

Cove: A Local Event Provider and Way Finder Application for Android Devices

A Capstone Project

Presented to the

Faculty of the College of Computer Studies

Lyceum of the Philippines University - Batangas

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Information Technology

Specialized in Multimedia Technologies

By:

Ervyn Christian S. Agoy

Mark Elo L. Dalangin

Joshua G. Grantoza

Kim Lemars A. Magnaye

Carl Adrian R. Sarrol

November 2020

Acknowledgement

The authors would like to acknowledge the following for making this study possible:

To our capstone adviser, Mrs. Maria Christina M. Ramos for guiding us in coming up with the research. To our professors, for giving us the knowledge and skills in developing this app. To our respective families for the inspiration, financial and moral support they have given all throughout the completion of the study.

Finally, to the Lord Almighty for the strength and wisdom to accomplish this work.

Ervyn Christian Agoy

Mark Elo Dalangin

Joshua Grantoza

Kim Lemars Magnaye

Carl Adrian Sarrol

Dedication

This humble piece of work is dedicated to the authors' teachers, classmates, friends, family, and the Lord Almighty.

Ervyn Christian Agoy

Mark Elo Dalangin

Joshua Grantoza

Kim Lemars Magnaye

Carl Adrian Sarrol

Cove: A Local Event Provider and Way Finder Application for Android Devices

Ervyn Christian S. Agoy

Lyceum of the Philippines University Batangas
Soro-soro Karsada, Batangas City
0956-804-8220
ervync23@gmail.com

Mark Elo L. Dalangin

Lyceum of the Philippines University Batangas
Sampaguita Country Homes, Gulod Itaas,
Batangas City
0977-741-2571
mrkdln0907@gmail.com

Joshua G. Grantoza

Lyceum of the Philippines University Batangas
Tampoy Gasang, Mabini Batangas
0999-193-1811
joshuagrantoza@gmail.com

Kim Lemars A. Magnaye

Lyceum of the Philippines University Batangas
Gamboa Rd, Kumintang Ibaba Batangas City
0917-921-0988
kimpoy114@gmail.com

Carl Adrian R. Sarrol

Lyceum of the Philippines University Batangas
Manghinao Proper, Bauan Batangas
0945-991-6626
callmynamecarl@gmail.com

ABSTRACT

The study utilized and determined the participant's attendance in public events using a mobile application. This can monitor the local public events by allowing the participant's device location access. The host of the specific public events will create and provide the information of the specific event and all those created in the app will be visible to every user. Every user can view the available local public events near in their specific location. Users has an option whether they want to join in their chosen public events. Users can also view where the local public events will be held by placing in the host the exact location and pinpointing in google maps the specific event location. If the user has joined their chosen public events, they could see the other participant's location in the maps that can be seen in the specific public events map page. Users have the option whether they want to see their specific location by turning on their location access in their own app. Users can also leave the specific public events by choosing to click the leave button in the events option menu. The public event host can easily monitor the participants' range by checking the participants' list in the app.

Keywords: *Local event finder, Mobile application for way finder, Mobile application for local public events*

APPROVAL SHEET

In partial fulfillment of the requirements for the degree Bachelor of Science in Information Technology (Specialized in Multimedia Technologies), this capstone project entitled “**Cove: A Local Event Provider and Way Finder Application for Android Devices**” has been prepared and submitted by **Ervyn Christian S. Agoy, Mark Elo L. Dalangin, Joshua G. Grantoza, Kim Lemars A. Magnaye** and **Carl Adrian R. Sarrol** and is hereby recommended for oral examination.

Maria Cristina M. Ramos, MSCS
Adviser

Defended in an oral examination before a duly constituted panel with a grade of

_____ .

Roselie B. Alday, MSCS, PhD Cand.
Chairman

Maria Julieta A. Saldua, MSIT
Member

Engr. Joselito A. Dolot, MSc
Member

Arnie Christian D. Villena, PhD
Member

Accepted in partial fulfillment of the requirements for the degree Bachelor of Science in Information Technology (Specialized in Multimedia Technologies).

Roselie B. Alday, MSCS, PhD Cand.
Dean, College of Computer Studies

TABLE OF CONTENTS

Title Page	i
Acknowledgement	ii
Dedication	iii
Abstract	iv
Approval Sheet.....	v
Table of Contents	vi
List of Figures	vii
1.0 INTRODUCTION.....	1
1.1 Objectives of the Study	1
2.0 LITERATURE REVIEW	1
2.1 Waze	1
2.2 Google Maps.....	2
2.3 FieldTrip	2
2.4 Doodle.....	2
2.5 Google News	2
3.0 METHODS	2
3.1 7 Phases of Mobile Application Development Lifecycle	2
4.0 RESULTS AND DISCUSSIONS	5
4.1 Flowchart	5
4.2 Screen Layouts	18
5.0 CONCLUSIONS AND RECOMENDATIONS	
5.1 Conclusions	23
5.2 Recommendations.....	23
REFERENCES.....	24
APPENDICES	25
Code Listing.....	25
Curriculum Vitae	49

LIST OF FIGURES

Figure		Page
1	Mobile App Development Lifecycle.....	2
2	Gantt Chart.....	4
3	Login and Sign up Screen (Cove).....	5
4	Main Menu (Cove).....	6
5	Settings Screen (Cove).....	7
6	Profile (Cove)	8
7	About (Cove).....	9
8	Event Joining and Information Page (Cove)	10
9	Event Creation Page (Cove)	11
10	Date Setting Page (Cove)	12
11	Time Setting Page (Cove)	13
12	Location Setting (Cove)	14
13	Event Map Location Setting (Cove)	15
14	Event Map Participants (Cove).....	16
15	About Event (Cove)	17
16	Login and Sign up Screen	18
17	Main Menu.....	18
18	Event Joining and Information Page	19
19	Event Creation Page.....	20
20	Date Setting	20
21	Time Setting.....	21
22	Location Setting.....	21
23	Event Location	22
24	Event Map Page.....	22

1.0 INTRODUCTION

Cove App aims to provide a mobile application for Local Public Events, and Public Organizations to help them easily provide directions and locations to their target members and audience for specific event and help them easily organize their plans.

Cove App will provide the ongoing, latest, and upcoming public events nearby with an event scheduler that can be used for many kinds of local public events such as Health, Entertainment, Education etc. Currently, some events tend to get a lot of late comers or even people that do not show up at all due to the inability to determine where the event location is or what the plans are.

Today, with the help of technology people do not need to use paper maps or ask for directions. This application will provide easier event planning and discussion with fellow users and will also have a GPS direction provider. With the increasing popularity of trendy mobile application, we develop Cove to experience the use of interactive maps and a chat room for social interaction.

The goal of Cove app is to provide convenience as regards to setting up a local public event

nearby where people can join in their chosen event. This app has GPS and internet connectivity limitations since most of its features require a stable internet whether Wi-Fi or Mobile Data.

Objectives of the Study

This study entitled, “Cove: A Local Event Provider and Way Finder” aimed to:

- develop a mobile application that will provide direction and location to the user and will keep track of their event and participant’s location with their consent.
- provide the whereabouts of the event participants using Google Maps, Network, and Mobile GPS connectivity and suggest a different type of local public event nearby where they can join.
- use Android Studio as a Mobile Application Development tool, Adobe Illustrator for icons and Adobe XD for layout designs.

2.0 LITERATURE REVIEW

Waze

The App encompasses a similar feature like Waze but rather than vehicles the researchers alter their users to people’s locations. Waze app

collects map data, travel times and traffic information from users and transmits it to the Waze server, at no cost. Waze can even identify the most affordable fuel station nearby or along the user's route, provided that Waze has enabled fuel prices for that country. On the other hand, the cove app can notify the members of the specific events if the members have gotten hold of their location

Google Maps

The Cove app developers integrated the Google Maps services, an internet mapping service developed by Google. It offers satellite imagery, aerial photography for the users to create certain locations for an event with precision.

FieldTrip

It also has the same features of FieldTrip app which acts as a "virtual tour guide" using the user's location to recommend nearby landmarks and various points of interest, providing information about them. The cove app also uses GPS and cellular signals like Wi-Fi or Mobile Data connectivity to see the user's location.

Doodle

The app has similar features with Doodle. This app can be a Swiss online calendar tool for time management and coordinating meetings. Unlike doodle, cove app encompasses a feature that users can view local events that near them.

Google News

The cove app encompasses a similar feature like Google News but rather than news gathered around the globe the cove app will display the various events nearby to the user's home/main screen while Google News contain a news aggregator app developed by Google that presents never-ending, customizable flow of articles organized from thousands of publishers and magazines worldwide. Google News app is available on Android, iOS, and the Web.

3.0 METHODS

In developing "Cove" the group applied the principles and techniques of the Mobile Development Lifecycle (MDLC). This process of building an App following an in depth one step at a time approach is termed Mobile App Development Lifecycle. It is a representation of the quality Software Development Lifecycle (SDLC) but from the angle of a mobile device.

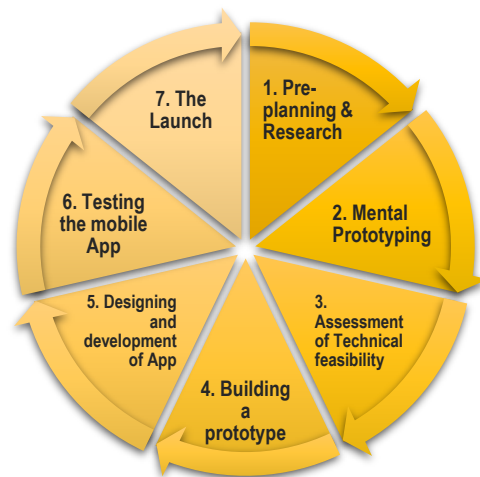


Figure 1. Mobile App Development Lifecycle

- **Pre-planning and research** – The primary phase was the foremost important. In here, the group did some brain storming where they gave their own ideas on what appropriate app should they develop. The ultimate aim of the app is to assist people find their way on a specific location where a public event is going to be held. The target market of the app are those individuals that are worried about visiting places they are not familiar with and provides ease on their way. It is intended for Android users which is free of charge in Google Play Store.
- **Mental Prototyping** – Once the group are through with the research and have laid down the cost involved in developing the app, they followed the next step which involves preparing a close scope of labor. They did a mental prototyping of the App and drew their ideas within the sort of visualizations and sketches on a paper. This was the primary visual representation of the group ideas which they collected in Phase 1 and helped them uncover usability issues.
- **Assessment of Technical Feasibility** - Knowing the visuals is not enough. Developers have to analyze whether the back-end systems will support the App's functionality or not. Hence, the group decided to form an app that is supported by Android Version 6.0 and above.
- **Building a prototype** - The group built a prototype using Adobe XD to determine the interface and outcome of the app and see where the buttons are to be placed and layouted for easy accessibility or use.
- **Designing and Development of App** - Before designing, the group decided to use appropriate colors for the users to make the program stand out. The software used in designing and developing the work is Adobe XD.
- **Testing the mobile App** - The members of the group tested the app by installing it on their own android devices and check if the application is functioning well. During this phase, the group manage to test errors on the app.
- **The Launch** – After testing for bugs and errors, the group created solution to make sure the app works.

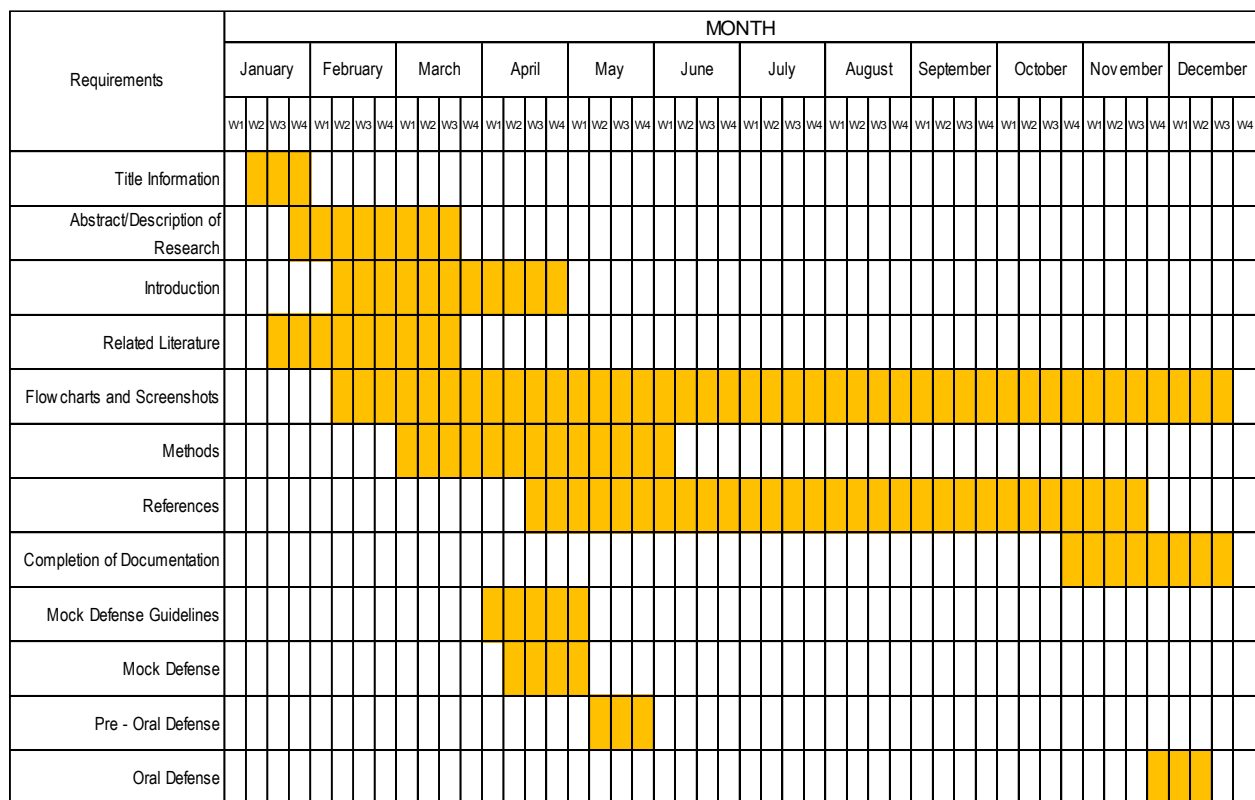


Figure 2. Gantt Chart

4.0 RESULTS AND DISCUSSIONS

Flowchart

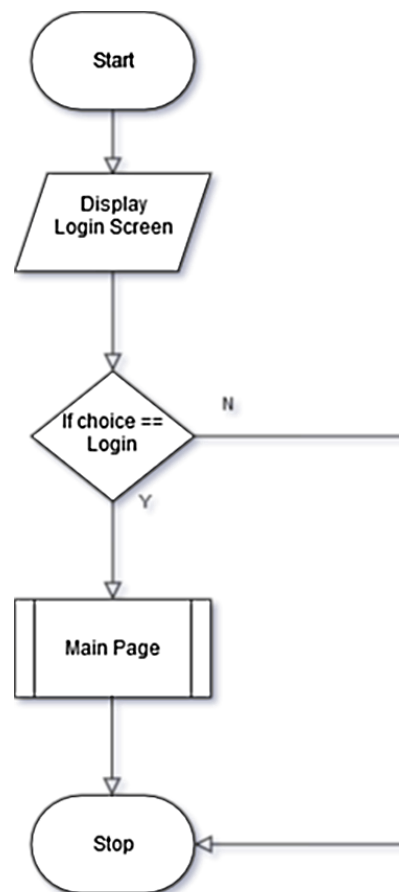


Figure 3. Login Screen (Cove)

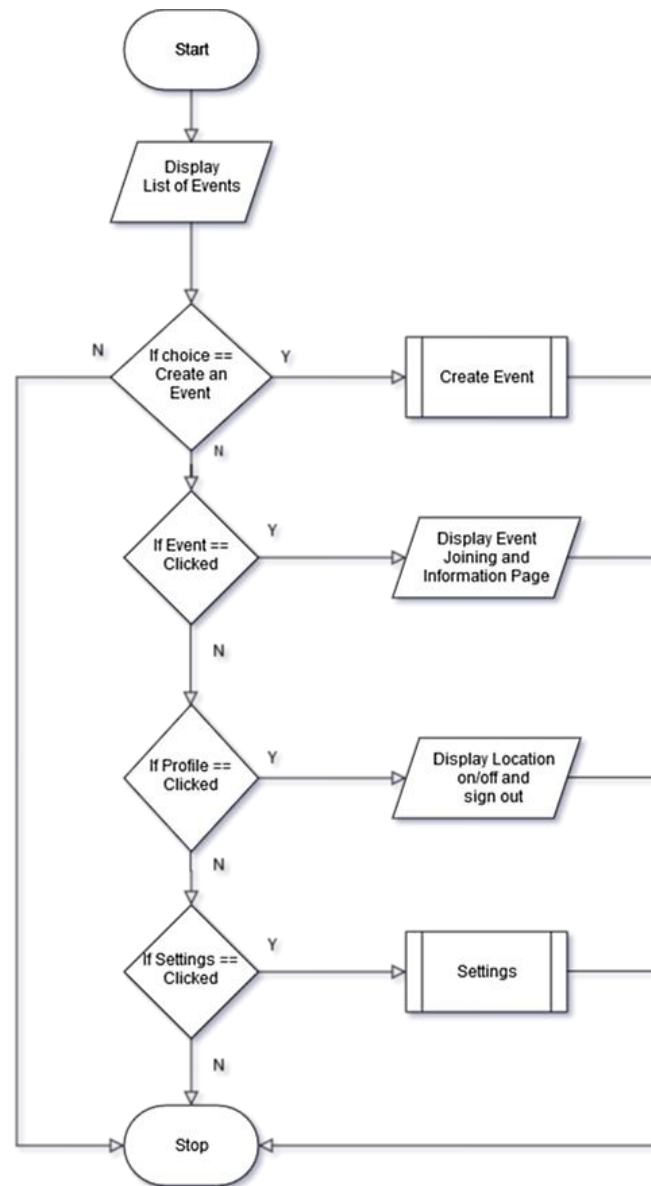


Figure 4. Main Menu Screen (Cove)

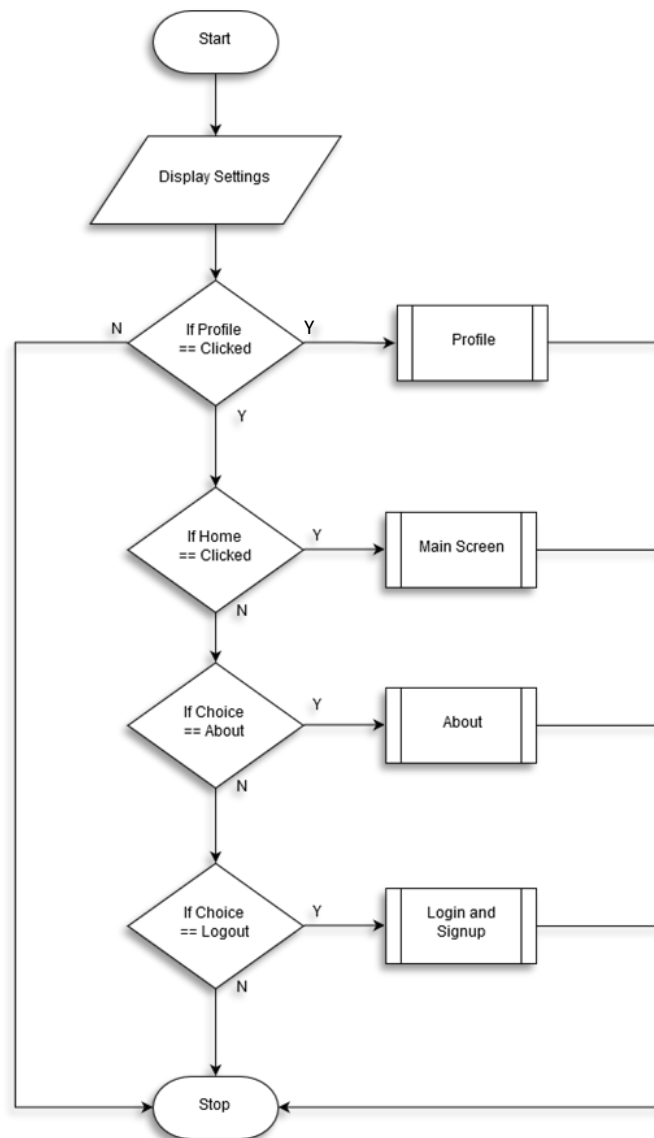


Figure 5. Settings Screen (Cove)

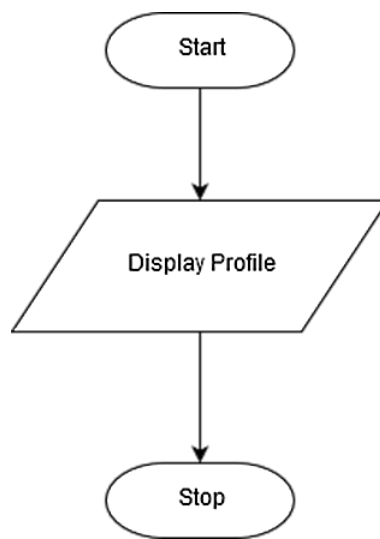


Figure 6. Profile (Cove)

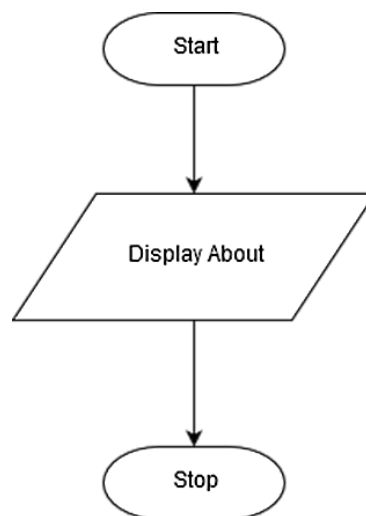


Figure 7. About (Cove)

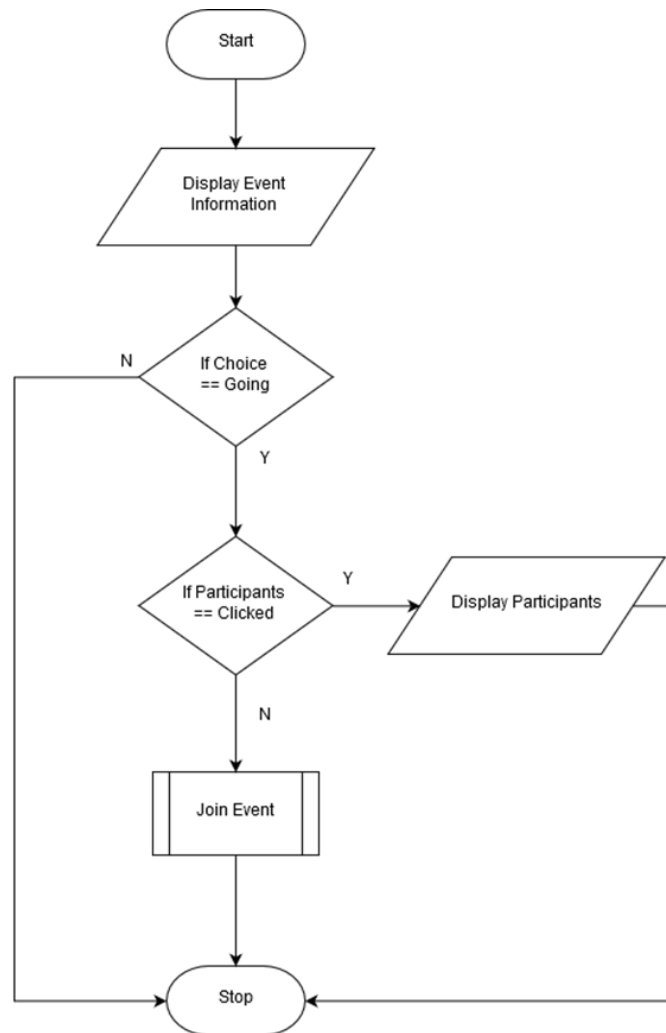


Figure 8. Event Joining and Information Page (Cove)

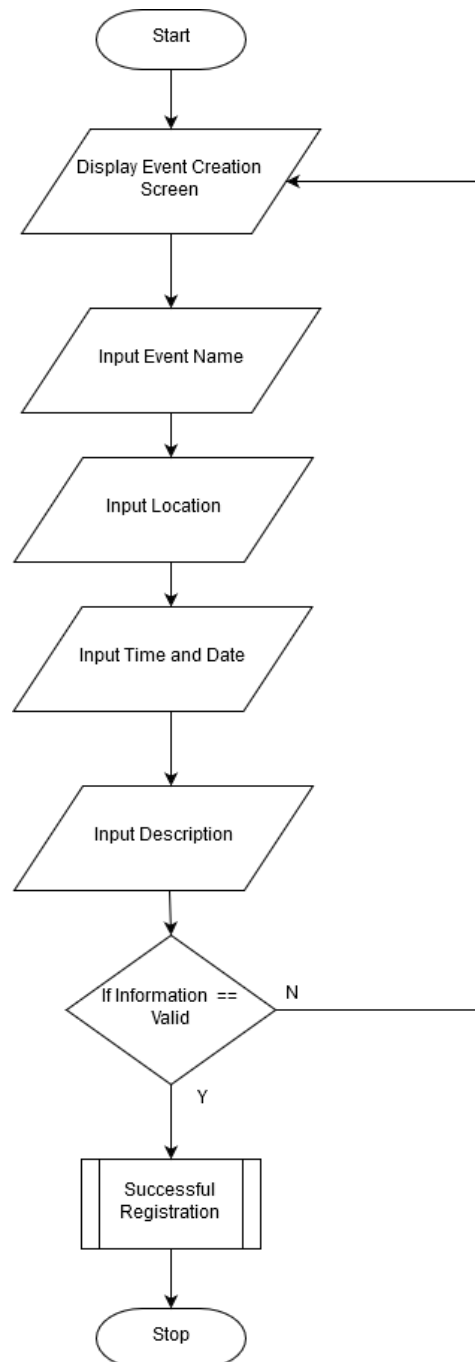


Figure 9. Event Creation Page (Cove)

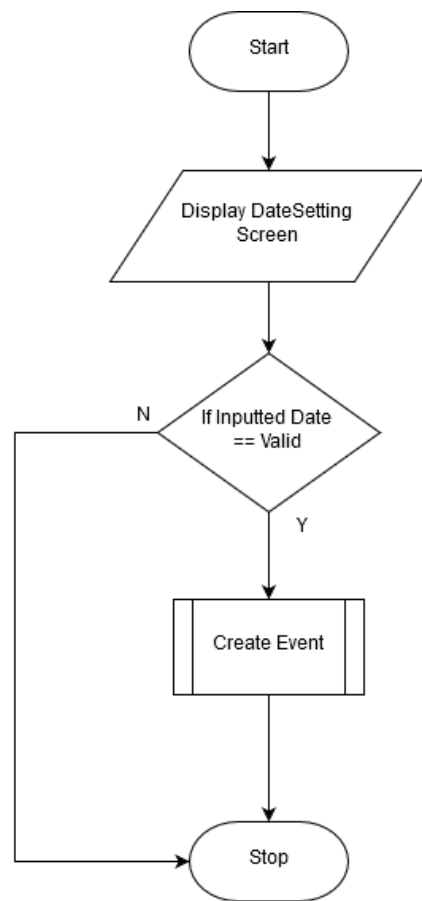


Figure 10. Date Setting (Cove)

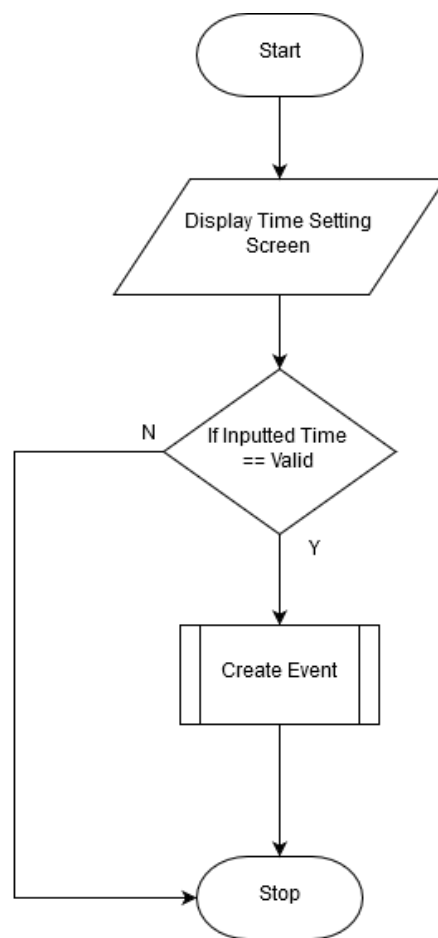


Figure 11. Time Setting (Cove)

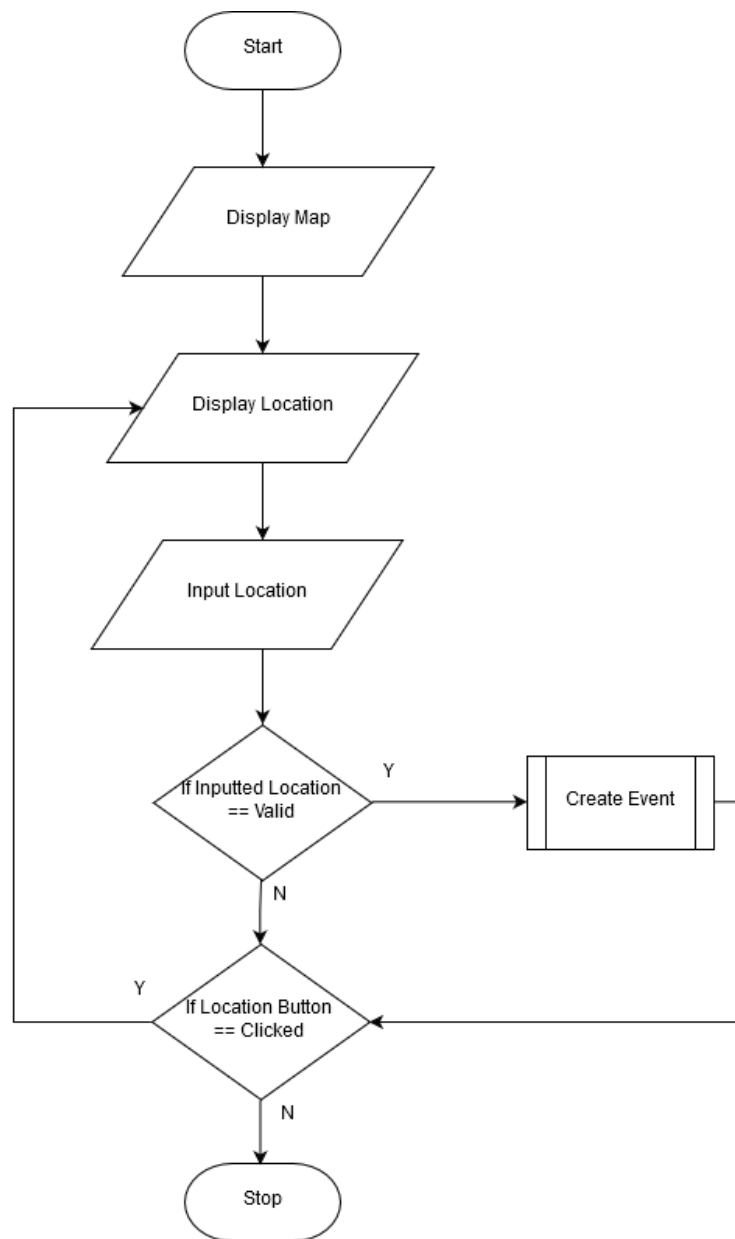


Figure 12. Location Setting (Cove)

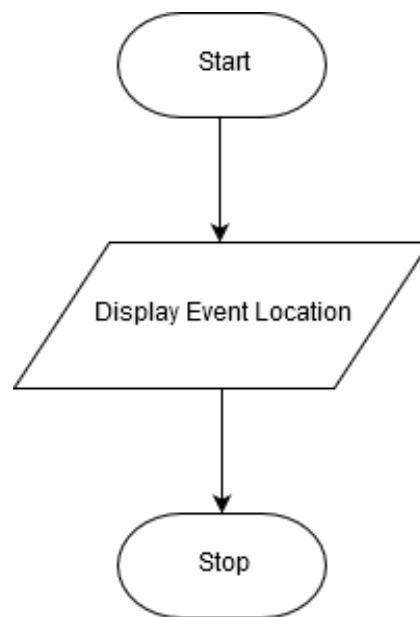


Figure 13. Event Map Location (Cove)

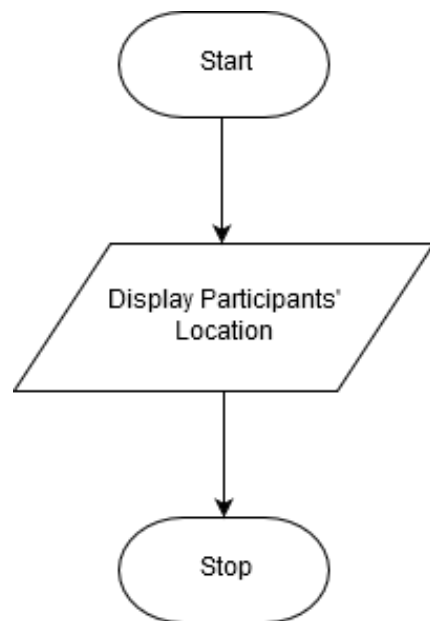


Figure 14. Event Map Participants Page (Cove)

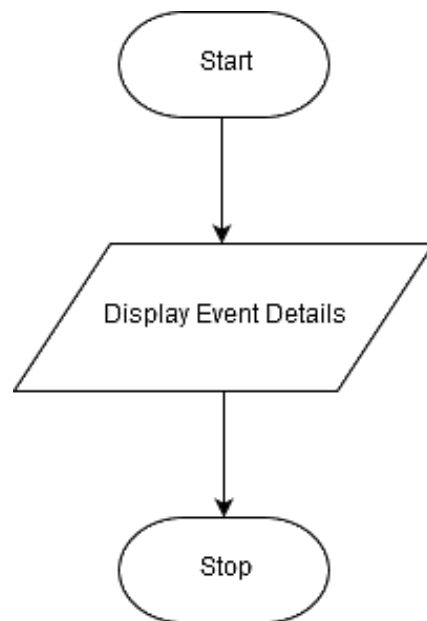


Figure 15. About Event (Cove)

Screen Layout



Figure 16. Login and Sign-up Screen

This is sample screenshot of the login and sign up screen of the app. A textbox of username and password will be prompted to the user. Logging in using Facebook Account is also included.

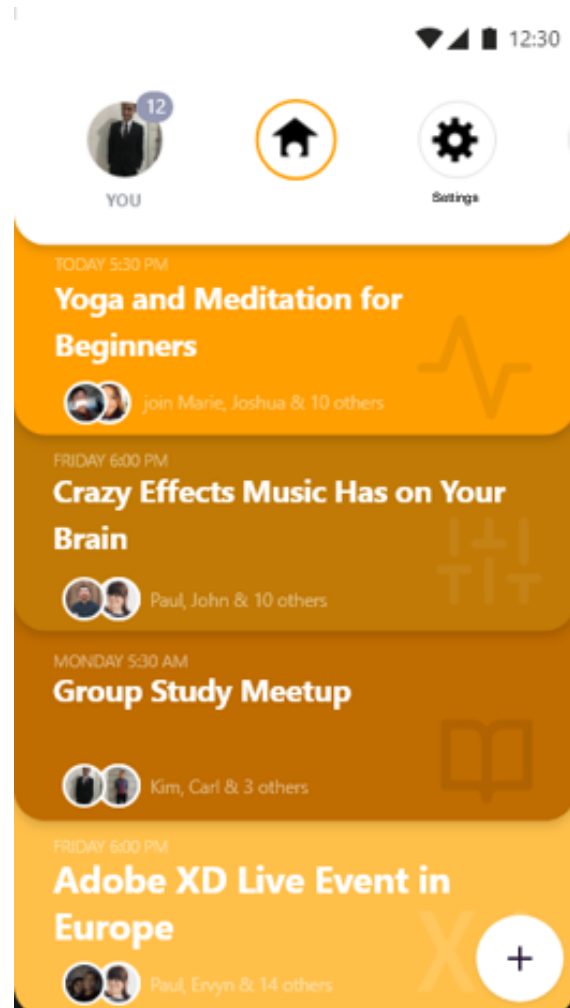


Figure 17. Main Menu

Once the user has successfully logged in, the list of events will be shown on the home screen including an option to create or join an event.

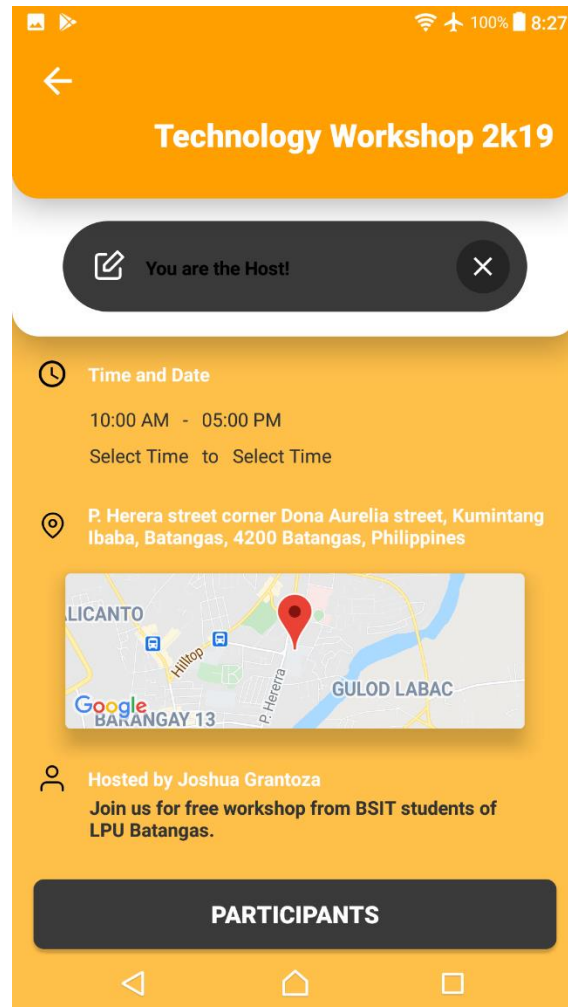
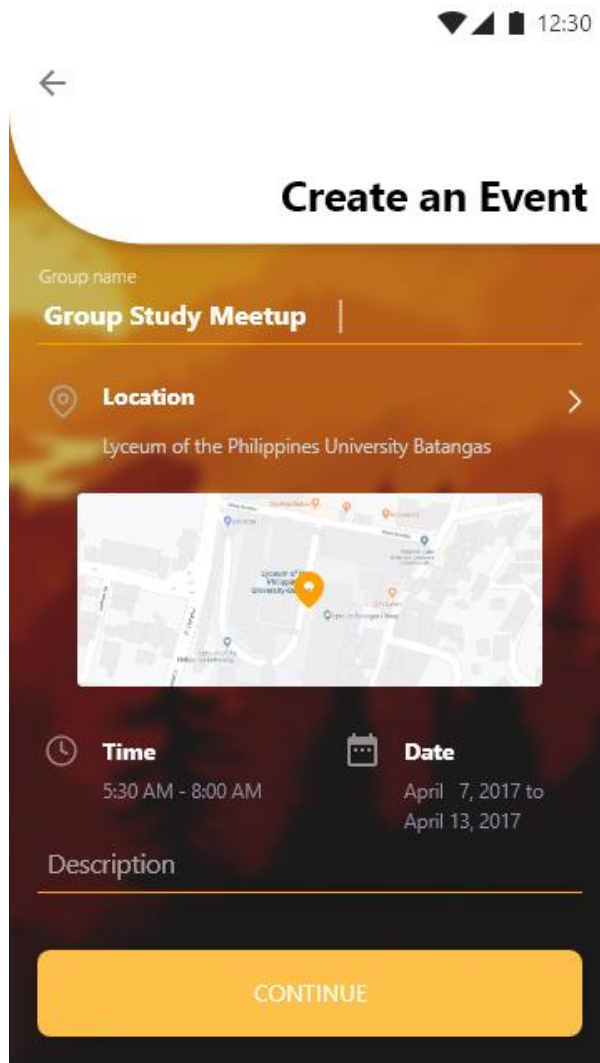


Figure 18. Event Joining and Information Page

After selecting an event, this screen will prompt on the screen to allow the user to decide if he/she is interested on the event. The location, time and date are also included on the details of the given event.



←

Create an Event

Group name
Group Study Meetup

Location
Lyceum of the Philippines University Batangas

Time
5:30 AM - 8:00 AM

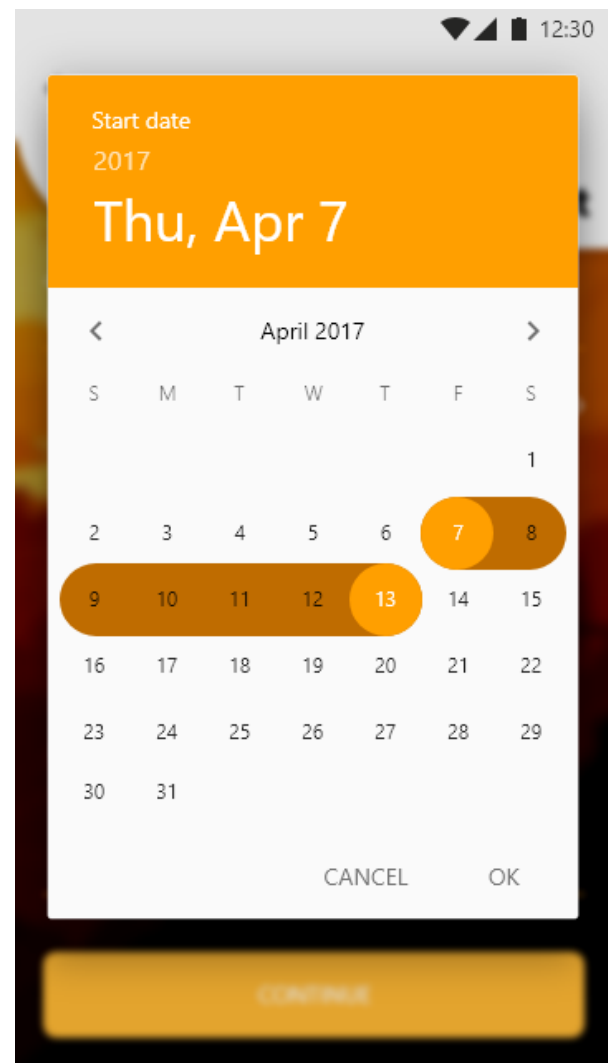
Date
April 7, 2017 to April 13, 2017

Description

CONTINUE

Figure 19. Event Creation Page

On creating an event, the given fields should be filled out including the location, description, time, and date.



Start date
2017
Thu, Apr 7

< April 2017 >

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
23	24	25	26	27	28	29
30	31					

CANCEL OK

CONTINUE

Figure 20. Date Setting

This shows how the user will set the date.

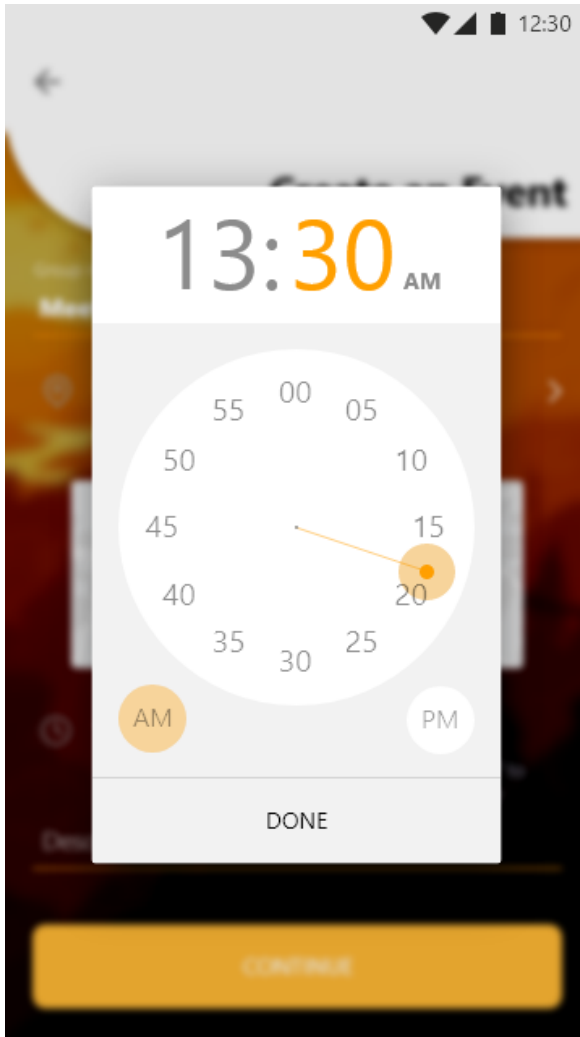


Figure 21. Time Setting

This shows how the user will set the date.



Figure 22. Location Setting

This shows where the user will set the location.

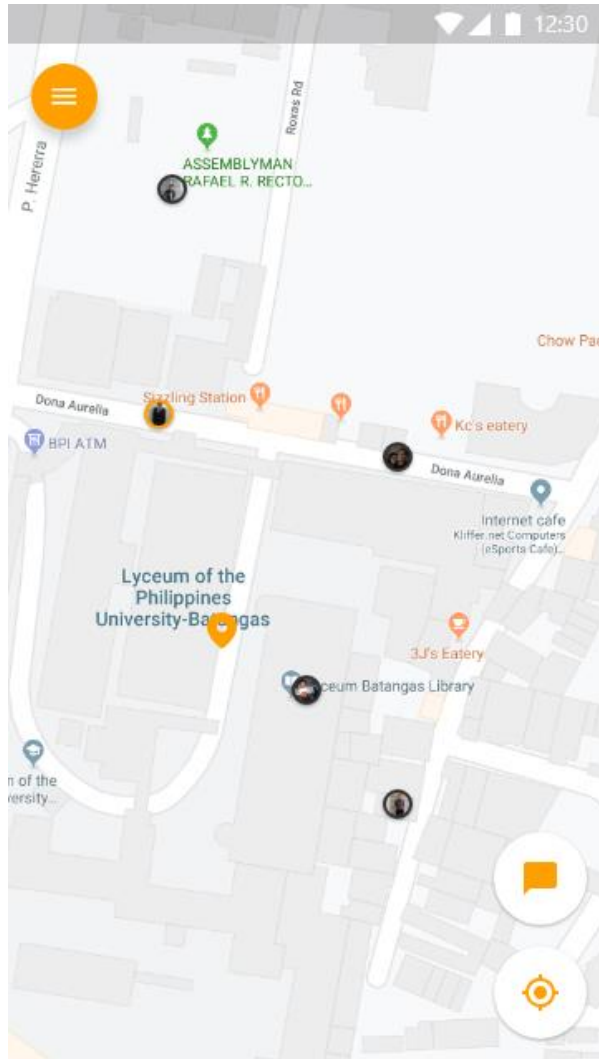


Figure 23. Event Location

This shows the location of the given event to the user.

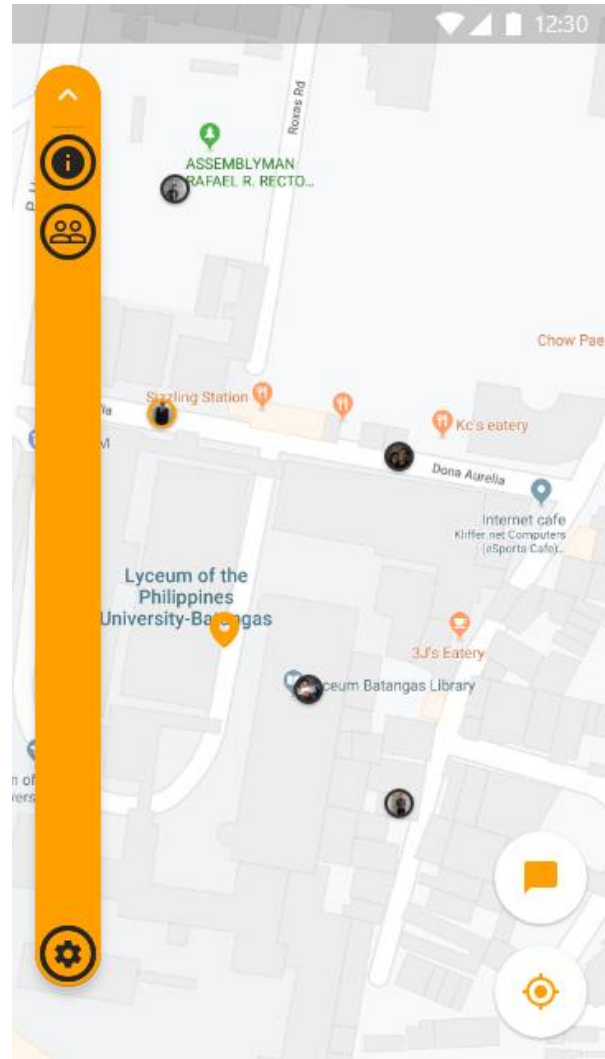


Figure 24. Event Map Page

This shows the location of the participants of the event.

5.0 CONCLUSIONS AND RECOMMENDATIONS:

Conclusions

The group accomplished the objectives of the app. The event host can keep track of their event participants' location since the participants location is displayed on the map. The identified tools are used in developing the application from designing, visualizing, editing, saving data and coding. As the user logged in, the list of events are prompted, and the event location are provided upon joining an event. Also, the Facebook account will be used to login and the list of events are displayed once the user successfully logged in. The participants are then displayed once the user joined an event and its details can also be viewed. Creating an event can be toggled by clicking the add button down below the screen. The participants can see their profile, about the application and logout on settings screen.

Recommendations

For future updates on the application, it is suggested that there should be added features for messaging, and privacy for specific events such as not displaying the event time, location, and date. Verification process for users should also be included for safety features. Pathfinder should be available to help users easily find the location of an event including GPS navigation and mapping app and the turn-by-turn voice navigation guidance. Downloading of maps offline should be included with real-time traffic information to help users find another way to avoid traffic.

REFERENCES

7 Phases of Mobile Development Lifecycle. (2018).
Retrieved from: 7 Phases of Mobile App
Development: <https://www.smallbizdaily.com/7-phases-mobile-app-development/>

Adobe XD. (2019). Retrieved from:
<https://www.adobe.com/sea/products/xd.html>
<https://theblog.adobe.com/xd-november-2019-update-coediting-more/>

Android Studio. (2014). Retrieved from: Android
Studio: <https://developer.android.com/studio/intro>,
https://en.wikipedia.org/wiki/Android_Studio

Doodle. (2018). Retrieved from:
<https://www.newlyswissed.com/doodle-scheduling-tool/>

Field Trip developed by Niantic Labs. (2012).
Retrieved from:
[https://en.wikipedia.org/wiki/Field_Trip_\(applicati
on\)](https://en.wikipedia.org/wiki/Field_Trip_(application))

Google News. (2002). Retrieved from:
https://en.wikipedia.org/wiki/Google_News

Google Maps owned by Google LLC. (2005).
Retrieved from: <https://www.google.com/maps>
https://en.wikipedia.org/wiki/Google_Maps

Waze owned by Google. (2006). Retrieve from:
<https://en.wikipedia.org/wiki/Waze>

APPENDICES

Code Listing

MainActivity

```
public class MainActivity extends AppCompatActivity {
    public static final String EXTRA_TEXT = "com.example.cove.EXTRA_TEXT";

    CircularImageView SettingsBtn;
    FloatingActionButton btn_CreateEvent;
    CircularImageView circularImageView;
    FirebaseRecyclerAdapter<Events, EventViewHolder> firebaseRecyclerAdapter;
    DatabaseReference imageUrl;
    LinearLayoutManager layoutManager;
    FirebaseAuth mAuth;
    DatabaseReference mDatabaseReference;
    RecyclerView mRecyclerView;
    List<Events> menuList;
    FirebaseRecyclerOptions<Events> options;

    private void loadImageUrl() {
        imageUrl.addValueEventListener(new ValueEventListener() {
            public void onCancelled(DatabaseError param1DatabaseError) {}
            public void onDataChange(DataSnapshot param1DataSnapshot) {
                for (DataSnapshot dataSnapshot : param1DataSnapshot.getChildren()) {
                    String str = ((UserProfile)param1DataSnapshot.getValue(UserProfile.class)).getImageUrl();
                    Picasso.get().load(str).into((ImageView)MainActivity.this.circularImageView);
                }
            }
        });
    }
    private void openCreateEvent() {
        startActivity(new Intent((Context)this, CreateEvent.class));
    }
    private void openProfile() {
        startActivity(new Intent((Context)this, ProfileActivity.class));
    }
    private void openSettings() {
        startActivity(new Intent((Context)this, SettingsActivity.class));
    }
    private void setupSystem() {
        this.mDatabaseReference.addValueEventListener(new ValueEventListener() {
            public void onCancelled(DatabaseError param1DatabaseError) {}

            public void onDataChange(DataSnapshot param1DataSnapshot) {
                MainActivity.this.menuList = new ArrayList<Events>();
                Iterator<DataSnapshot> iterator = param1DataSnapshot.getChildren().iterator();
                while (iterator.hasNext()) {
                    Events events = (Events)((DataSnapshot)iterator.next()).getValue(Events.class);
                    MainActivity.this.menuList.add(events);
                }
                MainActivity.this.firebaseRecyclerAdapter.notifyDataSetChanged();
                MainActivity.this.mRecyclerView.setAdapter((RecyclerView.Adapter)MainActivity.this.firebaseRecyclerAdapter);
            }
        });
    }
    private void updateList() {
        options = (new FirebaseRecyclerOptions.Builder()).setQuery((Query)this.mDatabaseReference, Events.class).build();
        FirebaseRecyclerAdapter<Events, EventViewHolder> firebaseRecyclerAdapter = new FirebaseRecyclerAdapter<Events,
        EventViewHolder>(this.options) {
```



```

        protected void onBindViewHolder(EventViewHolder param1EventViewHolder, int param1Int, final Events
events) {
    param1EventViewHolder.setDetails(MainActivity.this.getApplicationContext(), events.getName(),
events.getTimeFrom(), events.getTimeTo());
    int[] arrayOfInt = MainActivity.this.getResources().getIntArray(2130903040);
    param1EventViewHolder.constraintLayout.setBackgroundColor(arrayOfInt[param1Int]);
    param1EventViewHolder.itemView.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param2View) {
            String str = events.getName();
            MainActivity.this.mDatabaseReference.child(str).addListenerForSingleValueEvent(new ValueEventListener() {
                public void onCancelled(DatabaseError param3DatabaseError) {}
                public void onDataChange(DataSnapshot param3DataSnapshot) {
                    for (DataSnapshot dataSnapshot : param3DataSnapshot.getChildren()) {
                        if
(((Events)param3DataSnapshot.getValue(Events.class)).getHostUID().equals(FirebaseAuth.getInstance().getCurrentUser().getUi
d())) {
                            Intent intent1 = new Intent((Context)MainActivity.this, OpenEventHost.class);
                            intent1.putExtra("com.example.cove.EXTRA_TEXT", events.getName());
                            MainActivity.this.startActivity(intent1);
                            continue;
                        }
                        Intent intent = new Intent((Context)MainActivity.this, OpenEvent.class);
                        intent.putExtra("com.example.cove.EXTRA_TEXT", events.getName());
                        MainActivity.this.startActivity(intent);
                    }
                }
            });
        }
    });
}

    public EventViewHolder onCreateView(ViewGroup param1ViewGroup, int param1Int) {
        return new EventViewHolder(LayoutInflater.from(param1ViewGroup.getContext()).inflate(2131492926,
param1ViewGroup, false));
    }
};
this.firebaseRecyclerAdapter = firebaseRecyclerAdapter;
this.mRecyclerView.setAdapter((RecyclerView.Adapter)firebaseRecyclerAdapter);
this.mRecyclerView.setLayoutManager((RecyclerView.LayoutManager)this.layoutManager);
this.firebaseRecyclerAdapter.startListening();
this.firebaseRecyclerAdapter.notifyDataSetChanged();
}
protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(R.layout.activity_main);
    FloatingActionButton floatingActionButton = (FloatingActionButton)findViewById(R.id.btn_CreateEvent);
    btn_CreateEvent = floatingActionButton;
    floatingActionButton.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1View) {
            MainActivity.this.openCreateEvent();
        }
    });
    RecyclerView recyclerView = (RecyclerView)findViewById(R.id.eventRecyclerView);
    mRecyclerView = recyclerView;
    recyclerView.setHasFixedSize(true);
    LinearLayoutManager linearLayoutManager = new LinearLayoutManager((Context)this);
    layoutManager = linearLayoutManager;
    linearLayoutManager.setReverseLayout(false);
    layoutManager.setStackFromEnd(true);
    mRecyclerView.setLayoutManager((RecyclerView.LayoutManager)layoutManager);
    circularImageView = (CircularImageView)findViewById(R.id.profileImage);
    circularImageView.setOnClickListener(new View.OnClickListener() {

```

```

        public void onClick(View param1 View) {
            MainActivity.this.openProfile();
        }
    });
    SettingsBtn = (CircularImageView)findViewById(R.id.btn_Setting);
    circularImageView.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            MainActivity.this.openSettings();
        }
    });
    this mAuth = FirebaseAuth.getInstance();
    this.mDatabaseReference = FirebaseDatabase.getInstance().getReference().child("Events");
    this.imageUrl = FirebaseDatabase.getInstance().getReference("User
Profile").child(FirebaseAuth.getInstance().getCurrentUser().getUid());
    loadImageUrl();
    setupSystem();
    updateList();
    getWindow().setFlags(512, 512);
}
protected void onResume() {
    super.onResume();
    this.firebaseRecyclerAdapter.startListening();
}
protected void onStart() {
    super.onStart();
    setupSystem();
}
protected void onStop() {
    FirebaseRecyclerAdapter<Events, EventViewHolder> firebaseRecyclerAdapter = this.firebaseRecyclerAdapter;
    if (firebaseRecyclerAdapter != null)
        firebaseRecyclerAdapter.stopListening();
    super.onStop();
}
}
LoginActivity
class LoginActivity extends AppCompatActivity {
    private static final String TAG = "FACELOG";

    Button btn_login;
    Button btn_register;
    Button buttonFacebookLogin;
    EditText email1;
    String first_name, id, image_url, last_name;
    private FirebaseAuth mAuth;
    private CallbackManager mCallbackManager;
    EditText password1;
    DatabaseReference user_Profile;

    private void handleFacebookAccessToken(final AccessToken token) {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("handleFacebookAccessToken:");
        stringBuilder.append(token);
        Log.d("FACELOG", stringBuilder.toString());
        AuthCredential authCredential = FacebookAuthProvider.getCredential(token.getToken());
        this.mAuth.signInWithCredential(authCredential).addOnCompleteListener((Activity)this, new
        OnCompleteListener<AuthResult>() {
            public void onComplete(Task<AuthResult> param1Task) {
                GraphRequest graphRequest;
                if (param1Task.isSuccessful()) {
                    Log.d("FACELOG", "signInWithCredential:succes");
                    final FirebaseUser user = LoginActivity.this.mAuth.getCurrentUser();
                    LoginActivity.this.updateUI(firebaseUser);
                    graphRequest = GraphRequest.newMeRequest(token, new GraphRequest.GraphJSONObjectCallback() {

```

```

    public void onCompleted(JSONObject param2JSONObject, GraphResponse param2GraphResponse) {
        try {
            LoginActivity.this.first_name = param2JSONObject.getString("first_name");
            LoginActivity.this.last_name = param2JSONObject.getString("last_name");
            LoginActivity.this.id = param2JSONObject.getString("id");
            LoginActivity loginActivity = LoginActivity.this;
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("https://graph.facebook.com/");
            stringBuilder.append(LoginActivity.this.id);
            stringBuilder.append("/picture?type=normal");
            loginActivity.image_url = stringBuilder.toString();
            LoginActivity.this.user_Profile.child(user.getId()).setValue(new UserProfile(LoginActivity.this.first_name,
LoginActivity.this.last_name, LoginActivity.this.id, LoginActivity.this.image_url, Integer.valueOf(1)));
            return;
        } catch (JSONException jSONException) {
            jSONException.printStackTrace();
            return;
        }
    }
});
Bundle bundle = new Bundle();
bundle.putString("fields", "first_name,last_name,id");
graphRequest.setParameters(bundle);
graphRequest.executeAsync();
return;
}
Log.w("FACELOG", "signInWithCredential:failure", graphRequest.getException());
Toast.makeText((Context)LoginActivity.this, "Authentication failed.", 0).show();
LoginActivity.this.updateUI((FirebaseUser)null);
}
});
}
private void updateUI(FirebaseUser paramFirebaseUser) {
    startActivity(new Intent((Context)this, MainActivity.class));
    finish();
}
protected void onActivityResult(int paramInt1, int paramInt2, Intent paramIntIntent) {
    super.onActivityResult(paramInt1, paramInt2, paramIntIntent);
    this.mCallbackManager.onActivityResult(paramInt1, paramInt2, paramIntIntent);
}
protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131492895);
    this mAuth = FirebaseAuth.getInstance();
    this.password1 = (EditText)findViewById(2131296438);
    this.email1 = (EditText)findViewById(2131296436);
    this.user_Profile = FirebaseDatabase.getInstance().getReference("User Profile");
    this.mCallbackManager = CallbackManager.Factory.create();
    Button button = (Button)findViewById(2131296359);
    this.buttonFacebookLogin = button;
    button.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1View) {
            LoginManager.getInstance().loginWithReadPermissions((Activity)LoginActivity.this, Arrays.asList(new String[] {
"email", "public_profile" }));
            LoginManager.getInstance().registerCallback(LoginActivity.this.mCallbackManager, new
FacebookCallback<LoginResult>() {
                public void onCancel() {
                    Log.d("FACELOG", "facebook:onCancel");
                }
                public void onError(FacebookException param2FacebookException) {
                    Log.d("FACELOG", "facebook:onError", (Throwable)param2FacebookException);
                }
            }

```

```

        public void onSuccess(LoginResult param2LoginResult) {
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("facebook:onSuccess:");
            stringBuilder.append(param2LoginResult);
            Log.d("FACELOG", stringBuilder.toString());
            LoginActivity.this.handleFacebookAccessToken(param2LoginResult.getAccessToken());
            Toast.makeText(LoginActivity.this.getApplicationContext(), "Logging in...", 0).show();
        }
    });
}
});
getWindow().setFlags(512, 512);
}
public void onStart() {
    super.onStart();
    FirebaseUser firebaseUser = this.mAuth.getCurrentUser();
    if (firebaseUser != null)
        updateUI(firebaseUser);
}
}
}

```

Main Map

```

public class LoginActivity extends AppCompatActivity {
    private static final String TAG = "FACELOG";
    Button btn_login;
    Button btn_register;
    Button buttonFacebookLogin;
    EditText email1;
    String first_name;
    String id;
    String image_url;
    String last_name;
    private FirebaseAuth mAuth;
    private CallbackManager mCallbackManager;
    EditText password1;
    DatabaseReference user_Profile;

    private void handleFacebookAccessToken(final AccessToken token) {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("handleFacebookAccessToken:");
        stringBuilder.append(token);
        Log.d("FACELOG", stringBuilder.toString());
        AuthCredential authCredential = FacebookAuthProvider.getCredential(token.getToken());
        this.mAuth.signInWithCredential(authCredential).addOnCompleteListener((Activity)this, new
        OnCompleteListener<AuthResult>() {
            public void onComplete(Task<AuthResult> param1Task) {
                GraphRequest graphRequest;
                if (param1Task.isSuccessful()) {
                    Log.d("FACELOG", "signInWithCredential:success");
                    final FirebaseUser user = LoginActivity.this.mAuth.getCurrentUser();
                    LoginActivity.this.updateUI(firebaseUser);
                    graphRequest = GraphRequest.newMeRequest(token, new GraphRequest.GraphJSONObjectCallback() {
                        public void onCompleted(JSONObject param2JSONObject, GraphResponse param2GraphResponse) {
                            try {
                                LoginActivity.this.first_name = param2JSONObject.getString("first_name");
                                LoginActivity.this.last_name = param2JSONObject.getString("last_name");
                                LoginActivity.this.id = param2JSONObject.getString("id");
                                LoginActivity loginActivity = LoginActivity.this;
                                StringBuilder stringBuilder = new StringBuilder();
                                stringBuilder.append("https://graph.facebook.com/");
                                stringBuilder.append(LoginActivity.this.id);

```

```

        stringBuilder.append("/picture?type=normal");
        loginActivity.image_url = stringBuilder.toString();
        LoginActivity.this.user_Profile.child(user.getId()).setValue(new UserProfile(LoginActivity.this.first_name,
LoginActivity.this.last_name, LoginActivity.this.id, LoginActivity.this.image_url, Integer.valueOf(1)));
        return;
    } catch (JSONException jSONException) {
        jSONException.printStackTrace();
        return;
    }
}
});
Bundle bundle = new Bundle();
bundle.putString("fields", "first_name,last_name,id");
graphRequest.setParameters(bundle);
graphRequest.executeAsync();
return;
}
Log.w("FACELOG", "signInWithCredential:failure", graphRequest.getException());
Toast.makeText((Context)LoginActivity.this, "Authentication failed.", 0).show();
LoginActivity.this.updateUI((FirebaseUser)null);
}
});
}
private void updateUI(FirebaseUser paramFirebaseUser) {
    startActivity(new Intent((Context)this, MainActivity.class));
    finish();
}
protected void onActivityResult(int paramInt1, int paramInt2, Intent paramInt) {
    super.onActivityResult(paramInt1, paramInt2, paramInt);
    this.mCallbackManager.onActivityResult(paramInt1, paramInt2, paramInt);
}
protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131492895);
    this mAuth = FirebaseAuth.getInstance();
    this.password1 = (EditText)findViewById(2131296438);
    this.email1 = (EditText)findViewById(2131296436);
    this.user_Profile = FirebaseDatabase.getInstance().getReference("User Profile");
    this.mCallbackManager = CallbackManager.Factory.create();
    Button button = (Button)findViewById(2131296359);
    this.buttonFacebookLogin = button;
    button.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            LoginManager.getInstance().loginWithReadPermissions((Activity)LoginActivity.this, Arrays.asList(new String[] {
"email", "public_profile" }));
            LoginManager.getInstance().registerCallback(LoginActivity.this.mCallbackManager, new
FacebookCallback<LoginResult>() {
                public void onCancel() {
                    Log.d("FACELOG", "facebook:onCancel");
                }
                public void onError(FacebookException param2FacebookException) {
                    Log.d("FACELOG", "facebook:onError", (Throwable)param2FacebookException);
                }
                public void onSuccess(LoginResult param2LoginResult) {
                    StringBuilder stringBuilder = new StringBuilder();
                    stringBuilder.append("facebook:onSuccess:");
                    stringBuilder.append(param2LoginResult);
                    Log.d("FACELOG", stringBuilder.toString());
                    LoginActivity.this.handleFacebookAccessToken(param2LoginResult.getAccessToken());
                    Toast.makeText(LoginActivity.this.getApplicationContext(), "Logging in...", 0).show();
                }
            });
        }
    });
}

```

```

        });
    }
});
getWindow().setFlags(512, 512);
}
public void onStart() {
    super.onStart();
    FirebaseUser firebaseUser = this mAuth.getCurrentUser();
    if (firebaseUser != null)
        updateUI(firebaseUser);
}
}
}

```

OpenEvent

```

public class OpenEvent extends FragmentActivity implements OnMapReadyCallback {
    public static final String EXTRA_NAMEMAP = "com.example.cove.EXTRA_NAMEMAP";

    RelativeLayout Join;
    RelativeLayout Leave;
    ImageView back;
    RelativeLayout click;
    TextView dateOne, dateTwo, description, eventName, hostName;
    private DatabaseReference joinEvent;
    private double lat;
    private double lng;
    private DatabaseReference mDatabaseReference;
    private GoogleMap mMap;
    Button participants;
    public String text;
    TextView timeOne;
    TextView timeTwo;
    TextView txt_location;

    private void participantChecker() {
        this.mDatabaseReference.child("Participants").addListenerForSingleValueEvent(new ValueEventListener() {
            public void onCancelled(DatabaseError param1DatabaseError) {}
            public void onDataChange(DataSnapshot param1DataSnapshot) {
                if (param1DataSnapshot.hasChild(FirebaseAuth.getInstance().getCurrentUser().getUid())) {
                    OpenEvent.this.Join.setEnabled(false);
                    OpenEvent.this.Leave.setEnabled(true);
                    OpenEvent.this.Join.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099650));
                    OpenEvent.this.Leave.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099651));
                    OpenEvent.this.participants.setEnabled(true);
                    OpenEvent.this.txt_location.setEnabled(true);
                    OpenEvent.this.click.setEnabled(true);
                    return;
                }
                OpenEvent.this.Leave.setEnabled(false);
                OpenEvent.this.Join.setEnabled(true);
                OpenEvent.this.Leave.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099650));
                OpenEvent.this.Join.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099651));
                OpenEvent.this.participants.setEnabled(false);
                OpenEvent.this.txt_location.setEnabled(false);
                OpenEvent.this.click.setEnabled(false);
            }
        });
    }

    private void setupSystem() {
        this.mDatabaseReference.addValueEventListener(new ValueEventListener() {
            public void onCancelled(DatabaseError param1DatabaseError) {}
            public void onDataChange(DataSnapshot param1DataSnapshot) {

```

```

        for (DataSnapshot dataSnapshot : param1DataSnapshot.getChildren()) {
            Events events = (Events)param1DataSnapshot.getValue(Events.class);
            OpenEvent.access$102(OpenEvent.this, Double.valueOf(events.getLat()).doubleValue());
            OpenEvent.access$202(OpenEvent.this, Double.valueOf(events.getLng()).doubleValue());
            LatLng latLng = new LatLng(OpenEvent.this.lat, OpenEvent.this.lng);
            OpenEvent.this.mMap.addMarker((new MarkerOptions()).position(latLng));
            OpenEvent.this.mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 14.0F));
            OpenEvent.this.txt_location.setText(events.getLocationAddress());
            OpenEvent.this.timeOne.setText(events.getTimeFrom());
            OpenEvent.this.timeTwo.setText(events.getTimeTo());
            OpenEvent.this.dateOne.setText(events.getDateFrom());
            OpenEvent.this.dateTwo.setText(events.getDateTo());
            OpenEvent.this.description.setText(events.getDescription());
            TextView textView = OpenEvent.this.hostName;
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("Hosted by ");
            stringBuilder.append(events.getHost());
            textView.setText(stringBuilder.toString());
        }
    });
}

protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131492898);
    this.eventName = (TextView)findViewById(2131296260);
    this.txt_location = (TextView)findViewById(2131296739);
    this.timeOne = (TextView)findViewById(2131296269);
    this.timeTwo = (TextView)findViewById(2131296270);
    this.dateOne = (TextView)findViewById(2131296258);
    this.dateTwo = (TextView)findViewById(2131296259);
    this.description = (TextView)findViewById(2131296734);
    this.hostName = (TextView)findViewById(2131296738);
    this.participants = (Button)findViewById(2131296356);
    this.Join = (RelativeLayout)findViewById(2131296354);
    this.Leave = (RelativeLayout)findViewById(2131296355);
    this.click = (RelativeLayout)findViewById(2131296379);
    this.back = (ImageView)findViewById(2131296347);
    ((SupportMapFragment)getSupportFragmentManager().findFragmentById(2131296496)).getMapAsync(this);
    String str = getIntent().getStringExtra("com.example.cove.EXTRA_TEXT");
    this.text = str;
    this.eventName.setText(str);
    this.mDatabaseReference = FirebaseDatabase.getInstance().getReference().child("Events").child(this.text);
    this.joinEvent =
        FirebaseDatabase.getInstance().getReference().child("Events").child(this.text).child("Participants").child(FirebaseAuth.getInstance().getCurrentUser().getUid());
    participantChecker();
    this.back.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            Intent intent = new Intent((Context)OpenEvent.this, MainActivity.class);
            OpenEvent.this.startActivity(intent);
            OpenEvent.this.finish();
        }
    });
    this.txt_location.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            Intent intent = new Intent((Context)OpenEvent.this, Mainmap.class);
            intent.putExtra("com.example.cove.EXTRA_NAMEMAP", OpenEvent.this.text);
            OpenEvent.this.startActivity(intent);
        }
    });
    this.click.setOnClickListener(new View.OnClickListener() {

```

```

        public void onClick(View param1 View) {
            Intent intent = new Intent((Context)OpenEvent.this, Mainmap.class);
            intent.putExtra("com.example.cove.EXTRA_NAMEMAP", OpenEvent.this.text);
            OpenEvent.this.startActivity(intent);
        }
    });
    this.participants.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            Intent intent = new Intent((Context)OpenEvent.this, ListActivity.class);
            intent.putExtra("com.example.cove.EXTRA_NAMEMAP", OpenEvent.this.text);
            OpenEvent.this.startActivity(intent);
        }
    });
    this.Join.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            OpenEvent.this.Join.setEnabled(false);
            OpenEvent.this.Leave.setEnabled(true);
            OpenEvent.this.Join.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099650));
            OpenEvent.this.Leave.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099651));
            OpenEvent.this.participants.setEnabled(true);
            OpenEvent.this.txt_location.setEnabled(true);
            OpenEvent.this.click.setEnabled(true);
            OpenEvent.this.joinEvent.setValue(new User(FirebaseAuth.getInstance().getCurrentUser().getEmail(), "Online",
            FirebaseAuth.getInstance().getCurrentUser().getUid(), "joined"));
        }
    });
    this.Leave.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1 View) {
            OpenEvent.this.Leave.setEnabled(false);
            OpenEvent.this.Join.setEnabled(true);
            OpenEvent.this.Leave.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099650));
            OpenEvent.this.Join.setBackgroundTintList(OpenEvent.this.getResources().getColorStateList(2131099651));
            OpenEvent.this.participants.setEnabled(false);
            OpenEvent.this.txt_location.setEnabled(false);
            OpenEvent.this.click.setEnabled(false);
            OpenEvent.this.joinEvent.removeValue();
        }
    });
    getWindow().setFlags(512, 512);
}

public void onMapReady(GoogleMap paramGoogleMap) {
    setupSystem();
    this.mMap = paramGoogleMap;
    paramGoogleMap.getUiSettings().setAllGesturesEnabled(false);
    this.mMap.getUiSettings().setMapToolbarEnabled(false);
}
}

```

ProfileActivity

```

public class ProfileActivity extends AppCompatActivity {
    public static final String EXTRA_TEXT = "com.example.cove.EXTRA_TEXT";

    DatabaseReference allowLocation;
    Switch btn_location;
    CircularImageView btn_sign_out;
    CircularImageView circularImageView;
    FirebaseRecyclerAdapter<Events, EventViewHolder> firebaseRecyclerAdapter;
    DatabaseReference imageUrl;
    LinearLayoutManager layoutManager;
    FirebaseAuth mAuth;
    FloatingActionButton mBack;

```



```

DatabaseReference mDatabaseReference;
RecyclerView mRecyclerView;
List<Events> menuList;
FirebaseRecyclerOptions<Events> options;
DatabaseReference profileName;
TextView textView_profileName;

private void backToCreate() {
    startActivity(new Intent((Context)this, MainActivity.class));
}
private void loadAllowLocation() {
    this.allowLocation.addValueEventListener(new ValueEventListener() {
        public void onCancelled(DatabaseError param1DatabaseError) {}
        public void onDataChange(DataSnapshot param1DataSnapshot) {
            for (DataSnapshot dataSnapshot : param1DataSnapshot.getChildren()) {
                if (((UserProfile)param1DataSnapshot.getValue(UserProfile.class)).getAllowLocation().intValue() == 1) {
                    ProfileActivity.this.btn_location.setChecked(true);
                } else {
                    ProfileActivity.this.btn_location.setChecked(false);
                }
            }
            ProfileActivity.this.btn_location.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
                public void onCheckedChanged(CompoundButton param2CompoundButton, boolean param2Boolean) {
                    if (param2Boolean) { ProfileActivity.this.allowLocation.child("allowLocation").setValue(Integer.valueOf(1));
                        return;
                    }
                    ProfileActivity.this.allowLocation.child("allowLocation").setValue(Integer.valueOf(0));
                }
            });
        }
    });
}
private void loadImageUrl() {
    this.imageUrl.addValueEventListener(new ValueEventListener() {
        public void onCancelled(DatabaseError param1DatabaseError) {}
        public void onDataChange(DataSnapshot param1DataSnapshot) {
            for (DataSnapshot dataSnapshot : param1DataSnapshot.getChildren()) {
                String str = ((UserProfile)param1DataSnapshot.getValue(UserProfile.class)).getImageUrl();
                Picasso.get().load(str).into((ImageView)ProfileActivity.this.circularImageView);
            }
        }
    });
}
private void loadProfileName() {
    this.profileName.addValueEventListener(new ValueEventListener() {
        public void onCancelled(DatabaseError param1DatabaseError) {}
        public void onDataChange(DataSnapshot param1DataSnapshot) {
            for (DataSnapshot dataSnapshot : param1DataSnapshot.getChildren()) {
                UserProfile userProfile = (UserProfile)param1DataSnapshot.getValue(UserProfile.class);
                String str1 = userProfile.getFirstName();
                String str2 = userProfile.getLastName();
                TextView textView = ProfileActivity.this.textView_profileName;
                StringBuilder stringBuilder = new StringBuilder();
                stringBuilder.append(str1);
                stringBuilder.append(" ");
                stringBuilder.append(str2);
                textView.setText(stringBuilder.toString());
            }
        }
    });
}
private void setupSystem() {

```

```

this.mDatabaseReference.addValueEventListener(new ValueEventListener() {
    public void onCancelled(DatabaseError param1DatabaseError) {}
    public void onDataChange(DataSnapshot param1DataSnapshot) {
        ProfileActivity.this.menuList = new ArrayList<Events>();
        Iterator<DataSnapshot> iterator = param1DataSnapshot.getChildren().iterator();
        while (iterator.hasNext()) {
            Events events = (Events)((DataSnapshot)iterator.next()).getValue(Events.class);
            ProfileActivity.this.menuList.add(events);
        }
        ProfileActivity.this.firebaseRecyclerAdapter.notifyDataSetChanged();
    }
});
ProfileActivity.this.mRecyclerView.setAdapter((RecyclerView.Adapter)ProfileActivity.this.firebaseRecyclerAdapter);
}
private void updateList() {
    this.options = (new FirebaseRecyclerOptions.Builder()).setQuery((Query)this.mDatabaseReference, Events.class).build();
    FirebaseRecyclerAdapter<Events, EventViewHolder> firebaseRecyclerAdapter = new FirebaseRecyclerAdapter<Events,
EventViewHolder>(this.options) {
        protected void onBindViewHolder(EventViewHolder param1EventViewHolder, int param1Int, final Events events) {
            param1EventViewHolder.setDetails(ProfileActivity.this.getApplicationContext(), events.getName(),
events.getTimeFrom(), events.getTimeTo());
            int[] arrayOfInt = ProfileActivity.this.getResources().getIntArray(2130903040);
            param1EventViewHolder.constraintLayout.setBackgroundColor(arrayOfInt[param1Int]);
            param1EventViewHolder.itemView.setOnClickListener(new View.OnClickListener() {
                public void onClick(View param2View) {
                    String str = events.getName();
                    ProfileActivity.this.mDatabaseReference.child(str).addValueEventListener(new ValueEventListener() {
                        public void onCancelled(DatabaseError param3DatabaseError) {}

                        public void onDataChange(DataSnapshot param3DataSnapshot) {
                            for (DataSnapshot dataSnapshot : param3DataSnapshot.getChildren()) {
                                if
(((Events)param3DataSnapshot.getValue(Events.class)).getHostUID().equals(FirebaseAuth.getInstance().getCurrentUser().getUi
d())) {
                                    Intent intent1 = new Intent((Context)ProfileActivity.this, OpenEventHost.class);
                                    intent1.putExtra("com.example.cove.EXTRA_TEXT", events.getName());
                                    ProfileActivity.this.startActivity(intent1);
                                    continue;
                                }
                                Intent intent = new Intent((Context)ProfileActivity.this, OpenEvent.class);
                                intent.putExtra("com.example.cove.EXTRA_TEXT", events.getName());
                                ProfileActivity.this.startActivity(intent);
                            }
                        }
                    });
                }
            });
        }
    };
    public EventViewHolder onCreateView(ViewGroup param1ViewGroup, int param1Int) {
        return new EventViewHolder(LayoutInflater.from(param1ViewGroup.getContext()).inflate(2131492926,
param1ViewGroup, false));
    }
};
this.firebaseRecyclerAdapter = firebaseRecyclerAdapter;
this.mRecyclerView.setAdapter((RecyclerView.Adapter)firebaseRecyclerAdapter);
this.mRecyclerView.setLayoutManager((RecyclerView.LayoutManager)this.layoutManager);
this.firebaseRecyclerAdapter.startListening();
this.firebaseRecyclerAdapter.notifyDataSetChanged();
}
private void updateUI(FirebaseUser paramFirebaseUser) {
    startActivity(new Intent((Context)this, LoginActivity.class));
    finish();
}

```

```

    }
    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2131492900);
        RecyclerView recyclerView = (RecyclerView)findViewById(2131296440);
        this.mRecyclerView = recyclerView;
        recyclerView.setHasFixedSize(true);
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager((Context)this);
        this.layoutManager = linearLayoutManager;
        linearLayoutManager.setReverseLayout(false);
        this.layoutManager.setStackFromEnd(true);
        this.mRecyclerView.setLayoutManager((RecyclerView.LayoutManager)this.layoutManager);
        this.circularImageView = (CircularImageView)findViewById(2131296577);
        this.textView_profileName = (TextView)findViewById(2131296578);
        CircularImageView circularImageView = (CircularImageView)findViewById(2131296357);
        this.btn_sign_out = circularImageView;
        circularImageView.setOnClickListener(new View.OnClickListener() {
            public void onClick(View param1 View) {
                ProfileActivity.this mAuth.signOut();
                LoginManager.getInstance().logout();
                FirebaseAuth firebaseUser = ProfileActivity.this mAuth.getCurrentUser();
                ProfileActivity.this.updateUI(firebaseUser);
            }
        });
        this.btn_location = (Switch)findViewById(2131296352);
        this mAuth = FirebaseAuth.getInstance();
        this.mDatabaseReference = FirebaseDatabase.getInstance().getReference().child("Events");
        this.imageUrl = FirebaseDatabase.getInstance().getReference("User
Profile").child(FirebaseAuth.getInstance().getCurrentUser().getUid());
        this.profileName = FirebaseDatabase.getInstance().getReference("User
Profile").child(FirebaseAuth.getInstance().getCurrentUser().getUid());
        this.allowLocation = FirebaseDatabase.getInstance().getReference("User
Profile").child(FirebaseAuth.getInstance().getCurrentUser().getUid());
        loadImageUrl();
        loadProfileName();
        loadAllowLocation();
        setupSystem();
        updateList();
        getWindow().setFlags(512, 512);
    }
    protected void onResume() {
        super.onResume();
        this.firebaseRecyclerAdapter.startListening();
    }
    protected void onStart() {
        super.onStart();
        setupSystem();
    }
    protected void onStop() {
        FirebaseAuth firebaseRecyclerAdapter<Events, EventViewHolder> firebaseRecyclerAdapter = this.firebaseRecyclerAdapter;
        if (firebaseRecyclerAdapter != null)
            firebaseRecyclerAdapter.stopListening();
        super.onStop();
    }
}

```

select_location

```

public class select_location extends AppCompatActivity implements OnMapReadyCallback,
GoogleApiClient.OnConnectionFailedListener {
    private static final String COUSE_LOCATION = "android.permission.ACCESS_COARSE_LOCATION";
    private static final float DEFAULT_ZOOM = 15.0F;
    public static final String EXTRA_LAT = "com.example.cove.EXTRA_LAT";

```

```

public static final String EXTRA_LNG = "com.example.cove.EXTRA_LNG";
public static final String EXTRA_TEXT = "com.example.cove.EXTRA_TEXT";
private static final String FINE_LOCATION = "android.permission.ACCESS_FINE_LOCATION";

private static final LatLngBounds LAT_LNG_BOUNDS = new LatLngBounds(new LatLng(-40.0D, -168.0D), new
LatLng(71.0D, 136.0D));
private static final int LOCATION_PERMISSION_REQUEST_CODE = 1234;
private FloatingActionButton mBack;
private FusedLocationProviderClient mFusedLocationProviderClient;
private GoogleApiClient mGoogleApiClient;
private FloatingActionButton mGps;
private Boolean mLocationPermissionGranted = Boolean.valueOf(false);
private GoogleMap mMap;
private AutoCompleteTextView mSearchText;

private void backToCreate() {
    String str = this.mSearchText.getText().toString();
    Geocoder geocoder = new Geocoder((Context)this);
    List<Address> list = new ArrayList();
    try {
        List<Address> list1 = geocoder.getFromLocationName(str, 1);
        list = list1;
    } catch (IOException iOException) {}
    if (list.size() > 0) {
        Address address = list.get(0);
        double d1 = address.getLatitude();
        double d2 = address.getLongitude();
        Intent intent = new Intent((Context)this, CreateEvent.class);
        intent.putExtra("Check", 1);
        intent.putExtra("com.example.cove.EXTRA_TEXT", address.getAddressLine(0));
        intent.putExtra("com.example.cove.EXTRA_LAT", d1);
        intent.putExtra("com.example.cove.EXTRA_LNG", d2);
        startActivity(intent);
        finish();
    }
}

private void geoLocate() {
    String str = this.mSearchText.getText().toString();
    Geocoder geocoder = new Geocoder((Context)this);
    List<Address> list = new ArrayList();
    try {
        List<Address> list1 = geocoder.getFromLocationName(str, 1);
        list = list1;
    } catch (IOException iOException) {}
    if (list.size() > 0) {
        Address address = list.get(0);
        moveCamera(new LatLng(address.getLatitude(), address.getLongitude()), 15.0F, address.getAddressLine(0));
    }
}

private void getDeviceLocation() {
    this.mFusedLocationProviderClient = LocationServices.getFusedLocationProviderClient((Activity)this);
    try {
        if (this.mLocationPermissionGranted.booleanValue())
            this.mFusedLocationProviderClient.getLastLocation().addOnCompleteListener(new OnCompleteListener() {
                public void onComplete(Task param1Task) {
                    if (param1Task.isSuccessful()) {
                        Location location = (Location)param1Task.getResult();
                        select_location.this.moveCamera(new LatLng(location.getLatitude(), location.getLongitude()), 15.0F, "My
Location");
                    }
                    return;
                }
            });
        Toast.makeText((Context)select_location.this, "unable to get current location", 0).show();
    }
}

```

```

        }
    });
    return;
} catch (SecurityException securityException) {
    return;
}
}
}
private void getLocationPermission() {
    String[] arrayOfString = new String[2];
    arrayOfString[0] = "android.permission.ACCESS_FINE_LOCATION";
    arrayOfString[1] = "android.permission.ACCESS_COARSE_LOCATION";
    if (ContextCompat.checkSelfPermission(getApplicationContext(), "android.permission.ACCESS_FINE_LOCATION") ==
0) {
        if (ContextCompat.checkSelfPermission(getApplicationContext(),
"android.permission.ACCESS_COARSE_LOCATION") == 0) {
            this.mLocationPermissionGranted = Boolean.valueOf(true);
            initMap();
            return;
        }
        ActivityCompat.requestPermissions((Activity)this, arrayOfString, 1234);
        return;
    }
    ActivityCompat.requestPermissions((Activity)this, arrayOfString, 1234);
}
private void hideSoftKeyboard() {
    getWindow().setSoftInputMode(3);
}
private void init() {
    this.mGoogleApiClient = (new
GoogleApiClient.Builder((Context)this).addApi(Places.GEO_DATA_API).addApi(Places.PLACE_DETECTION_API).enable
AutoManage((FragmentActivity)this, this).build();
    this.mSearchText.setOnEditorActionListener(new TextView.OnEditorActionListener() {
        public boolean onEditorAction(TextView param1TextView, int param1Int, KeyEvent param1KeyEvent) {
            if (param1Int == 3 || param1Int == 6 || param1KeyEvent.getAction() == 0 || param1KeyEvent.getAction() == 66)
                select_location.this.geoLocate();
            return false;
        }
    });
    this.mGps.setOnClickListener(new View.OnClickListener() {
        public void onClick(View param1View) {
            select_location.this.getDeviceLocation();
        }
    });
    hideSoftKeyboard();
}
private void initMap()
{((SupportMapFragment)getSupportFragmentManager().findFragmentById(2131296495)).getMapAsync(this);
}
private void moveCamera(LatLng paramLatLng, float paramFloat, String paramString) {
    this.mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(paramLatLng, paramFloat));
    if (!paramString.equals("My Location")) {
        MarkerOptions markerOptions = (new MarkerOptions()).position(paramLatLng).title(paramString);
        this.mMap.addMarker(markerOptions);
    }
    hideSoftKeyboard();
}
}
public void onConnectionFailed(ConnectionResult paramConnectionResult) {}
protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2131492902);
    this.mSearchText = (AutoCompleteTextView)findViewById(2131296652);
    this.mGps = (FloatingActionButton)findViewById(2131296264);
}

```

```

FloatingActionButton floatingActionButton = (FloatingActionButton)findViewById(2131296262);
this.mBack = floatingActionButton;
floatingActionButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        select_location.this.backToCreate();
    }
});
getLocationPermission();
getWindow().setFlags(512, 512);
}
public void onMapReady(GoogleMap paramGoogleMap) {
    this.mMap = paramGoogleMap;
    if (this.mLocationPermissionGranted.booleanValue()) {
        getDeviceLocation();
        if (ActivityCompat.checkSelfPermission((Context)this, "android.permission.ACCESS_FINE_LOCATION") != 0 &&
        ActivityCompat.checkSelfPermission((Context)this, "android.permission.ACCESS_COARSE_LOCATION") != 0)
            return;
        this.mMap.setMyLocationEnabled(true);
        this.mMap.getUiSettings().setMyLocationButtonEnabled(false);
        this.mMap.getUiSettings().setMapToolbarEnabled(false);
        this.mMap.getUiSettings().setCompassEnabled(false);
        init();
    }
}

public void onRequestPermissionsResult(int paramInt, String[] paramArrayOfString, int[] paramArrayOfint) {
    Boolean bool = Boolean.valueOf(false);
    this.mLocationPermissionGranted = bool;
    if (paramInt != 1234)
        return;
    if (paramArrayOfint.length > 0) {
        for (paramInt = 0; paramInt < paramArrayOfint.length; paramInt++) {
            if (paramArrayOfint[paramInt] != 0) {
                this.mLocationPermissionGranted = bool;
                return;
            }
        }
        this.mLocationPermissionGranted = Boolean.valueOf(true);
        initMap();
    }
}
}
}

```

CreateEvent

```

public class CreateEvent extends FragmentActivity implements OnMapReadyCallback, DatePickerDialog.OnDateSetListener,
TimePickerDialog.OnTimeSetListener {
    private DatabaseReference HostName;
    ImageView IV_Back;
    String amPm1;
    String amPm2;
    Button btn_continue;
    RelativeLayout click;
    TextView displayDateFrom;
    TextView displayDateTo;
    TextView displayTimeFrom;
    TextView displayTimeTo;
    EditText edt_description;
    EditText edt_eventName;
    String hourString;
    String hourStringEnd;
    private double lat;

```

```

private double lng;
private String location;
private boolean mAutoHighlight;
private DatabaseReference mDatabaseReference;
private GoogleMap mMap;
String minuteString;
String minuteStringEnd;
TextView selectDate;
TextView selectLocation;
TextView selectTime;
private DatabaseReference setHost;
String time;
String timeEnd;

private void uploadDatabase() {
    this.HostName.addValueEventListener(new ValueEventListener() {
        public void onCancelled(DatabaseError param1DatabaseError) {}
        public void onDataChange(DataSnapshot param1DataSnapshot) {
            String str3 = CreateEvent.this.edt_eventName.getText().toString();
            String str2 = CreateEvent.this.displayTimeFrom.getText().toString();
            String str1 = CreateEvent.this.displayTimeTo.getText().toString();
            String str4 = CreateEvent.this.displayDateFrom.getText().toString();
            String str5 = CreateEvent.this.displayDateTo.getText().toString();
            String str6 = CreateEvent.this.edt_description.getText().toString();
            Intent intent = CreateEvent.this.getIntent();
            for (DataSnapshot dataSnapshot : param1DataSnapshot.getChildren()) {
                UserProfile userProfile = (UserProfile)param1DataSnapshot.getValue(UserProfile.class);
                StringBuilder stringBuilder = new StringBuilder();
                stringBuilder.append(userProfile.getFirstName());
                stringBuilder.append(" ");
                stringBuilder.append(userProfile.getLastName());
                String str7 = stringBuilder.toString();
                String str8 = FirebaseAuth.getInstance().getUid();
                if (intent.getIntExtra("Check", 0) == 1) {
                    CreateEvent.access$202(CreateEvent.this, intent.getStringExtra("com.example.cove.EXTRA_TEXT"));
                    CreateEvent.access$302(CreateEvent.this, intent.getDoubleExtra("com.example.cove.EXTRA_LAT", 0.0D));
                    CreateEvent.access$402(CreateEvent.this, intent.getDoubleExtra("com.example.cove.EXTRA_LNG", 0.0D));
                }
                if (str3.isEmpty()) {
                    Toast.makeText((Context)CreateEvent.this, "There are empty fields!", 0).show();
                    continue;
                }
                CreateEvent.this.mDatabaseReference.child(str3).setValue(new Events(str3, CreateEvent.this.location,
String.valueOf(CreateEvent.this.lat), String.valueOf(CreateEvent.this.lng), str2, str1, str4, str5, str6, str7, str8));
                CreateEvent.this.mDatabaseReference.child(str3).child("Participants").child(FirebaseAuth.getInstance().getCurrentUser().getUid
()).setValue(new User(FirebaseAuth.getInstance().getCurrentUser().getEmail(), "Online",
FirebaseAuth.getInstance().getCurrentUser().getUid(), "joined"));
                Intent intent1 = new Intent((Context)CreateEvent.this, MainActivity.class);
                CreateEvent.this.startActivity(intent1);
                CreateEvent.this.finish();
            }
        }
    });
}

protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(R.layout.activity_create_event);
    this.selectLocation = (TextView)findViewById(2131296739);
    this.selectTime = (TextView)findViewById(2131296750);
    this.selectDate = (TextView)findViewById(2131296733);
    this.displayTimeFrom = (TextView)findViewById(2131296747);
    this.displayTimeTo = (TextView)findViewById(2131296748);
}

```

```

this.displayDateFrom = (TextView)findViewById(2131296745);
this.displayDateTo = (TextView)findViewById(2131296746);
this.edt_eventName = (EditText)findViewById(2131296261);
this.edt_description = (EditText)findViewById(2131296407);
this.IV_Back = (ImageView)findViewById(2131296348);
this.btn_continue = (Button)findViewById(2131296349);
this.click = (RelativeLayout)findViewById(2131296379);
((SupportMapFragment)getSupportFragmentManager().findFragmentById(2131296496)).getMapAsync(this);
this.mDatabaseReference = FirebaseDatabase.getInstance().getReference("Events");
this.HostName = FirebaseDatabase.getInstance().getReference("User
Profile").child(FirebaseAuth.getInstance().getCurrentUser().getUid());
this.IV_Back.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        Intent intent = new Intent((Context)CreateEvent.this, MainActivity.class);
        CreateEvent.this.startActivity(intent);
        CreateEvent.this.finish();
    }
});
this.selectLocation.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        Intent intent = new Intent((Context)CreateEvent.this, select_location.class);
        CreateEvent.this.startActivity(intent);
    }
});
this.click.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        Intent intent = new Intent((Context)CreateEvent.this, select_location.class);
        CreateEvent.this.startActivity(intent);
    }
});
this.selectTime.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        Calendar calendar = Calendar.getInstance();
        TimePickerDialog timePickerDialog = TimePickerDialog.newInstance(CreateEvent.this, calendar.get(11),
calendar.get(12), false);
        timePickerDialog.setOnCancelListener(new DialogInterface.OnCancelListener() {
            public void onCancel(DialogInterface param2DialogInterface) {
                Log.d("TimePicker", "Dialog was cancelled");
            }
        });
        timePickerDialog.show(CreateEvent.this.getFragmentManager(), "Timepickerdialog");
    }
});
this.selectDate.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        Calendar calendar = Calendar.getInstance();
        DatePickerDialog datePickerDialog = DatePickerDialog.newInstance(CreateEvent.this, calendar.get(1),
calendar.get(2), calendar.get(5));
        datePickerDialog.setAutoHighlight(CreateEvent.this.mAutoHighlight);
        datePickerDialog.show(CreateEvent.this.getFragmentManager(), "Datepickerdialog");
    }
});
this.btn_continue.setOnClickListener(new View.OnClickListener() {
    public void onClick(View param1 View) {
        CreateEvent.this.uploadDatabase();
    }
});
getWindow().setFlags(512, 512);
}
public void onDateSet(DatePickerDialog paramDatePickerDialog, int paramInt1, int paramInt2, int paramInt3, int paramInt4,
int paramInt5, int paramInt6) {
    StringBuilder stringBuilder1 = new StringBuilder();

```



```

        stringBuilder1.append(paramInt3);
        stringBuilder1.append("/");
        stringBuilder1.append(paramInt2 + 1);
        stringBuilder1.append("/");
        stringBuilder1.append(paramInt1);
        String str1 = stringBuilder1.toString();
        StringBuilder stringBuilder2 = new StringBuilder();
        stringBuilder2.append(paramInt6);
        stringBuilder2.append("/");
        stringBuilder2.append(paramInt5 + 1);
        stringBuilder2.append("/");
        stringBuilder2.append(paramInt4);
        String str2 = stringBuilder2.toString();
        this.displayDateFrom.setText(str1);
        this.displayDateTo.setText(str2);
    }
    public void onMapReady(GoogleMap paramGoogleMap) {
        this.mMap = paramGoogleMap;
        paramGoogleMap.getUiSettings().setAllGesturesEnabled(false);
        this.mMap.getUiSettings().setMapToolbarEnabled(false);
        Intent intent = getIntent();
        if (intent.getIntExtra("Check", 0) == 1) {
            String str = intent.getStringExtra("com.example.cove.EXTRA_TEXT");
            ((TextView)findViewById(2131296744)).setText(str);
            LatLng latLng = new LatLng(intent.getDoubleExtra("com.example.cove.EXTRA_LAT", 0.0D),
intent.getDoubleExtra("com.example.cove.EXTRA_LNG", 0.0D));
            this.mMap.addMarker(new MarkerOptions().position(latLng));
            this.mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 14.0F));
        }
    }
    public void onResume() {
        super.onResume();
        DatePickerDialog datePickerDialog = (DatePickerDialog)getFragmentManager().findFragmentByTag("Datepickerdialog");
        if (datePickerDialog != null)
            datePickerDialog.setOnDateSetListener(this);
    }
    public void onTimeSet(RadialPickerLayout paramRadialPickerLayout, int paramInt1, int paramInt2, int paramInt3, int
paramInt4) {
        if (paramInt1 > 11) {
            if (paramInt1 == 12) {
                stringBuilder1 = new StringBuilder();
                stringBuilder1.append("");
                stringBuilder1.append(paramInt1);
                this.hourString = stringBuilder1.toString();
                this.amPm1 = "PM";
            } else {
                paramInt1 -= 12;
                if (paramInt1 < 10) {
                    stringBuilder1 = new StringBuilder();
                    stringBuilder1.append("0");
                } else {
                    stringBuilder1 = new StringBuilder();
                    stringBuilder1.append("");
                }
                stringBuilder1.append(paramInt1);
                this.hourString = stringBuilder1.toString();
                this.amPm1 = "PM";
            }
        }
        } else if (paramInt1 == 0) {
            stringBuilder1 = new StringBuilder();
            stringBuilder1.append("");
            stringBuilder1.append(paramInt1 + 12);

```

```

        this.hourString = stringBuilder1.toString();
        this.amPm1 = "AM";
    } else {
        if (paramInt1 < 10) {
            stringBuilder1 = new StringBuilder();
            stringBuilder1.append("0");
        } else {
            stringBuilder1 = new StringBuilder();
            stringBuilder1.append("");
        }
        stringBuilder1.append(paramInt1);
        this.hourString = stringBuilder1.toString();
        this.amPm1 = "AM";
    }
    if (paramInt3 > 11) {
        if (paramInt3 == 12) {
            stringBuilder1 = new StringBuilder();
            stringBuilder1.append("");
            stringBuilder1.append(paramInt3);
            this.hourStringEnd = stringBuilder1.toString();
            this.amPm2 = "PM";
        } else {
            paramInt1 = paramInt3 - 12;
            if (paramInt1 < 10) {
                stringBuilder1 = new StringBuilder();
                stringBuilder1.append("0");
            } else {
                stringBuilder1 = new StringBuilder();
                stringBuilder1.append("");
            }
            stringBuilder1.append(paramInt1);
            this.hourStringEnd = stringBuilder1.toString();
            this.amPm2 = "PM";
        }
    } else if (paramInt3 == 0) {
        stringBuilder1 = new StringBuilder();
        stringBuilder1.append("");
        stringBuilder1.append(paramInt3 + 12);
        this.hourStringEnd = stringBuilder1.toString();
        this.amPm2 = "AM";
    } else {
        if (paramInt3 < 10) {
            stringBuilder1 = new StringBuilder();
            stringBuilder1.append("0");
        } else {
            stringBuilder1 = new StringBuilder();
            stringBuilder1.append("");
        }
        stringBuilder1.append(paramInt3);
        this.hourStringEnd = stringBuilder1.toString();
        this.amPm2 = "AM";
    }
    if (paramInt2 < 10) {
        stringBuilder1 = new StringBuilder();
        stringBuilder1.append("0");
    } else {
        stringBuilder1 = new StringBuilder();
        stringBuilder1.append("");
    }
    stringBuilder1.append(paramInt2);
    this.minuteString = stringBuilder1.toString();
    if (paramInt4 < 10) {

```

```

        stringBuilder1 = new StringBuilder();
        stringBuilder1.append("0");
    } else {
        stringBuilder1 = new StringBuilder();
        stringBuilder1.append("");
    }
    stringBuilder1.append(paramInt4);
    this.minuteStringEnd = stringBuilder1.toString();
    StringBuilder stringBuilder1 = new StringBuilder();
    stringBuilder1.append(this.hourString);
    stringBuilder1.append(":");
    stringBuilder1.append(this.minuteString);
    this.time = stringBuilder1.toString();
    stringBuilder1 = new StringBuilder();
    stringBuilder1.append(this.hourStringEnd);
    stringBuilder1.append(":");
    stringBuilder1.append(this.minuteStringEnd);
    this.timeEnd = stringBuilder1.toString();
    TextView textView = this.displayTimeFrom;
    StringBuilder stringBuilder2 = new StringBuilder();
    stringBuilder2.append(this.time);
    stringBuilder2.append(" ");
    stringBuilder2.append(this.amPm1);
    textView.setText(stringBuilder2.toString());
    textView = this.displayTimeTo;
    stringBuilder2 = new StringBuilder();
    stringBuilder2.append(this.timeEnd);
    stringBuilder2.append(" ");
    stringBuilder2.append(this.amPm2);
    textView.setText(stringBuilder2.toString());
}
}

```

Ervyn Christian S. Agoy

Address : Soro-Soro Karsada Batangas City

Contact Number : 0956-804-8220

Email Address : ervync23@gmail.com



PERSONAL INFORMATION

Age : 22

Date of Birth : October 23, 1997

Gender : Male

Civil Status : Single

Height : 5'3"

Weight : 55kg

Nationality : Filipino

Religion : Roman Catholic

EDUCATIONAL BACKGROUND:

Primary : **Saint Bridget College**
M.H Del Pillar, Batangas City

Secondary : **Saint Bridget College**
M.H Del Pillar, Batangas City

Tertiary : **Lyceum of the Philippines University Batangas**
Capitol Site, Batangas City

Mark Elo L. Dalangin

Address : Gulod Itaas, Batangas City

Contact Number : 0977-741-2571

Email Address : mrkdln0907@gmail.com



PERSONAL INFORMATION

Age : 21
Date of Birth : September 29, 1998
Gender : Male
Civil Status : Single
Height : 5'10"
Weight : 70kg
Nationality : Filipino
Religion : Roman Catholic

EDUCATIONAL BACKGROUND:

Primary : **Gulod Elementary School**
Gulod Itaas, Batangas City

Secondary : **Batangas National High School**
Rizal Ave, Poblacion, Batangas

Tertiary : **Lyceum of the Philippines University Batangas**
Capitol Site, Batangas City

Joshua G. Grantoza

Address : 151 Tampoy Gasang, Mabini Batangas

Contact Number : 0999-193-1811

Email Address : joshuagrantoza@gmail.com



PERSONAL INFORMATION

Age : 21
Date of Birth : October 17, 1998
Gender : Male
Civil Status : Single
Height : 5'8"
Weight : 98kg
Nationality : Filipino
Religion : Roman Catholic

EDUCATIONAL BACKGROUND

Primary : **Gasang Elementary School**
Barangay Gasang, Mabini Batangas

Secondary : **Saint Francis Academy**
Poblacion, Mabini Batangas

Tertiary : **Lyceum of the Philippines University Batangas**
Capitol Site, Batangas City

Kim Lemars A. Magnaye

Address : Kumintang Ibaba, Batangas City

Contact Number : 0917-921-0988

Email Address : kimpoy114@gmail.com



PERSONAL INFORMATION

Age : 25

Date of Birth : January 14, 1994

Gender : Male

Civil Status : Single

Height : 5'6"

Weight : 55kg

Nationality : Filipino

Religion : Roman Catholic

EDUCATIONAL BACKGROUND

Primary : **Holy Child Academy**
Rizal Ave, Poblacion, Batangas City

Secondary : **University of Batangas**
Hilltop Kumintang Ibaba, Batangas City

Tertiary : **Lyceum of the Philippines University Batangas**
Capitol Site, Batangas City

Carl Adrian R. Sarrol

Address : Villa Ferlin Resort, Manghinao Proper,
Bauan, Batangas

Contact Number : 0945 991 6626

Email Address : sarrolcarladrian@gmail.com



PERSONAL INFORMATION

Age : 21

Date of Birth : November 19, 1998

Gender : Male

Civil Status : Single

Height : 5'6"

Weight : 40kg

Nationality : Filipino

Religion : Roman Catholic

EDUCATIONAL BACKGROUND:

Primary : **Casa Del Bambino Emmanuel Montessori**
Contreras Compound 4200 Batangas City, Philippines

Secondary : **Gratia De Regina**
San Pascual, Batangas

Tertiary : **Lyceum of the Philippines University Batangas**
Capitol Site, Batangas City