OPEN SOURCE SOFWARE LAB
MINOR PROJECT
FINAL REPORT

BATCH: B5

TITLE: JIIT HUBDATES

Under the guidance of Ms. Ankita Wadhwa and Ms. Megha Rathi

Submitted By:
Shubhra Singh – 14103177
Sanya Deshmukh - 14103304
Pratigya Agarwal - 14103180

ABSTRACT

In today’s era, getting awareness about day-to-day happenings is a tedious task, especially when you are the one being in an institution or a college. The one thing that has made this possible is “Facebook” but also to a certain extent .But as everybody is not on Facebook and also, the freshers are not aware about these groups being active on Facebook. In order to make this possible and a little easy for everybody, why don’t we create an android application that makes all these things available on our fingertips.

In our own college, when talking about various hubs and societies like Programming hub, Robotics hubs, biotech hub i.e. Ribose and many more, it certainly becomes a hardship to pay attention to all those activities around. So, to have access to everything at one platform with every update in one touch, we designed this application. With a user friendly and interactive UI, it comes with easily accessible future events, updates, notifications etc.

With a quick access to the list of hubs, this application, which provides a secure login using Facebook, also provides user with the option to enable the push notifications on marking any event of any hub.

OBJECTIVES

- Instead of joining several pages and groups on Facebook, some of which are not even known to us, rather having a single platform for all the hubs of our college.
- Secure login using Facebook to avoid any privacy leaks.
- Separate login for the admin.
- Easy access to all the upcoming events of each hub under their

respective names at one place (using open source technologies).

- Push Notifications option for marking any specific event to be held.

BACKGROUND STUDY AND FINDINGS

There are many apps made by other colleges like NSIT, dtu etc. for their societies. But, when searching about the apps made for our college, we found that there are several applications already existing made for JIIT like Webkiosk, Annapurna Menu and the one recently made “JIIT Connect”. These apps were made for different purposes but none of the app covered the hubs of our college. No app was ever made which could provide information about all the existing societies or any one of it. This was a major concern as making an application of this kind could really help students avoid the hustle of joining any closed Facebook group for each of the hubs. This application, which is based on the open source technologies, is very beneficial for being up-to-date about the future events, their venues, timings, turning on push notifications. Also, only the admin has the access to create events which makes it secure. The most important aspect of it is the secured Facebook login.

DESIGNING

- Navigation drawer: The navigation drawer is a panel that displays the app’s main navigation options on the left edge of the screen .Sliding Menu (Navigation Drawer) will be hidden and can be shown by swiping the screen from left edge to right or tapping the app icon on the action bar. In our application, Navigation drawer is containing the list of all the hubs.

- Spinner: Spinners provide a quick way to select one value from a set. In the default state, a spinner shows its currently selected value. Touching the spinner displays a dropdown menu with all other available values, from which the user can select a new one. We have used the spinner for the choice of hubs in when admin logins.

- ❖ Action Bar: The action bar is a dedicated piece of real estate at the top of each screen that is generally persistent throughout the app. It can display the activity title, icon, actions which can be triggered, additional views and other interactive items. It contains the app’s icon as well.

- ❖ List View: ListView is a view group that displays a list of scrollable items. The list items are automatically inserted to the list using an adapter that pulls content from a source such as an array or database query and converts each item result into a view that's placed into the list. In our application, we have used the list view to display the list of events. Also, it is used to display the list of books for the “parola”hub.

- ❖ Facebook customized login button: The Login Button is a simple way to trigger the Facebook login process on the app. The login button is a UI element that wraps functionality available in the Login Manager. So when someone clicks on the button, the login is initiated with the permissions set in the Login Manager. The button follows the login state, and displays the correct text based on someone's authentication state. We have used the customized button for secure login as also it is an open source technology.

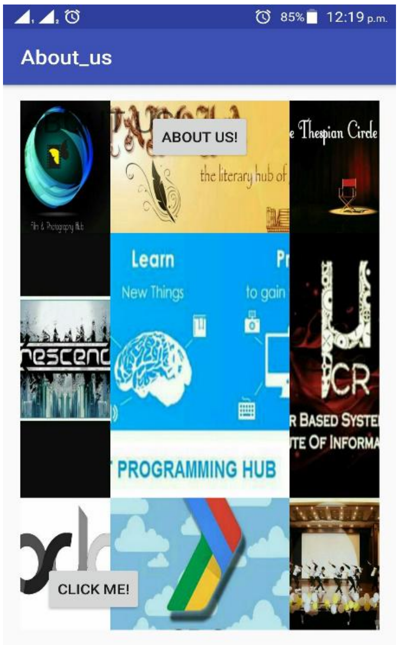
IMPLEMENATION

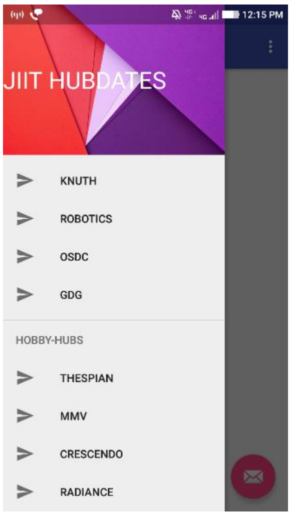
Code:

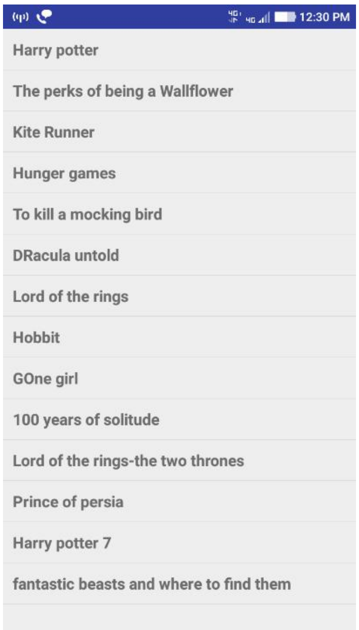
Main Activity

```
package com.example.pratigya.hamblaster.activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;
import com.bumptech.glide.load.engine.DiskCacheStrategy;
import com.example.pratigya.hamblaster.ParolaActivity;
import com.example.pratigya.hamblaster.R;
import com.example.pratigya.hamblaster.fragment.HomeFragment;
import com.example.pratigya.hamblaster.fragment.KnuthFragment;
import com.example.pratigya.hamblaster.fragment.MoviesFragment;
import com.example.pratigya.hamblaster.fragment.NotificationsFragment;
import com.example.pratigya.hamblaster.fragment.PhotosFragment;
import com.example.pratigya.hamblaster.fragment.SettingsFragment;
import com.example.pratigya.hamblaster.other.CircleTransform;

public class MainActivity extends AppCompatActivity {

    private NavigationView navigationView;
    private DrawerLayout drawer;
    private View navHeader;
    private ImageView imgNavHeaderBg, imgProfile;
    private TextView txtName, txtWebsite;
    private Toolbar toolbar;
   // private FloatingActionButton fab;

    // urls to load navigation header background image
    // and profile image
    private static final String urlNavHeaderBg =
"https://upload.wikimedia.org/wikipedia/en/5/56/Logo-jiit.png";
    private static final String urlProfileImg =
"https://upload.wikimedia.org/wikipedia/en/5/56/Logo-jiit.png";

    // index to identify current nav menu item
    public static int navItemIndex = 0;

    // tags used to attach the fragments
    private static final String TAG_HOME = "home";
    private static final String TAG_PHOTOS = "photos";
    private static final String TAG_MOVIES = "movies";
    private static final String TAG_NOTIFICATIONS = "notifications";
    private static final String TAG_SETTINGS = "settings";
    private static final String TAG_KNUTH="Knuth";
    public static String CURRENT_TAG = TAG_HOME;

    // toolbar titles respected to selected nav menu item
    private String[] activityTitles;

    // flag to load home fragment when user presses back key
    private boolean shouldLoadHomeFragOnBackPress = true;
    private Handler mHandler;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        mHandler = new Handler();

        drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        navigationView = (NavigationView) findViewById(R.id.nav_view);
       // fab = (FloatingActionButton) findViewById(R.id.fab);

        // Navigation view header
        navHeader = navigationView.getHeaderView(0);
        txtName = (TextView) navHeader.findViewById(R.id.name);
        txtWebsite = (TextView) navHeader.findViewById(R.id.website);
        imgNavHeaderBg = (ImageView)
navHeader.findViewById(R.id.img_header_bg);
        imgProfile = (ImageView) navHeader.findViewById(R.id.img_profile);
        // load toolbar titles from string resources
        activityTitles =
getResources().getStringArray(R.array.nav_item_activity_titles);

     /*   fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });*/

        // load nav menu header data
        loadNavHeader();

        // initializing navigation menu
        setUpNavigationView();

        if (savedInstanceState == null) {
            navItemIndex = 0;
            CURRENT_TAG = TAG_HOME;
            loadHomeFragment();
        }
    }

    /***
     * Load navigation menu header information
     * like background image, profile image
     * name, website, notifications action view (dot)
     */
    private void loadNavHeader() {
        // name, website
        txtName.setText("JIIT HUBDATES");
        txtWebsite.setText("www.jiit.ac.in");

        // loading header background image
        Glide.with(this).load(urlNavHeaderBg)
                .crossFade()
                .diskCacheStrategy(DiskCacheStrategy.ALL)
                .into(imgNavHeaderBg);

        // Loading profile image
        Glide.with(this).load(urlProfileImg)
                .crossFade()
                .thumbnail(0.5f)
                .bitmapTransform(new CircleTransform(this))
                .diskCacheStrategy(DiskCacheStrategy.ALL)
                .into(imgProfile);

        // showing dot next to notifications label
        navigationView.getMenu().getItem(3).setActionView(R.layout.menu_dot);
    }

    /***
     * Returns respected fragment that user
     * selected from navigation menu
     */
    private void loadHomeFragment() {
        // selecting appropriate nav menu item
        selectNavMenu();

        // set toolbar title
        setToolbarTitle();

        // if user select the current navigation menu again, don't do anything
        // just close the navigation drawer
        if
(getSupportFragmentManager().findFragmentByTag(CURRENT_TAG) != null)
{
            drawer.closeDrawers();

            // show or hide the fab button
       //    toggleFab();
            return;
        }

        // Sometimes, when fragment has huge data, screen seems hanging
        // when switching between navigation menus
        // So using runnable, the fragment is loaded with cross fade effect
        // This effect can be seen in GMail app
        Runnable mPendingRunnable = new Runnable() {
            @Override
            public void run() {
                // update the main content by replacing fragments
                Fragment fragment = getHomeFragment();
                FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
                fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
                        android.R.anim.fade_out);
                fragmentTransaction.replace(R.id.frame, fragment, CURRENT_TAG);
                fragmentTransaction.commitAllowingStateLoss();
            }
        };

        // If mPendingRunnable is not null, then add to the message queue
        if (mPendingRunnable != null) {
            mHandler.post(mPendingRunnable);
        }

        // show or hide the fab button
      //  toggleFab();

        //Closing drawer on item click
        drawer.closeDrawers();

        // refresh toolbar menu
        invalidateOptionsMenu();
    }

    private Fragment getHomeFragment() {
        switch (navItemIndex) {
            case 0:
                // home
                HomeFragment homeFragment = new HomeFragment();
                return homeFragment;
            case 1:
                // photos
                PhotosFragment photosFragment = new PhotosFragment();
                return photosFragment;
            case 2:
                // movies fragment
                MoviesFragment moviesFragment = new MoviesFragment();
                return moviesFragment;
            case 3:
                // notifications fragment
                NotificationsFragment notificationsFragment = new
NotificationsFragment();
                return notificationsFragment;
            case 4:
                // settings fragment
                SettingsFragment settingsFragment = new SettingsFragment();
                return settingsFragment;
            case 5:
                KnuthFragment knuthFragment=new KnuthFragment();
                return knuthFragment;
            default:
                return new HomeFragment();
        }
    }

    private void setToolbarTitle() {
        getSupportActionBar().setTitle(activityTitles[navItemIndex]);
    }

    private void selectNavMenu() {
        navigationView.getMenu().getItem(navItemIndex).setChecked(true);
    }

    private void setUpNavigationView() {
        //Setting Navigation View Item Selected Listener to handle the item click
of the navigation menu
        navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {

            // This method will trigger on item Click of navigation menu
            @Override
            public boolean onNavigationItemSelected(MenuItem menuItem) {

                //Check to see which item was being clicked and perform appropriate
action
                switch (menuItem.getItemId()) {
                    //Replacing the main content with ContentFragment Which is our
Inbox View;
                    case R.id.nav_home:
                        navItemIndex = 0;
                        CURRENT_TAG = TAG_HOME;
                        break;
                    case R.id.nav_photos:
                        navItemIndex = 1;
                        CURRENT_TAG = TAG_PHOTOS;
                        break;
                    case R.id.nav_movies:
                        navItemIndex = 2;
                        CURRENT_TAG = TAG_MOVIES;
                        break;
                    case R.id.nav_notifications:
                        navItemIndex = 3;
                        CURRENT_TAG = TAG_NOTIFICATIONS;
                        break;
                    case R.id.nav_settings:
                        navItemIndex = 4;
                        CURRENT_TAG = TAG_SETTINGS;
                        break;
                    case R.id.nav_knuth:
                        navItemIndex=5;
                        CURRENT_TAG=TAG_KNUTH;
                    case R.id.nav_about_us:
                        // launch new intent instead of loading fragment
                        startActivity(new Intent(MainActivity.this,
AboutUsActivity.class));
                        drawer.closeDrawers();
                        return true;
                    case R.id.nav_privacy_policy:
                        // launch new intent instead of loading fragment
                        startActivity(new Intent(MainActivity.this,
PrivacyPolicyActivity.class));
                        drawer.closeDrawers();
                        return true;
                    case R.id.nav_parola_hub:
                        startActivity(new Intent(MainActivity.this,ParolaActivity.class));
                    default:
                        navItemIndex = 0;
                }

                //Checking if the item is in checked state or not, if not make it in
checked state
                if (menuItem.isChecked()) {
                    menuItem.setChecked(false);
                } else {
                    menuItem.setChecked(true);
                }
                menuItem.setChecked(true);
                loadHomeFragment();

                return true;
            }
        });


        ActionBarDrawerToggle actionBarDrawerToggle = new
ActionBarDrawerToggle(this, drawer, toolbar, R.string.openDrawer,
R.string.closeDrawer) {

            @Override
            public void onDrawerClosed(View drawerView) {
                // Code here will be triggered once the drawer closes as we dont want
anything to happen so we leave this blank
                super.onDrawerClosed(drawerView);
            }

            @Override
            public void onDrawerOpened(View drawerView) {
                // Code here will be triggered once the drawer open as we dont want
anything to happen so we leave this blank
                super.onDrawerOpened(drawerView);
            }
        };

        //Setting the actionbarToggle to drawer layout
        drawer.addDrawerListener(actionBarDrawerToggle);

        //calling sync state is necessary or else your hamburger icon wont show up
        actionBarDrawerToggle.syncState();
    }

    @Override
    public void onBackPressed() {
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawers();
            return;
        }

        // This code loads home fragment when back key is pressed
        // when user is in other fragment than home
        if (shouldLoadHomeFragOnBackPress) {
            // checking if user is on other navigation menu
            // rather than home
            if (navItemIndex != 0) {
                navItemIndex = 0;
                CURRENT_TAG = TAG_HOME;
                loadHomeFragment();
                return;
            }
        }

        super.onBackPressed();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.

        // show menu only when home fragment is selected
        if (navItemIndex == 0) {
            getMenuInflater().inflate(R.menu.main, menu);
        }

        // when fragment is notifications, load the menu created for notifications
        if (navItemIndex == 3) {
            getMenuInflater().inflate(R.menu.notifications, menu);
        }
        return true;
    }
   /* @Override
    public void onPhotosFragmentInteraction(String string) {
        // Do stuff
    }
*/
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_logout) {
            Toast.makeText(getApplicationContext(), "Logout user!",
Toast.LENGTH_LONG).show();
            return true;
        }

        // user is in notifications fragment
        // and selected 'Mark all as Read'
        if (id == R.id.action_mark_all_read) {
            Toast.makeText(getApplicationContext(), "All notifications marked as
read!", Toast.LENGTH_LONG).show();
        }

        // user is in notifications fragment
        // and selected 'Clear All'
        if (id == R.id.action_clear_notifications) {
            Toast.makeText(getApplicationContext(), "Clear all notifications!",
Toast.LENGTH_LONG).show();
        }

        return super.onOptionsItemSelected(item);
    }

    // show or hide the fab
  //  private void toggleFab() {
  //      if (navItemIndex == 0)
   //         fab.show();
   //     else
   //         fab.hide();
   // }
}
```

SCREENSHOTS







12:30 PM

Harry potter

The perks of being a Wallflower

Kite Runner

Hunger games

To kill a mocking bird

DRacula untold

Lord of the rings

Hobbit

GOne girl

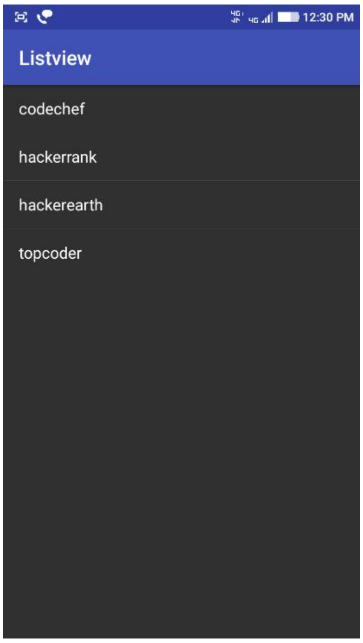
100 years of solitude

Lord of the rings-the two thrones

Prince of persia

Harry potter 7

fantastic beasts and where to find them



12:30 PM

Listview

codechef

hackerrank

hackerearth

topcoder

TESTING

Test Suite:

S.No	Code field	Input	Expected output	output
1.	Admin Login	Username(email id): mehak123@gmail.com Password: *1234*	Valid email id. Login accepted	Login successful(due to firebase's own login validations)
2.	Admin Login	Username(email id): mehak123@nothing.com Password: *1234*	Not a valid email id.	Login Unsuccessful.
3.	Adding Events	Name: Thespian orientation workshop Venue: LT-3, E block Date: 2-01-2017 Time: 5:30 PM	Event created	Event addition successful
4.	Adding Events	Name: GDG Workshop Venue: G-4 Date: 2-01-2017 Time:	Event created	Event addition successful (NULL field accepted)
5.	User Login (through Facebook)	Facebook Username/email id : shubhrag@gmail.com Password: ******	Valid email id	Login successful
6.	User login (through Facebook)	Facebook Username/email id : shubhra1234@gmail.com Password: ******	Username Incorrect	Login Unsuccessful

7.	Admin Login	Username: vivek9663@gmail.com Password:	Please enter the Password.	Login Unsuccessful when Password field is left blank.
8.	Admin Login	Username: Password:*****	Please enter the Username.	Login Unsuccessful when the Username field is left blank.

REFERENCES

- https://firebase.google.com/docs/database/admin/save-data
- http://www.android-examples.com/android-select-image-from-gallery-upload-to-server-example/
- www.androidhive.info
- http://www.vogella.com/tutorials/android.html
- https://developer.android.com/guide/topics/connectivity/index.html
- http://stackoverflow.com/questions/40131300/how-to-not-get-data-to-overwritten-in-firebase-and-store-an-attribute-as-a-list