VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

A Mini Project Report

On

“MEETING SCHEDULER”

Submitted in the partial fulfillment for the requirements for
the conferment of degree of

BACHELOR OF ENGINEERING

In

INFORMATION SCIENCE AND ENGINEERING

By

Mr. NARAHARI M V **1BY20IS094**

Under the guidance of

Dr. Gireesh Babu C.N.
Asst. Professor

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
YELAHANKA, BENGALURU-560064
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ISE Dept.
Transform Here

2022-2023

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

BMS INSTITUTE OF TECHNOLOGY& MANAGEMENT

YELAHANKA,BENGALURU-560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **“MEETING SCHEDULER”** is a bonafide work carried out by **Mr. Narahari M V (1BY20IS094)** in partial fulfillment of Mobile Application Development Laboratory with Mini Project (18CSMP68) for the award of **Bachelor of Engineering Degree in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work for the B.E Degree.

Signature of the Guide
Dr. Gireesh Babu C.N.
Associate Professor
Department of ISE

Signature of the HOD
Dr. Pushpa S.K.
Professor and Head
Department of ISE

EXTERNAL EXAMINERS

Name of the Examiners Signature with Date

1.

2.

ABSTRACT

The meeting scheduler app is a powerful tool designed to streamline the process of scheduling and managing meetings. With the ever-increasing complexity of modern work environments and the need for effective collaboration, this app aims to simplify the task of organizing meetings while maximizing productivity and minimizing scheduling conflicts.

While By leveraging advanced calendar integration and automated reminders, the app eliminates the traditional back-and-forth communication, reducing the time and effort required to coordinate meeting times. Users can seamlessly view their availability, propose meeting slots, and receive instant notifications of upcoming meetings, ensuring that all participants stay informed and engaged.

One of the key features of the app is its ability to handle time zone differences effortlessly. With teams spread across different regions, scheduling meetings can become challenging. However, the meeting scheduler app automatically adjusts meeting times based on participants' time zones, ensuring that everyone is synchronized and avoiding any confusion or missed appointments.

ACKNOWLEDGEMENT

We heartily thank our Principal **Dr. Mohan Babu G N**, BMS Institute of technology and Management for his constant encouragement and Inspiration in doing the project.

We heartily thank our Head of the Department **Dr. Pushpa S K**, Dept of Information Science and Engineering, BMS Institute of technology and Management for her constant encouragement and Inspiration in doing the project.

We gracefully thank our project guide **Dr. Gireesh Babu C N**, Assistant Professor, Dept. of Information Science and Engineering for his constant support, advice, guidance and inspiration for the project.

Nevertheless, we express our gratitude towards our family and friends for the encouragement and support which helped us to finish this project successfully.

TABLE OF CONTENTS

List of Figures/Tables

z

CHAPTER-1

1. INTRODUCTION

To address these challenges, the meeting scheduler app has emerged as a powerful solution. This app is designed to simplify the process of scheduling meetings, enhance productivity, and streamline communication among participants. By leveraging technology and automation, it revolutionizes the way meetings are organized and managed, saving time and improving efficiency.

Whether accessed through a desktop or mobile device, the meeting scheduler app offers convenience and accessibility, empowering users to stay organized and on top of their schedules wherever they are.

1.1 INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

Mobile application development is the process to making software for Smartphone and digital assistants, most commonly for Android and iOS.
The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, C# and HTML5.

Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transaction and access information. Today, mobile devices- and the mobile applications that unlock their value-are the most popular way for people and business to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners and employees' demand.

1.2WHAT IS MOBILE APP?

A mobile application or mobile app is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so that there are now millions of apps available.

Apps are generally downloaded from application distribution platforms which are operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store.
Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.

Mobile App has many advantages like within a short app we can communicate a lot of information to the client/customers and even it is an ease of access to client/customer for services update or sale/purchase activity.

1.2 WHAT IS MOBILE OS?

A mobile operating system is an operating system for mobile phones, tablets, smart watches, 2-in-1 PCs, smart speakers, or other mobile devices. While computers such as typical laptops are ‘mobile’, the operating systems used on them are generally not considered mobile ones, as they were originally designed for desktop computers that historically did not have or need specific mobile features. This distinction is becoming blurred in some newer operating systems that are hybrid made for both uses.

A mobile OS is responsible for identifying and defining mobile device features and functions, including keypads, application synchronization, email, thumbwheel and text messaging. A mobile OS is similar to a standard OS (like Windows, Linux, and Mac) but is relatively simple and light and primarily manages the wireless variations of local and broadband connections, mobile multimedia and various input methods.

1.3 INTRODUCTION TO 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.
- Apply Changes to push code and resource changes to your running app without restarting your app.
- 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.

Android Studio provides a unified environment where we can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto.

1.4 ANDROID ARCHITECTURE

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services. Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

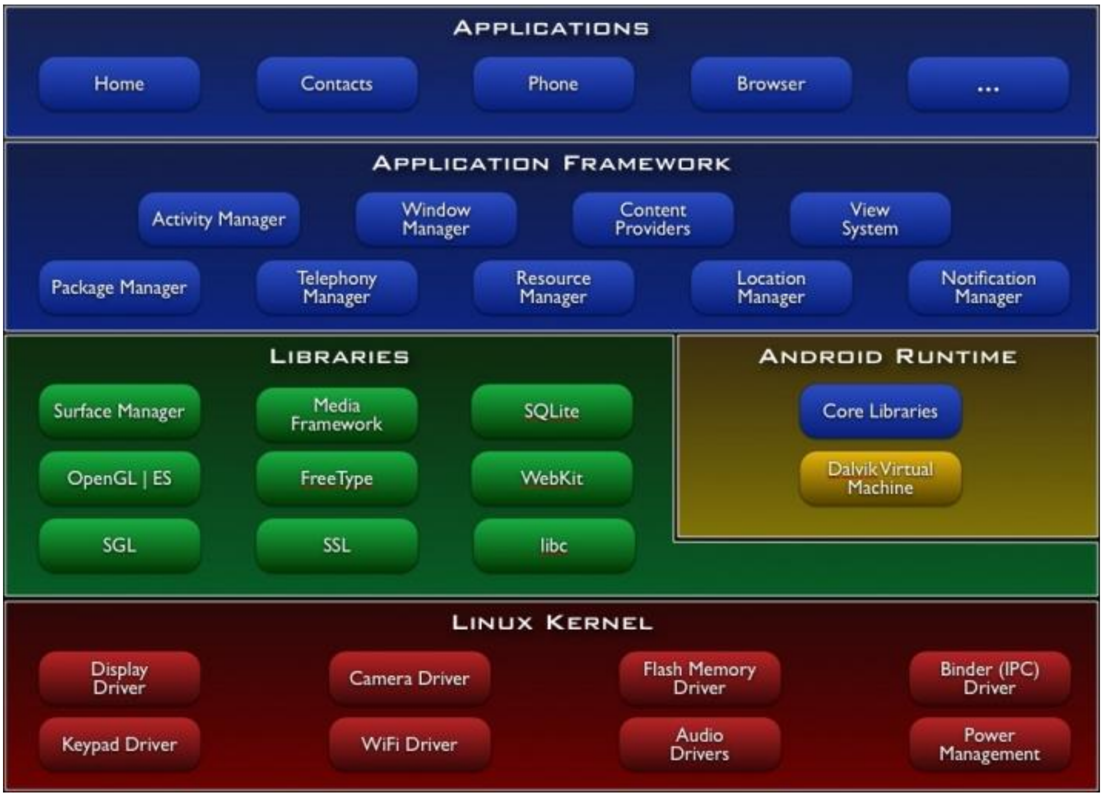
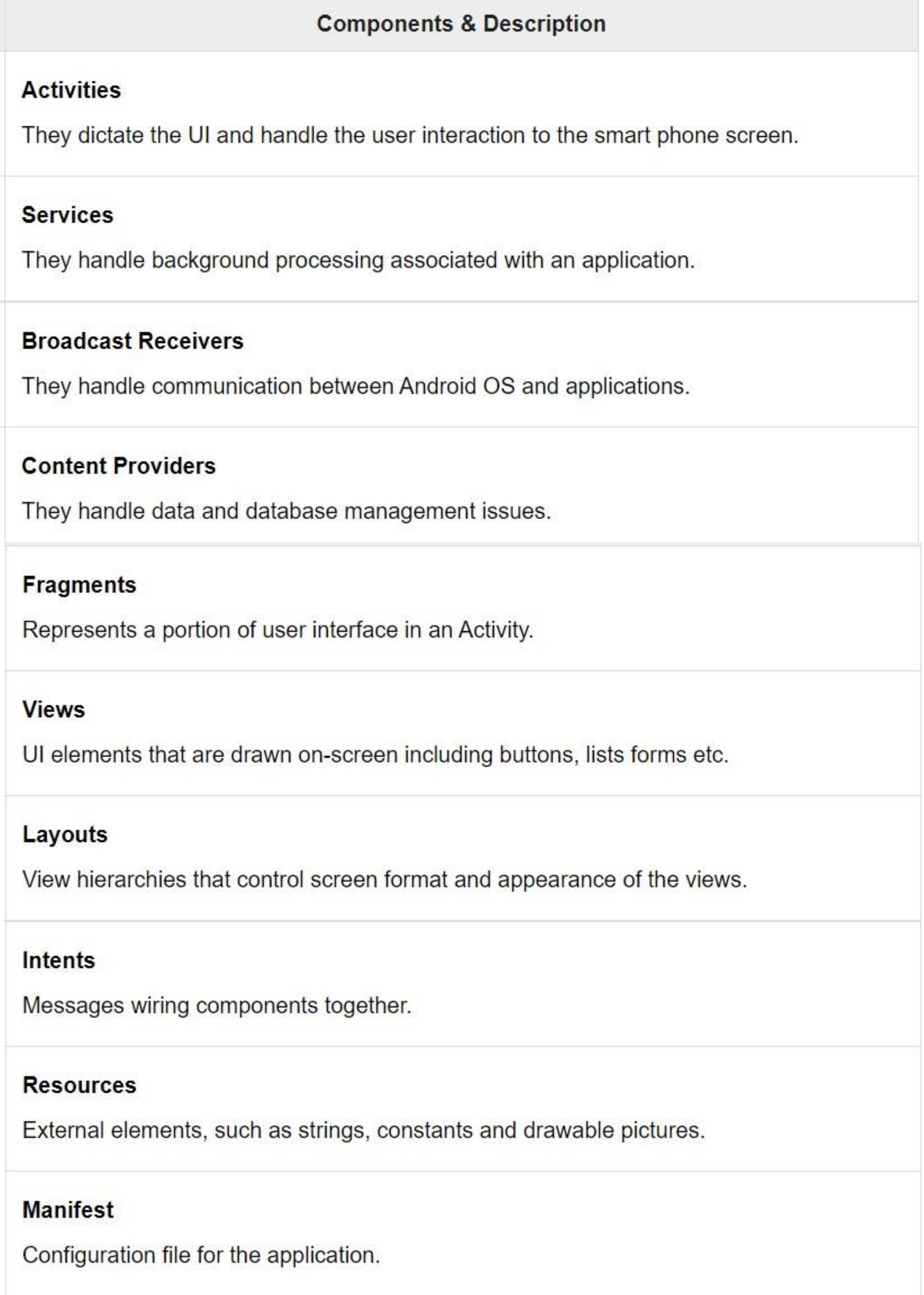


fig 1.1: Android Architecture

1.5 ANDROID APPLICATION COMPONENTS

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact.



Components & Description
Activities They dictate the UI and handle the user interaction to the smart phone screen.
Services They handle background processing associated with an application.
Broadcast Receivers They handle communication between Android OS and applications.
Content Providers They handle data and database management issues.
Fragments Represents a portion of user interface in an Activity.
Views UI elements that are drawn on-screen including buttons, lists forms etc.
Layouts View hierarchies that control screen format and appearance of the views.
Intents Messages wiring components together.
Resources External elements, such as strings, constants and drawable pictures.
Manifest Configuration file for the application.

Fig 1.2 Android components

1.6 PROBLEM STATEMENT

The current process of scheduling and managing meetings is time-consuming, error-prone, and inefficient, leading to scheduling conflicts and wasted resources. The current process of scheduling and managing meetings is time-consuming, error-prone, and inefficient, leading to scheduling conflicts and wasted resources.

1.7 OBJECTIVE

- A Develop a meeting scheduler app that simplifies the process of scheduling
- Enhance collaboration by providing features for file sharing, note-taking, and task assignment within the app.
- Optimize meeting efficiency through automated reminders, time zone management, and customization options..

1.8 PROJECT APPLICATION

- A meeting scheduler app can provide a centralized platform for users to schedule and manage meetings efficiently. It eliminates the need for back-and-forth emails or phone calls, saving time and reducing scheduling conflicts.
- A mobile app version of the meeting scheduler ensures that users can access their schedules and manage meetings on the go. It provides convenience and flexibility, enabling users to stay organized even when they are not in front of their computers. This ensures that all appointments are synchronized and avoids double bookings.

CHAPTER-2

2. SYSTEM DESIGN

2.1 APPLICATION COMPONENTS OF PROJECT

1. Activities components functioning:

The system shall support organization of meetings and determine, for each meeting request, a meeting date and location so that most of the intended participants will effectively participate Organize meetings based on Contact.

User can scroll through the meetings scheduled with a particular contact, add new meetings with the same contact and make changes to the existing meeting. Purge all meetings of a contact. Organize meetings based on Contact. User can scroll through the meetings scheduled with a particular contact, add new meetings with the same contact and make changes to the existing meeting. This feature allows the user to remove all the scheduled meetings with a particular contact

Re-plan a meeting to support the changing user constraints, that include modification to the exclusion set, preference set and/or preferred location before a meeting date/location is proposed Re-plan a meeting to support some external constraints, after a date and location have been proposed The system shall support conflict resolution according to resolution policies. The system shall support the negotiation and conflict resolution processes

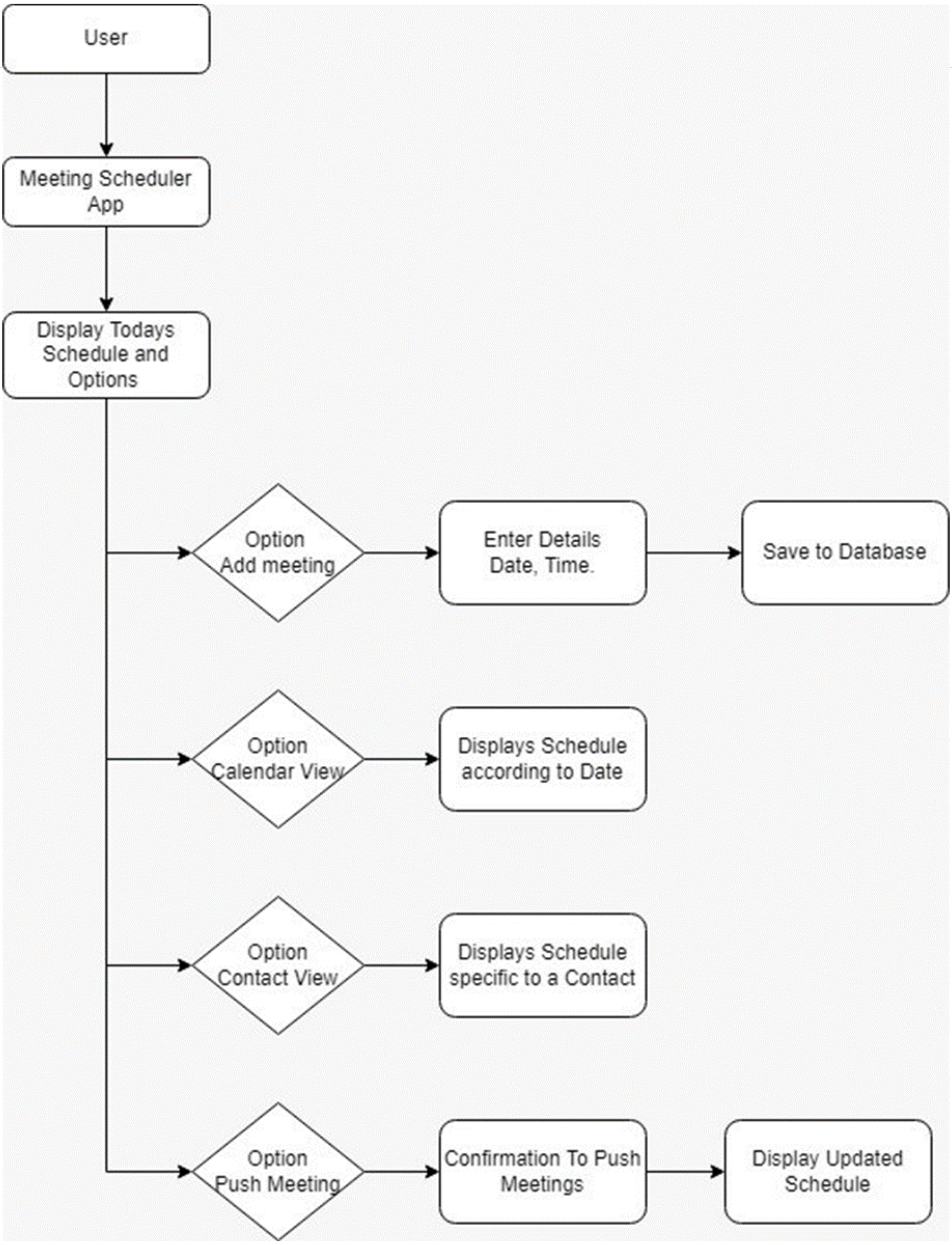

User
Meeting Scheduler App
Display Todays Schedule and Options
Option Add meeting
Enter Details Date, Time.
Save to Database
Option Calendar View
Displays Schedule according to Date
Option Contact View
Displays Schedule specific to a Contact
Option Push Meeting
Confirmation To Push Meetings
Display Updated Schedule

Fig 2.1: Component Design

2.2 USER INTERFACE DESIGN

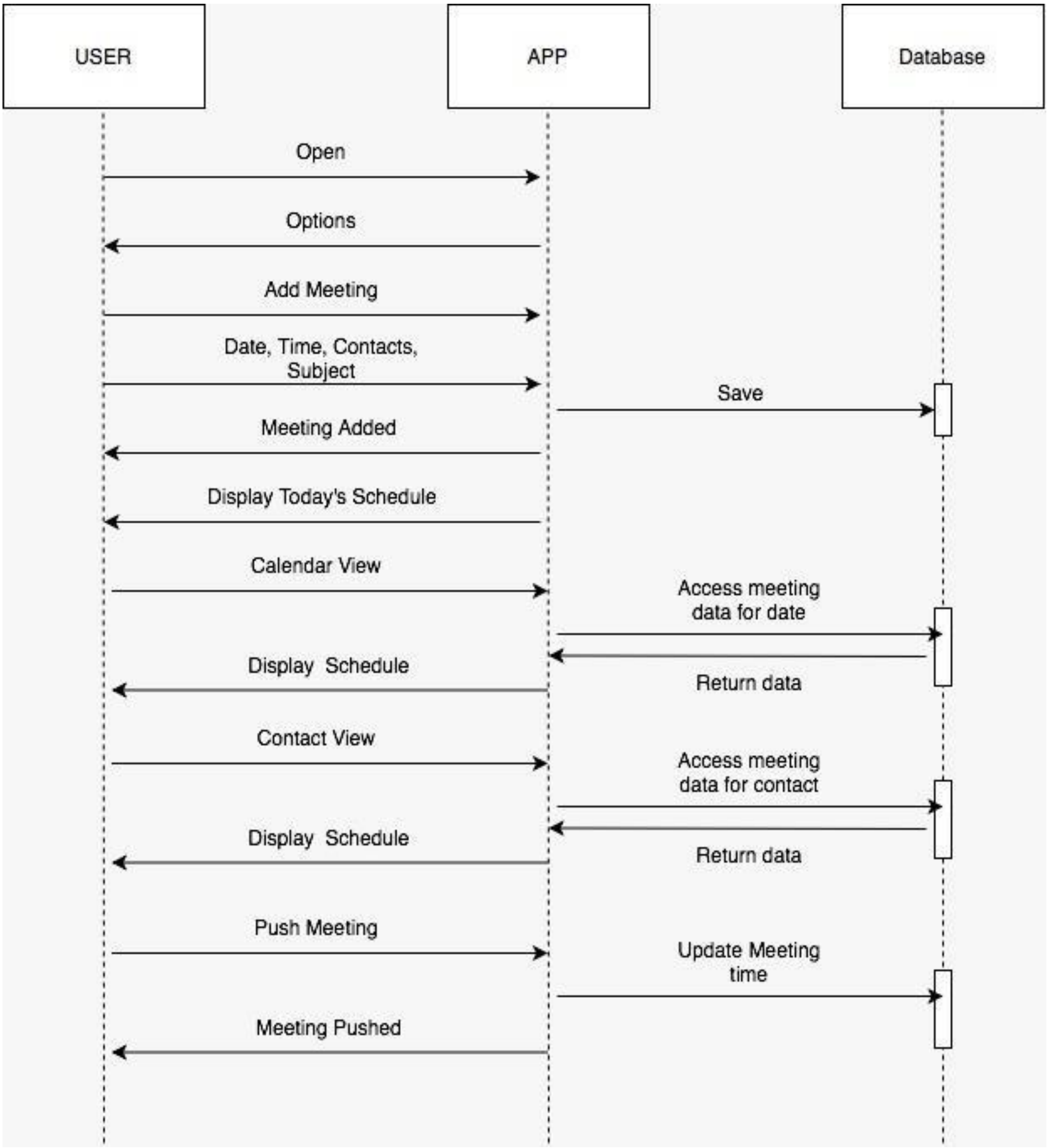


Fig 2.2: Component Design

The user interface (UI) for an Android app is built as a hierarchy of layouts and widgets. The layouts are View Group objects, containers that control how their child views are positioned on the screen. Widgets are View objects, UI components such as buttons and text boxes.

The Application is designed using the components of android such as, Layouts, Buttons, Text Views, Edit Texts etc. The above figure gives a brief design of the application and its flow. The app is designed using Constraint Layout which is a View Group containing many Views such as Edit Texts, Buttons etc. The Constraint Layout gives the option of constraining the views with respect to the parent i.e., View Group or other Views in the layout.

The We can say that it is a type of container which contains the abstract methods. It provides the facility to use the methods which declaration is already done and we can use them according to our need.

Inheritance provides ease to the developer to develop the applications. By using inheritance, you can use basic methods without defining it in the program by extending the class. It is simple also because multiple inheritance is not allowed in Java which reduces the complexity.

It allows you to do the tasks on multiple threads instead of a single thread which reduce the chances of crash your app. If a single thread is there, then the load increases on that activity and it results into crash of the app. It provides built-in class which facilitates us to work easily with multiple threads named Async Task.

Java is platform independent, it means that Java code can run on any platform. It doesn’t require the source code on that machine at which platform, where it will be executed. A class file (.class file) can be executed on any platform i.e Windows, Linux, Mac etc. It means your code is portable.

That’s why an application can run on several devices without changing any code. This feature allows that the application can run on any smartphone whether it is Samsung, Nokia, Lenovo etc. Android Studio (IDE) uses DVM (Dalvik Virtual Machine) instead of JVM (Java Virtual Machine). The file generated by DVM is the dex format.

CHAPTER-3

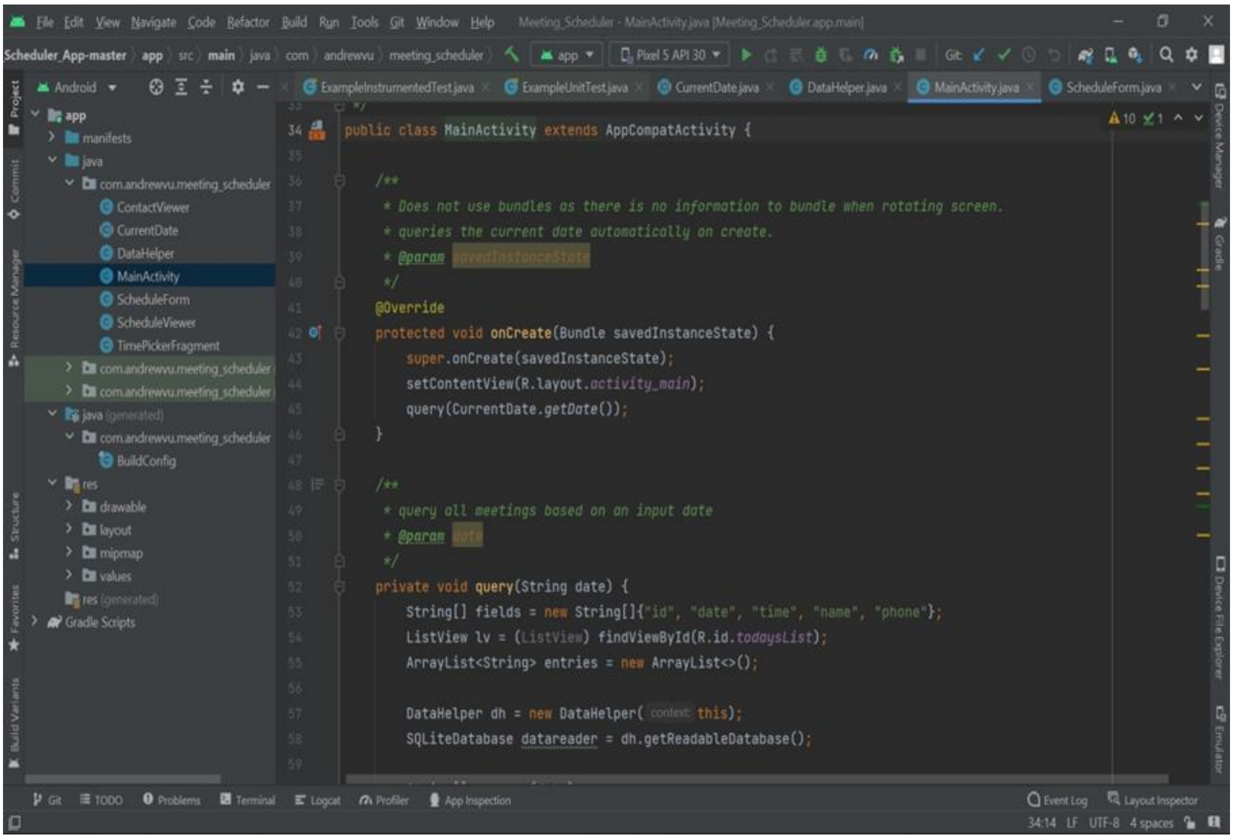
3. IMPLEMENTATION

This is the implementation of the app explained with different modules with respective Features and code.

3.1 EXPLANATION OF THE MODULES WITH JAVA AND XML CODE

1. MainActivity.java:

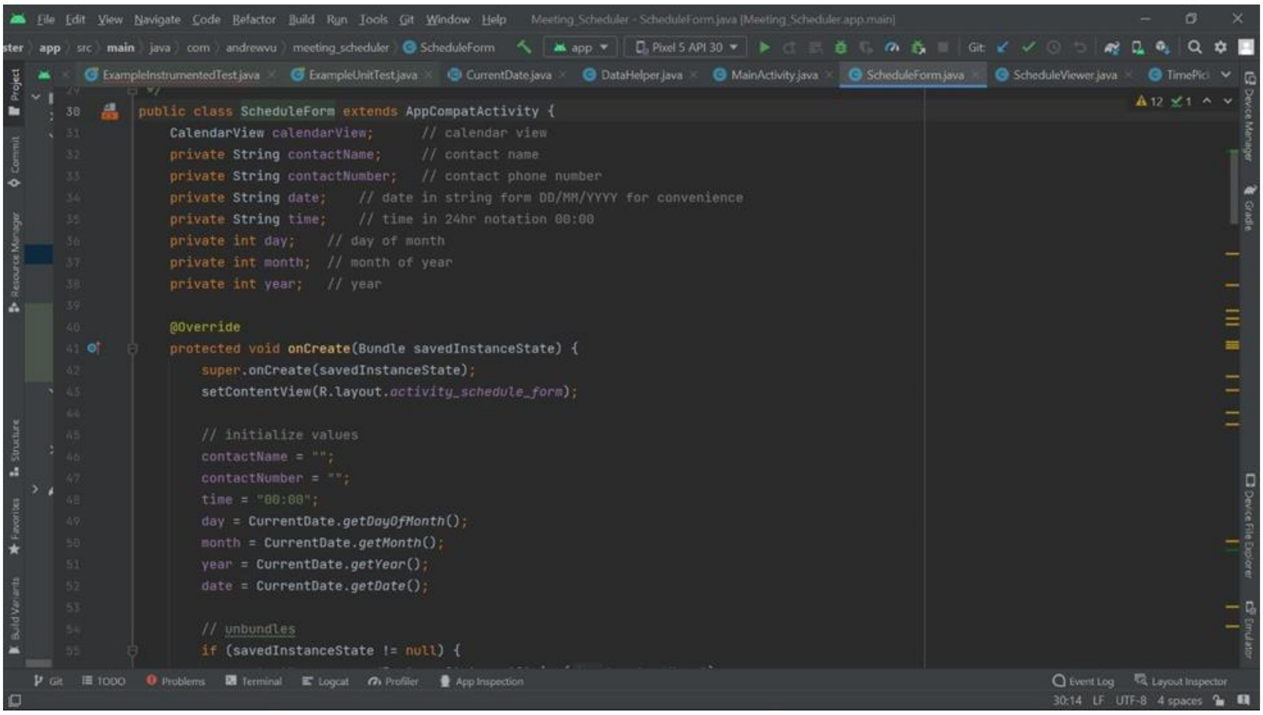
Displays all meetings for the current date automatically. Buttons are present to add additional meetings, switch to a calendar-based schedule view, switch to a contact-based schedule view, or push today's meetings to another date. The date pushed to will be the next weekday, if the current day is a weekday, or the next weekend day otherwise



```java
public class MainActivity extends AppCompatActivity {

    /**
     * Does not use bundles as there is no information to bundle when rotating screen.
     * queries the current date automatically on create.
     * @param savedInstanceState
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        query(CurrentDate.getDate());
    }

    /**
     * query all meetings based on an input date
     * @param date
     */
    private void query(String date) {
        String[] fields = new String[]{"id", "date", "time", "name", "phone"};
        ListView lv = (ListView) findViewById(R.id.todaysList);
        ArrayList<String> entries = new ArrayList<>();

        DataHelper dh = new DataHelper( context: this);
        SQLiteDatabase datareader = dh.getReadableDatabase();

```

2. Scheduling a Meeting:

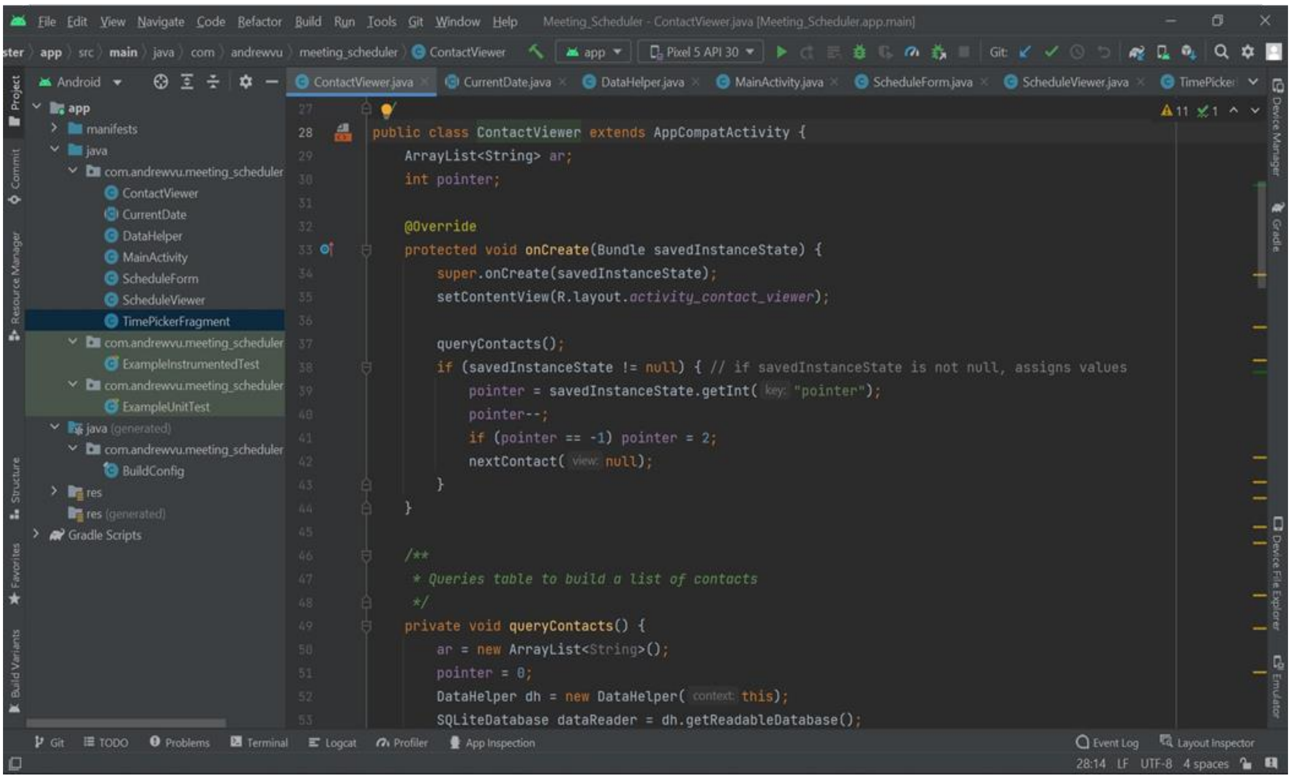
The meeting form includes a calendar widget to select the date, a button and corresponding textview to generate a time picker widget fragment to select the time, a button and corresponding textview to create a new intent to select a contact, and a save and cancel button. Pressing the cancel button bring up a prompt to prevent accidental information loss. Contact information is not required to submit a meeting.The save button commits the form data to an SQLite database.



```java
public class ScheduleForm extends AppCompatActivity {
    CalendarView calendarView;      // calendar view
    private String contactName;     // contact name
    private String contactNumber;   // contact phone number
    private String date;    // date in string form DD/MM/YYYY for convenience
    private String time;    // time in 24hr notation 00:00
    private int day;    // day of month
    private int month;  // month of year
    private int year;   // year

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_schedule_form);

        // initialize values
        contactName = "";
        contactNumber = "";
        time = "00:00";
        day = CurrentDate.getDayOfMonth();
        month = CurrentDate.getMonth();
        year = CurrentDate.getYear();
        date = CurrentDate.getDate();

        // unbundles
        if (savedInstanceState != null) {
```

Contact Based Schedule View:

This activity displays a listview and two buttons. Pressing the left button will iterate through an array list of contacts, updating the listview with all meetings associated with the contact. Any meetings with no contacts listed is grouped together in an appropriately labeled category. Pressing the right button will purge all meetings with the contact. A confirmation prompt is used to prevent accidental data loss.



```java
public class ContactViewer extends AppCompatActivity {
    ArrayList<String> ar;
    int pointer;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_contact_viewer);

        queryContacts();
        if (savedInstanceState != null) { // if savedInstanceState is not null, assigns values
            pointer = savedInstanceState.getInt("pointer");
            pointer--;
            if (pointer == -1) pointer = 2;
            nextContact(null);
        }
    }

    /**
     * Queries table to build a list of contacts
     */
    private void queryContacts() {
        ar = new ArrayList<String>();
        pointer = 0;
        DataHelper dh = new DataHelper(this);
        SQLiteDatabase dataReader = dh.getReadableDatabase();
```

CHAPTER-4

RESULTS

These are the result outlook of the current app and its visibility with representing Features.

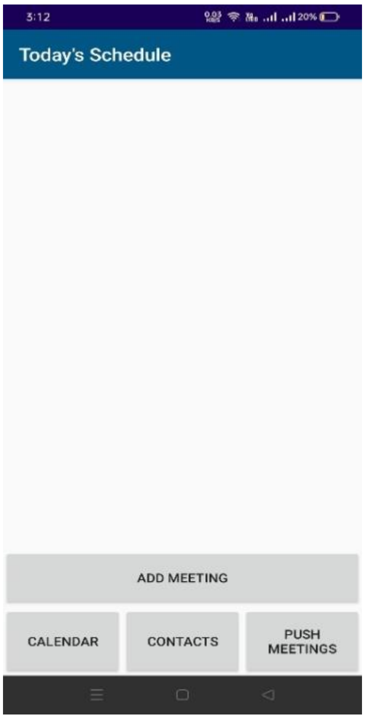
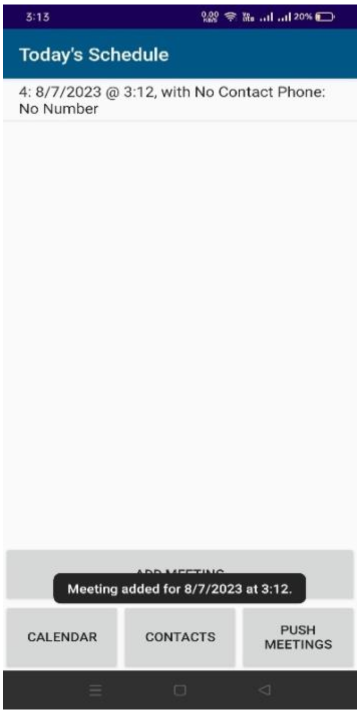


Fig. 4.1 This figure represents the Todays Schedule for the meeting involved.



4: 8/7/2023 @ 3:12, with No Contact Phone: No Number

Meeting added for 8/7/2023 at 3:12.

Fig. 4.2 This figure shows the Meeting Added in the todays Schedule

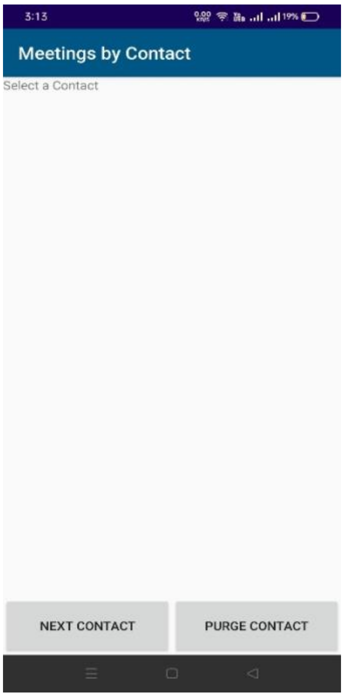
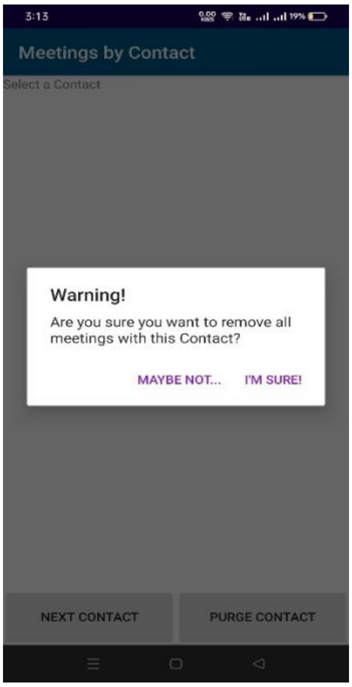


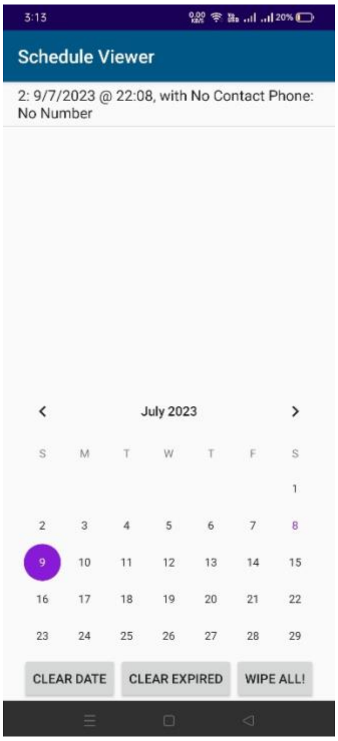
Fig. 4.3 This figure shows meetings by contact



Warning!

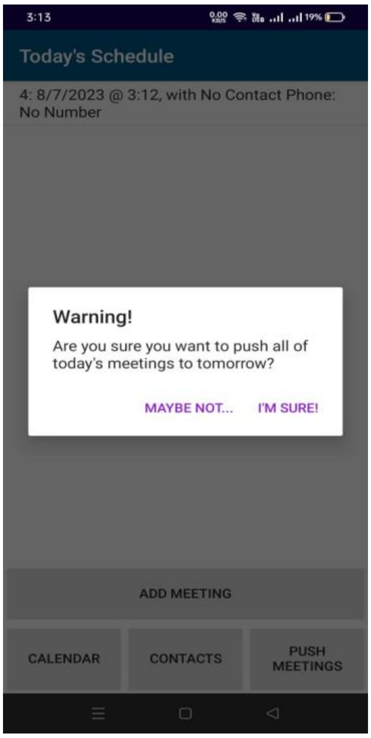
Are you sure you want to remove all meetings with this Contact?

Fig. 4.4 This figure shows a message for removing meeting contact.



2: 9/7/2023 @ 22:08, with No Contact Phone: No Number

Fig. 4.5 This figure shows view of Schedules

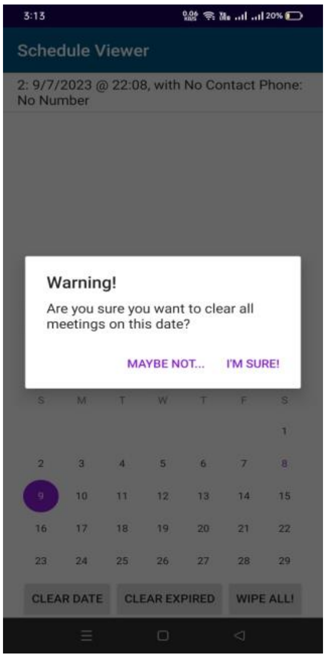


4: 8/7/2023 @ 3:12, with No Contact Phone: No Number

Warning!

Are you sure you want to push all of today's meetings to tomorrow?

Fig. 4.6 This figure shows the notification for Pushing down meetings for tomorrow.



2: 9/7/2023 @ 22:08, with No Contact Phone: No Number

Warning!

Are you sure you want to clear all meetings on this date?

Fig. 4.7 This figure shows to clear the meeting on this Date.

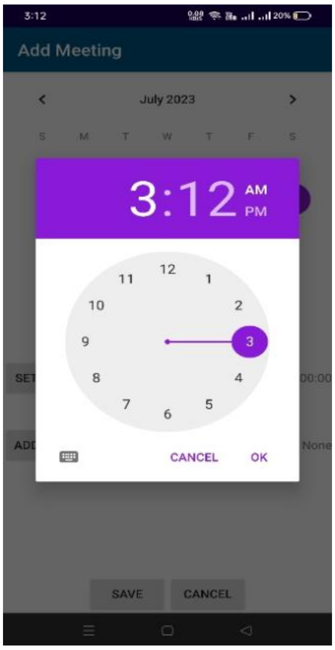


Fig. 4.8 This figure shows the option to select the time for the meeting.

CHAPTER-5

5.APPLICATION TESTING

Application testing shows where the app is tested under various system unit with test cases and obtained

System unit	Test case	Result
Main Activity	Click on open	Navigates to options
Main Activity	Click on options	Navigates to add meeting
Main Activity	Click on Date, time, contacts	Navigates to save
Access meeting	Click on date	Returns data
Access meeting	Meeting data	Display Schedule
Meeting Activity	Accessing data.	Navigates to data
Push meeting	Push button	Postponing the meeting
Push meeting	meeting	Navigates to pushed meeting
purge	Purge button	Purging contacts
calendar	Calendar button	Calendar view

Table 5.1. This table represents the application testing

CHAPTER-6

CONCLUSION

6.1. CONCLUSION

App development is a continuous process with many iterations. There is no such thing as a perfect design and the app has to dynamically be able to change to follow the trends because the App World is dynamic.This application is intended to be more useful than existing schedule reminder application by offering dynamic notifications to user. The application manages the event schedules for user by keeping track of all the scheduled events.It keeps track of all events that are scheduled based on contacts, days and dates. It helps get an overall view of all meetings scheduled with a particular contact which can be quite resourceful whilst scheduling and organizing future meetings with the same person.

6.2 FUTURE ENHANCEMENTS

Implement additional functionality to broadcast meeting information with all the contacts. Allow a single meeting to be created with multiple

Intelligent meeting suggestions: Implement AI algorithms to analyze participants' availability and preferences, and provide intelligent meeting suggestions based on optimal time slots, reducing the need for manual coordination

Intelligent meeting suggestions: Implement AI algorithms to analyze participants' availability and preferences, and provide intelligent meeting suggestions based on optimal time slots, reducing the need for manual coordination..

CHAPTER-7

REFERENCES

[1]. C. Ng, S. Cheong, E. Hajimohammadhosseinmemar and W. Yap, "Mobile outdoor parking space detection application," 2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), 2017, pp. 81-86, doi: 10.1109/ICSGRC.2017.8070573.

[2]. B. K. Patil, A. Deshpande, S. Suryavanshi, R. Magdum and B. Manjunath, "Smart Parking System for Cars," 2018 International Conference on Recent Innovations in Electrical, Electronics & Engineering (ICRIEECE), 2018, pp. 1118-1121, doi: 10.1109/ICRIEECE44171.2018.9008662.

[3]. Calendly: “Simplify your schedule." - Calendly is a popular meeting scheduler app that Offers seamless calendar integration, automated reminders, and time zone management to simplify the scheduling process. (Website: https://calendly.com)

[4]. “Doodle: Easy scheduling." - Doodle is a meeting scheduler app that allows participants to propose meeting times, vote on options, and select the most suitable time slot, reducing scheduling conflicts. (Website: https://doodle.com)