ABSTRACT

The Meeting information application aimed to streamline the management of meeting-related information within an organization, improving efficiency, collaboration, and decision-making processes. The project involved the development and implementation of comprehensive system or tool to facilitate the collection, organization, and retrieval of meeting- related data, such as agendas, minutes, action items, and participant information.

Key outcomes of the meeting information application included the successful development and implementation of the meeting information system, improved accessibility and retrieval of meeting-related data, enhanced collaboration among team members, and increased efficiency in meeting-related processes. Additionally, the project identified areas for further improvement and provided recommendations for ongoing system enhancements and maintenance.

Overall, the meeting information project significantly contributed to optimizing the management of meeting-related information, resulting in improved organizational effectiveness and decision-making.

Contents

INTRODUCTION

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.

Meeting information application designed to revolutionize the way organizations handle meeting-related data. The application provides a centralized platform for managing and organizing various aspects of meetings, including agendas, minutes, action items, participant information, and follow-up tasks. By leveraging technology and best practices, the meeting information application aims to streamline meeting processes, enhance collaboration, and improve overall productivity.

Problems

- Information Disorganization
- Poor Collaboration and Communication
- Reduced Accountability
- Reduced Productivity
- Ineffective Follow-up
- Missed Opportunities for Knowledge Sharing

REQUIREMENTS AND SPECIFICATION

HARDWARE REQUIREMENTS

- 1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor.
- 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)
- 16 GB available hard disk space (32-bit) or 20 GB (64-bit)
- DirectX 9 graphics device with WDDM 1.0 or higher driver.

SOFTWARE REQUIREMENTS

- Operating system: WINDOWS 7,8,8.1,10,11.
- Idle : Android Studio

- Java
- Android SDK
- .java JDK
- XM

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        SectionsPagerAdapter sectionsPagerAdapter = new
SectionsPagerAdapter(this, getSupportFragmentManager());
        ViewPager viewPager = findViewById(R.id.view_pager);
        viewPager.setAdapter(sectionsPagerAdapter);
        TabLayout tabs = findViewById(R.id.tabs);
        tabs.setupWithViewPager(viewPager);




    }
}
```

SOURCE CODE

Main activity: java code

```
package com.example.meetinginfo;

import android.os.Bundle;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

import com.google.android.material.snackbar.Snackbar;

import com.google.android.material.tabs.TabLayout;


import androidx.viewpager.widget.ViewPager;

import androidx.appcompat.app.AppCompatActivity;


import android.view.Menu;

import android.view.MenuItem;

import android.view.View;


import com.example.meetinginfo.ui.main.SectionsPagerAdapter;


public class MainActivity extends AppCompatActivity {


  @Override
```

Xml code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/design_default_color_error"
        android:theme="@style/Theme.MeetingInfo.AppBarOverlay">

        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
```

```
        android:background="@color/purple_500"
        android:gravity="center"
        android:minHeight="?actionBarSize"
        android:padding="@dimen/appbar_padding"
        android:text="Meeting Details"
        android:textAlignment="center"
android:textAppearance="@style/TextAppearance.Widget.AppCompat.Toolbar.Title" />

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="37dp"
        android:background="#FFEB3B"
        app:tabIndicatorColor="#00E5FF" />
  </com.google.android.material.appbar.AppBarLayout>

  <androidx.viewpager.widget.ViewPager
    android:id="@+id/view_pager"
    android:layout_width="match_parent"
    android:layout_height="499dp"
    app:layout_behavior="@string/appbar_scrolling_view_behavior">
```

```
        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="TextView" />
    </androidx.viewpager.widget.ViewPager>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Fragmentation code

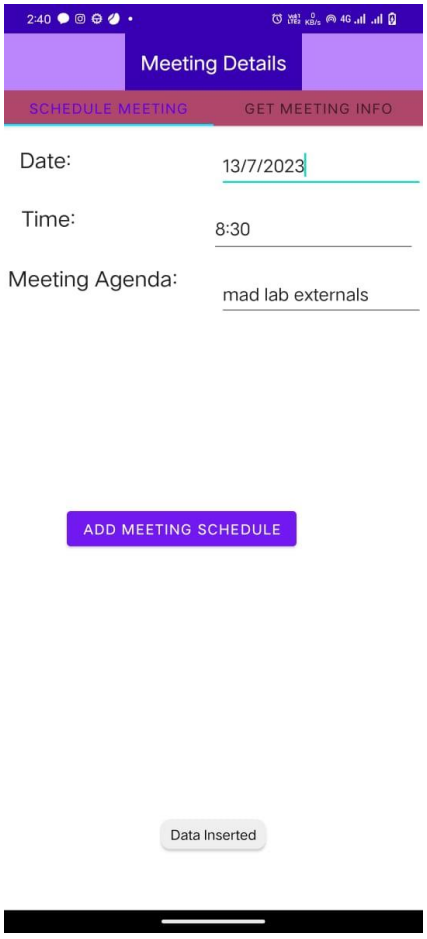
```
package com.example.meetinginfo;


import android.content.Context;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.inputmethod.InputMethodManager;

import android.widget.Button;

import android.widget.CalendarView;

import android.widget.EditText;

import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;


import static androidx.core.content.ContextCompat.getSystemService;


public class Fragment1 extends Fragment {
```

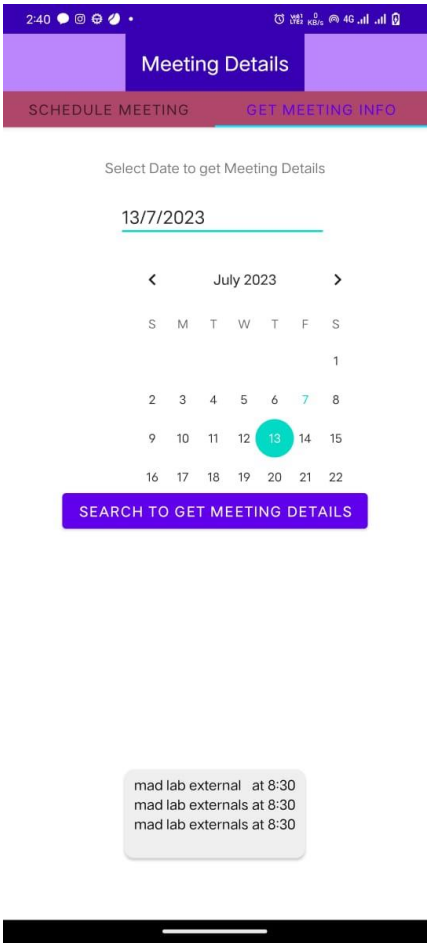
```
    EditText date,time,agenda;
    DataBaseConn dbc;
    CalendarView calendarView;
    Button btn;
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.fragment_layout1,container,false);
        date=view.findViewById(R.id.txtDate);
        time=view.findViewById(R.id.txtTime);
        agenda=view.findViewById(R.id.txtAgenda);
        btn=view.findViewById(R.id.btn1);
        calendarView=view.findViewById(R.id.mCal);
        dbc=new DataBaseConn(getActivity());    //need to initialize here only
        calendarView.setVisibility(View.INVISIBLE);
        date.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                closeKeyBoard();
                calendarView.setVisibility(View.VISIBLE);
                calendarView.setOnDateChangeListener(new
CalendarView.OnDateChangeListener() {
```

```
            @Override
            public void onSelectedDayChange(@NonNull CalendarView view, int year, int
month, int dayOfMonth) {
                String d=dayOfMonth+"/"+(month+1)+"/"+year;
                date.setText(d);
                calendarView.setVisibility(View.INVISIBLE);
            }
        });
    }
});
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String mdate,mTime,mAgenda;
        mdate=date.getText().toString();
        mTime=time.getText().toString();
        mAgenda=agenda.getText().toString();

       Boolean insert=dbc.insertvalue(mdate,mTime,mAgenda);
        if(insert==true){
            Toast.makeText(getActivity(),"Data Inserted",Toast.LENGTH_SHORT).show();

        }
```

```
            else
                Toast.makeText(getActivity(),"Data NOT
Inserted",Toast.LENGTH_SHORT).show();
            //txt.setText("NOT INSERTED");



          }



       });







    return view;
  }
  private void closeKeyBoard(){
    View  view = getActivity().getCurrentFocus();
     if (view != null) {
        InputMethodManager imm = (InputMethodManager)
              getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
      }
```

SNAPSHORTS


2:40
Meeting Details
SCHEDULE MEETING
GET MEETING INFO
Date:
13/7/2023
Time:
8:30
Meeting Agenda:
mad lab externals
ADD MEETING SCHEDULE
Data Inserted


2:40
Meeting Details
SCHEDULE MEETING
GET MEETING INFO
Select Date to get Meeting Details
13/7/2023
July 2023
S M T W T F S
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
SEARCH TO GET MEETING DETAILS
mad lab external at 8:30
mad lab externals at 8:30
mad lab externals at 8:30

IMPLEMENTATION

The meeting information app developed in Android Studio provides a simple yet effective solution for managing and organizing meeting-related data. The app allows users to create meetings, view agendas, and manage participants efficiently.

Through a user-friendly interface, users can input meeting details such as title, date, time, and location. The app stores this information in a database, facilitating easy retrieval and display of meeting agendas. Participants can view the agenda items, helping them stay informed and prepared for the meeting.

CONCULSION

The Meeting Information App aims to simplify meeting management by providing a centralized platform for creating, scheduling, and organizing meetings. Through thoughtful implementation and continuous improvement, the app can enhance productivity within an organization. Users with the necessary permissions can create meetings, specifying details like title, date, time, duration,

BILOGAPHY

- www.google.com
- open.ai
- http://www.github.com
- youtube.com