHAPTER 4**

**IMPLIMENTIONS AND TESTING**

**4.1 Code**

**Login page**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:background="@drawable/clg"  
 tools:context=".MainActivity" >  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginLeft="82dp"  
 android:layout\_marginTop="43dp"  
 android:text="Login here.."  
 android:textAppearance="?android:attr/textAppearanceLarge" />  
  
 <EditText  
 android:id="@+id/editTextpassword"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/textView3"  
 android:layout\_below="@+id/textView3"  
 android:layout\_marginTop="15dp"  
 android:background="@drawable/roundedtextview"  
 android:ems="10"  
 android:inputType="textPassword" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/editTextusername"  
 android:layout\_below="@+id/editTextusername"  
 android:layout\_marginTop="14dp"  
 android:text="Password"  
 android:textAppearance="?android:attr/textAppearanceMedium" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/editTextusername"  
 android:layout\_below="@+id/textView1"  
 android:layout\_marginTop="110dp"  
 android:text="Username"  
 android:textAppearance="?android:attr/textAppearanceMedium" />  
  
 <EditText  
 android:id="@+id/editTextusername"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/textView2"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="20dp"  
 android:background="@drawable/roundedtextview"  
 android:ems="10" >  
  
 <requestFocus />  
 </EditText>  
  
 <Spinner  
 android:id="@+id/spinnerloginas"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBottom="@+id/textView2"  
 android:layout\_alignLeft="@+id/textView2"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginBottom="32dp" />  
  
 <Button  
 android:id="@+id/buttonlogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignRight="@+id/spinnerloginas"  
 android:layout\_marginBottom="34dp"  
 android:text="Login" />  
  
</RelativeLayout>

**Add items code: -**

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical" >  
  
 <ListView  
 android:id="@+id/listview"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="403dp" >  
 </ListView>  
   
</LinearLayout>

**View Students Code: -**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:background="@drawable/clg"  
 tools:context=".MainActivity" >  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="29dp"  
 android:text="Select branch"  
 android:textAppearance="?android:attr/textAppearanceSmall" />  
  
 <Spinner  
 android:id="@+id/spinnerbranchView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 android:layout\_below="@+id/textView1"  
 android:layout\_marginTop="14dp" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/textView1"  
 android:layout\_below="@+id/spinnerbranchView"  
 android:text="Select year"  
 android:textAppearance="?android:attr/textAppearanceSmall" />  
  
 <Spinner  
 android:id="@+id/spinneryearView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/textView2"  
 android:layout\_centerHorizontal="true" />  
  
 <Button  
 android:id="@+id/submitButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/spinneryearView"  
 android:layout\_marginTop="80dp"  
 android:background="@drawable/roundedbutton"  
 android:text="Submit" />  
  
</RelativeLayout>

**Menu of list modules: -**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/clg"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context=".MainActivity" >  
  
 <Button  
 android:id="@+id/buttonViewstudent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/buttonaddstudent"  
 android:background="@drawable/roundedbutton"  
 android:layout\_marginTop="18dp"  
 android:text="View student" />  
  
 <Button  
 android:id="@+id/buttonlogout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/buttonviewfaculty"  
 android:layout\_alignParentBottom="true"  
 android:background="@drawable/roundedbutton"  
 android:text="Logout" />  
  
 <Button  
 android:id="@+id/buttonviewfaculty"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_above="@+id/buttonlogout"  
 android:layout\_alignLeft="@+id/buttonaddfaculty"  
 android:layout\_marginBottom="16dp"  
 android:background="@drawable/roundedbutton"  
 android:text="View faculty" />  
  
 <Button  
 android:id="@+id/buttonaddfaculty"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_above="@+id/buttonviewfaculty"  
 android:layout\_alignLeft="@+id/buttonViewstudent"  
 android:layout\_marginBottom="14dp"  
 android:background="@drawable/roundedbutton"  
 android:text="Add faculty" />  
  
 <Button  
 android:id="@+id/buttonaddstudent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/buttonViewstudent"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="22dp"  
 android:background="@drawable/roundedbutton"  
 android:text="Add student" />  
  
 <Button  
 android:id="@+id/attendancePerStudentButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/buttonViewstudent"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="29dp"  
 android:background="@drawable/roundedbutton"  
 android:text="Attendance Per Student" />  
  
</RelativeLayout>

**Test Layout: -**

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical" >  
  
 <RadioGroup  
 android:id="@+id/radioGroup"  
 android:layout\_width="250dp"  
 android:layout\_height="120dp"  
 android:weightSum="1" >  
  
 <RadioButton  
 android:id="@+id/PresentradioButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:focusable="false"  
 android:focusableInTouchMode="false"  
 android:text="P" />  
  
 <RadioButton  
 android:id="@+id/AbsentradioButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:focusable="false"  
 android:focusableInTouchMode="false"  
 android:text="A" />  
  
 <Button  
 android:id="@+id/attendanceSubmitButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/roundedbutton"  
 android:text="Submit" />  
  
 </RadioGroup>  
</LinearLayout>

**Test main Layout**: -

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context=".TestActivity" >  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/hello\_world" />  
  
 <Button  
 android:id="@+id/button1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/textView1"  
 android:layout\_marginTop="129dp"  
 android:layout\_toRightOf="@+id/textView1"  
 android:text="Button" />  
  
</RelativeLayout>

**4.2 TESTING APPROACH**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding. Here are some objectives of testing given below:

* Testing is a process of executing a program with the intent of finding an error.
* good test case is the one that has high portability of finding an as-yet undiscovered error.
* A successful test is one that uncovers an as-yet discovered error.

The main objectives here will be to design test cases to uncovered different classes of errors and to do so with minimum amount of time and efforts. If testing is conducted successfully, it will be uncovered error in the software. Another advantage is that it demonstrates that software functions appear to be working according to the specification and performance requirement have been met. But testing cannot show the absence of defects it can show only that software error are present.

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should

Systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts.

Two classes of inputs are provided to test the process.

* An application configuration that includes an application requirement specification, a design specification and source code.
* An application configuration that includes a test plan and procedure, any testing too land test cases and their expected results.

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.

Software testing has different goals and objectives. The major objectives of Software testing are as follows:

* Finding defects which may get created by the programmer while developing the software.
* Gaining confidence in and providing information about the level of quality.
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* •To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product.

Testing can never completely identify all the defects within software. Every software product as a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders, Software testing is the process of attempting to make this assessment**.**

**Testing Methods**

1. Static Testing
2. Dynamic Testing

**Static Testing**: It is also known as Verification in Software Testing. Verification is a static method of checking documents and files. Verification is the process, to ensure that whether we are building the product right i.e., to verify the requirements which we have and to verify whether we are developing the product accordingly or not. Activities involved here are Inspections, Reviews, Walkthroughs

**Dynamic Testing:** It is also known as Validation in Software Testing. Validation is a dynamic process of testing the real product. Validation is the process, whether we are building the right product i.e., to validate the product which we have developed is right or not.

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

**Testing Approaches:**

There are three types of software testing approaches.

1. White Box Testing
2. Black Box Testing
3. Grey Box Testing

**White Box Testing:** It is also called as Glass Box, Clear Box, Structural Testing. White Box Testing is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.

**Black Box Testing:** It is also called as Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.

**Grey Box Testing:** Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

**4.3 Unit Testing:**

Unit testing is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

**4.4 Integration Testing:**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing and delivers as its output the integrated system ready for system testing.

**CHAPTER 5**

**RESULTS AND DISSCUSSION**

We have made a remote controlled car by using Bluetooth technology. In the project, we have given power supply and commands through Bluetooth app as input. We have taken output from digital input/output pins of Arduino uno. There are basically 14 digital I/O pins in which we have utilized pin number 9, 10, 11 and 12 as output.

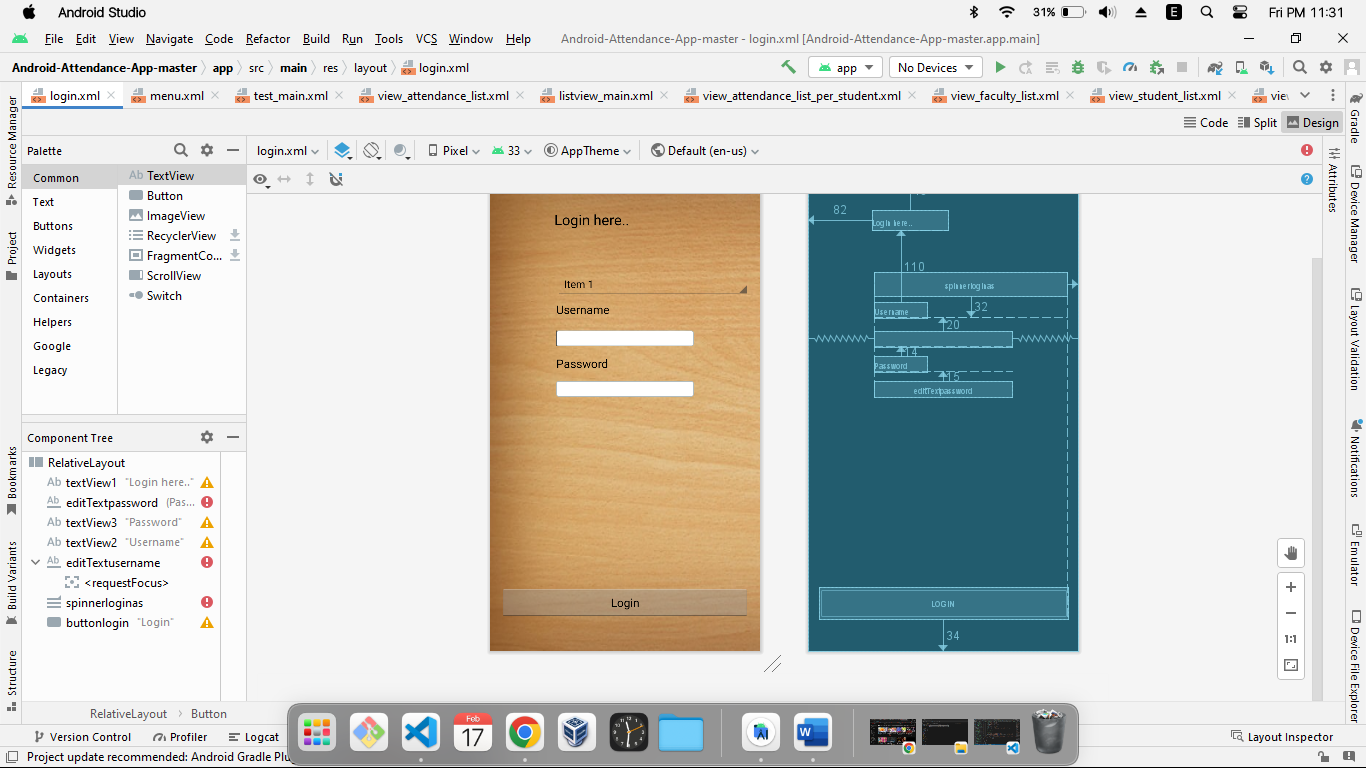
We are controlling the car from Bluetooth. Controlling the car means moving the car in all four direction that is forward, backward, left and right. We have to capital letters as an input. We have given ‘F’ for moving the car in forward direction, ‘B’ for the car in backward direction, ‘R’ for shifting the car in rightward direction and ‘L’ for shifting the car in leftward direction.

Robot is controlled through Bluetooth commands given by the user who is operating the project.

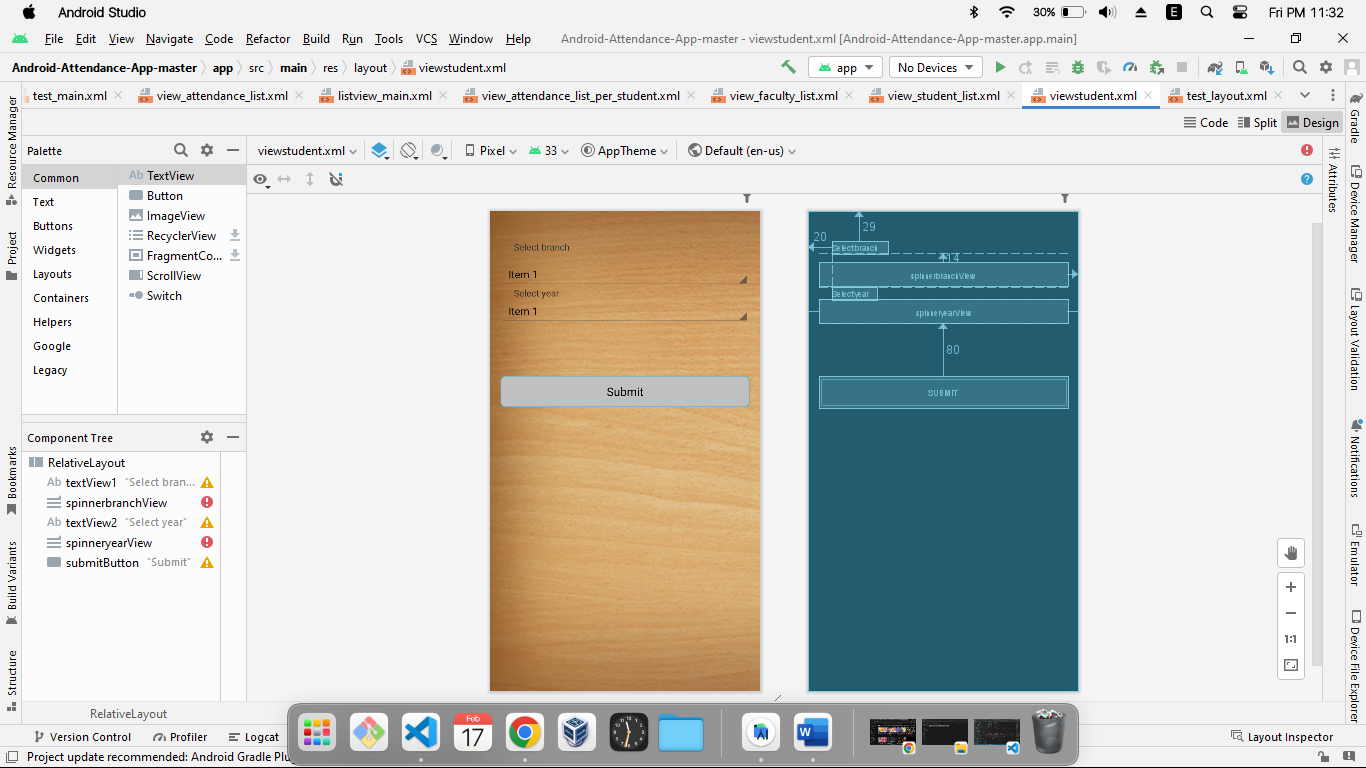
These Bluetooth command needs to be given through an android app which is installed on the users android mobile.

**5.1 Screen Shots**

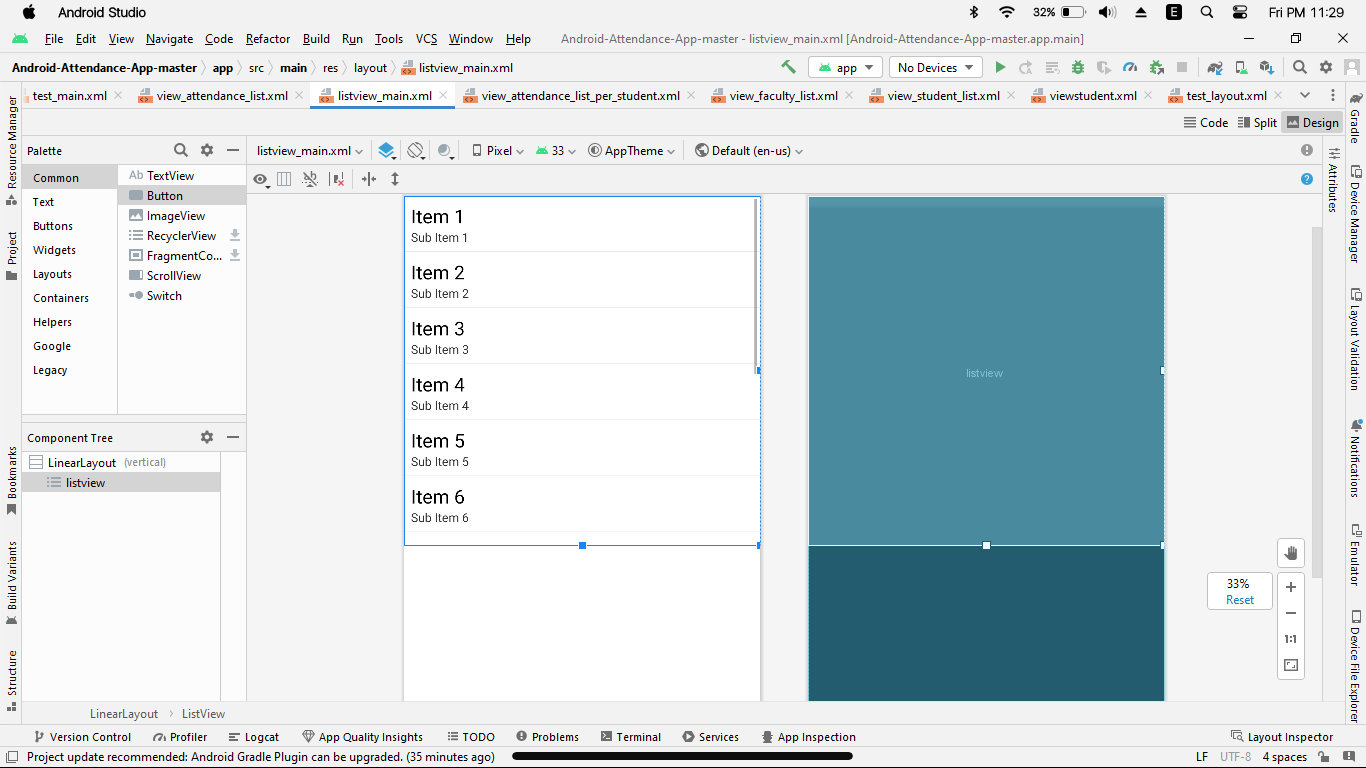
**Login Page: -**



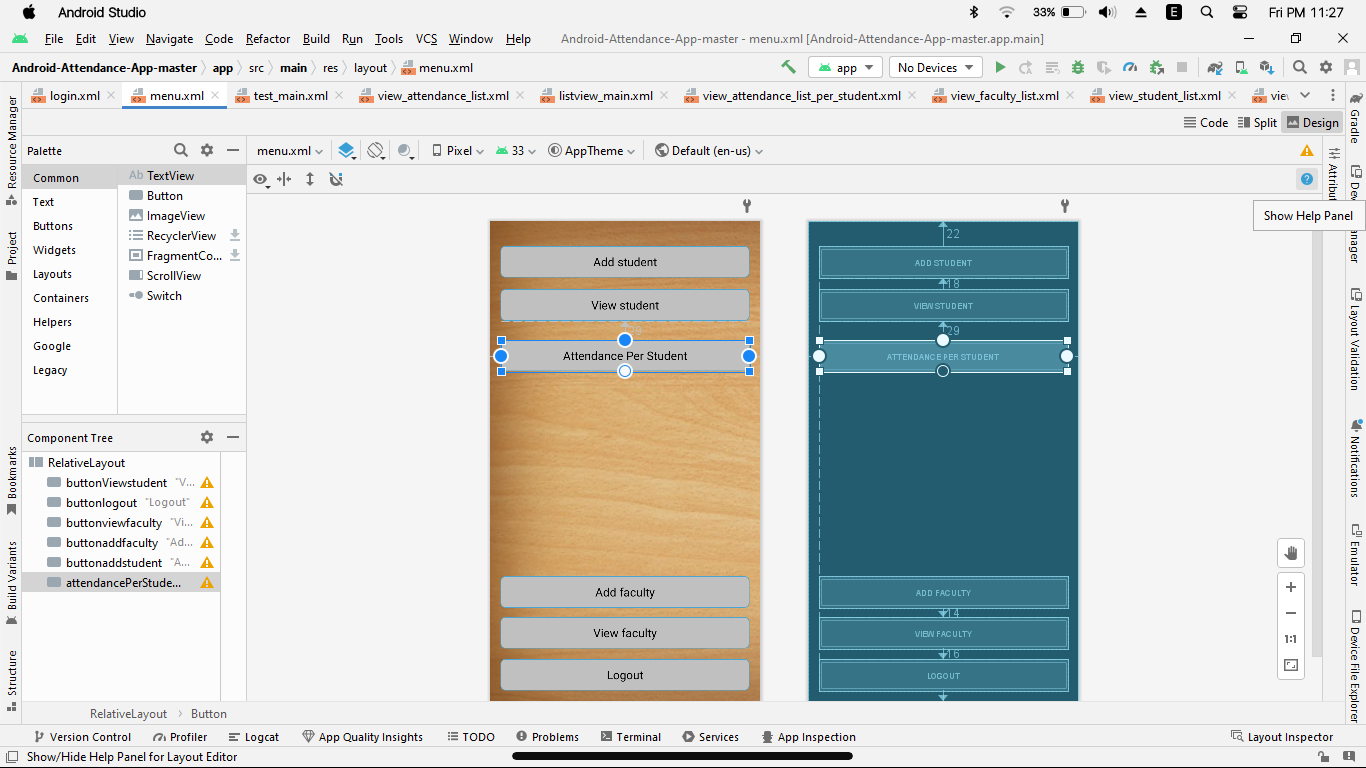
**View Students Page: -**



**Add items Page: -**



**Add Student Page : -**



**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**Conclusion**

To produce the XML files, go to Solution Explorer, right-click on Photo Editor Application, and choose Properties. In the Configuration Properties dialog, specify the file name to be used as the XML documentation file. The XML file will be generated with every build, and the file will be saved in the build directory (bin or bind in the case of the photo editor project). The compiler will now create warnings if public members of a class do not specify an XML-style comment.

addition to the documentation, we apply a label to the software and the documents (at least it should be done for the manually generated documents such as the project plan, the requirements document, and so on). To add a label in Visual Source Safe, open the Source Safe application and select the project. In the File menu choose Label. A window opens where you can specify the label. The label will be something like "Version 0.1.0." This complies with the Microsoft .NET standard, which uses three numbers to identify a specific version. Because the version produced is the first intermediate result, it is labeled with 0 (zero) for the main release, 1 for the intermediate release, and 0 for the minor releases. In addition, the AssemblyInfo.cs file of the Photo Editor Application should be adjusted to correspond to the version in the label before checkin.

What remains is to produce the documentation from the source code. To provide a user's manual in the form of HTML pages, we produce the comment Web pages by going to the Tools menu and selecting the option Build Comment Web Pages. In the displayed message box, we specify the destination where the files will be saved to (in our case, the documentation is stored in the doc folder of the project). Sample comment Web pages can be found in the doc folder in the sample solution. In addition, we produce the XML documentation, which is used for tracking (because it shows the requirement keys that have been implemented).

**FUTURE WORK**

The growth of deep learning technologies has led to the rapid acceleration of computer vision in open-source projects, which has only increased the need for image processing tools. The demand for professionals with [key skills in deep learning](https://www.simplilearn.com/machine-learning-and-deep-learning-skills-to-master-article) technologies is growing at a rapid pace every year. If you want to learn more about image processing and [deep learning benefits](https://www.simplilearn.com/flaws-in-machine-learning-and-how-deep-learning-is-helping-article), Simplilearn’s [Best Deep Learning Course](https://www.simplilearn.com/deep-learning-course-with-tensorflow-training) (with Keras & TensorFlow) Certification Training is the perfect way to set you on the right path. You will master all the deep learning concepts and models using [Keras and TensorFlow frameworks](https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article" \t "_blank" \o "Keras and TensorFlow frameworks) and implement deep learning algorithms. Get started with this course today to get started on a successful career path in deep learning.

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

**CHAPTER 7**

**REFERENCES**

* Y. Jia, C. Rong, C. Wu and Y. Yang, "Research on the decomposition and fusion method for the infrared and visible images based on the guided image filtering and Gaussian filter," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), 2017, pp. 1797-1802, doi: 10.1109/CompComm.2017.8322849.
* S. Yamamoto, Y. Nakajima, S. Tamura, Y. Sato and S. Harino, "Extraction of fluorescent dot traces from a scanning laser ophthalmoscope image sequence by spatio-temporal image analysis: Gabor filter and radon transform filtering," in IEEE Transactions on Biomedical Engineering, vol. 46, no. 11, pp. 1357-1363, Nov. 1999, doi: 10.1109/10.797996.
* G. Healey and D. Slater, "Computing illumination-invariant descriptors of spatially filtered color image regions," in IEEE Transactions on Image Processing, vol. 6, no. 7, pp. 1002-1013, July 1997, doi: 10.1109/83.597275.
* J. Xie, P. -A. Heng and M. Shah, "Image Diffusion Using Saliency Bilateral Filter," in IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 6, pp. 768-771, Nov. 2008, doi: 10.1109/TITB.2008.926462.
* S. Uzun and D. AkgÜn, "A Literature Review on Quadratic Image Filters," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2018, pp. 1-5, doi: 10.1109/ISMSIT.2018.8567038.
* Jian Shen and E. S. Ebbini, "Filter-based coded-excitation system for high-speed ultrasonic imaging," in IEEE Transactions on Medical Imaging, vol. 17, no. 6, pp. 923-934, Dec. 1998, doi: 10.1109/42.746625.