Project LFG

CMPT 362 Final Project

Group 4 members:
Allen Wei
Chien An Chen (Edward)
Zachariah Delft

Project Goal

To create a social event scheduling application with the following features:

- A user will be able to login to or register an account
- Registered users can create or join events in their local areas or abroad through a map based implementation
- Users will be able to communicate with other registered users through the following methods:
    - Event chat rooms
    - Direct messaging

Why Project was worth exploring

- One of the main challenges of Project LFG was the need for a centralized data hub for all users
- This was accomplished using Google's Firebase Realtime Database (FRD)
- FRD had advantages over other databases such as SQlite databases and Room for the following reasons:
  - FRD is Google's NoSQL cloud based database
  - FRD uses data synchronization
  - Data is stored in a JSON format

Why Project was worth exploring

- Using FDR advantages we were able to allow users to
  - Register and login to accounts
  - Create, join, comment and rate events
  - Communicate between users directly or in chatrooms

Challenges and Scope

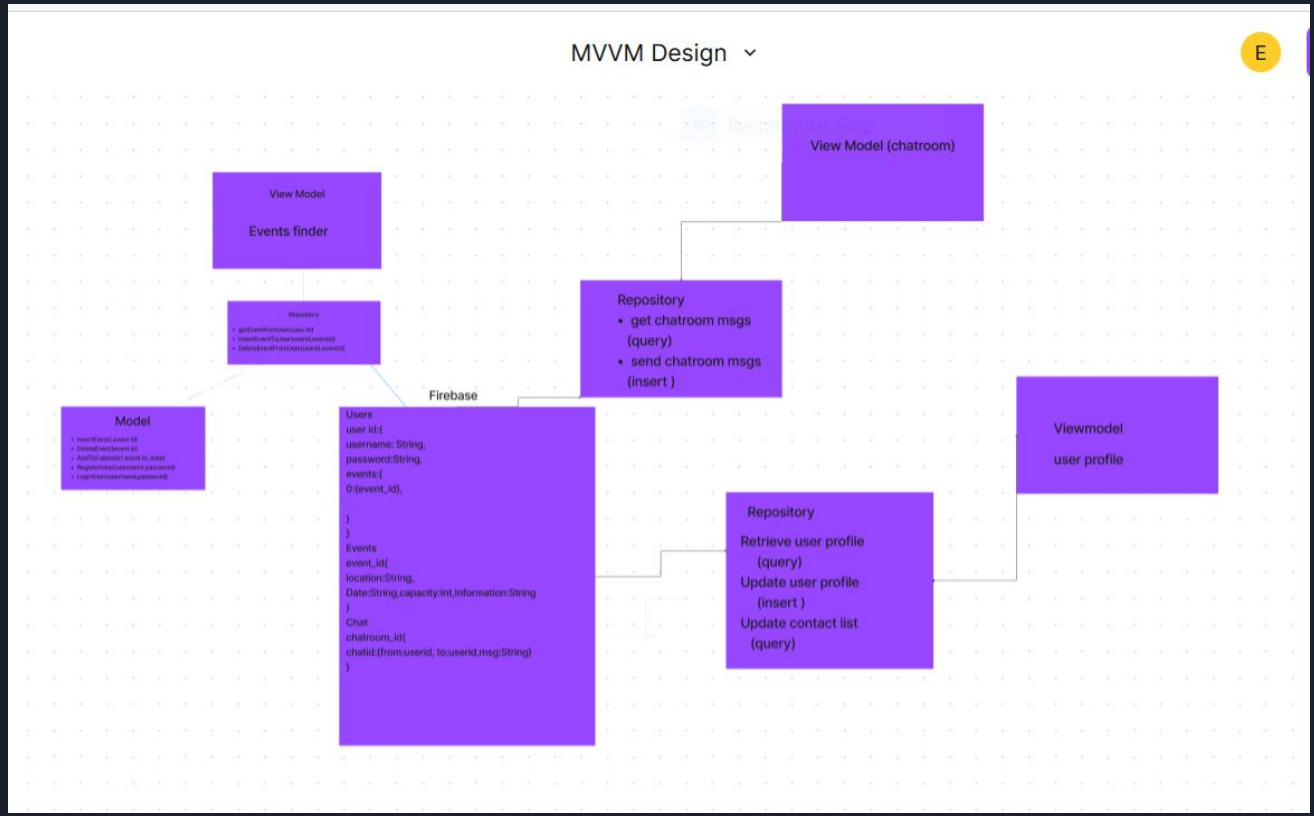
Our group faced a number of challenges during the development process

Unfortunately, one of our team members had to withdraw due to extraneous circumstance in their personal life and others had to deal with sickness, such as Covid-19

The group overcame these issues while developing the knowledge and understanding needed to implement Google's Firebase Realtime Database which was a vital part of the overall project

Our group managed to use the newly acquired knowledge applied to principles developed over the course of the term to produce an application that allows large number of users to create and join events, as well as interact through different communication methods creating an interactive social element

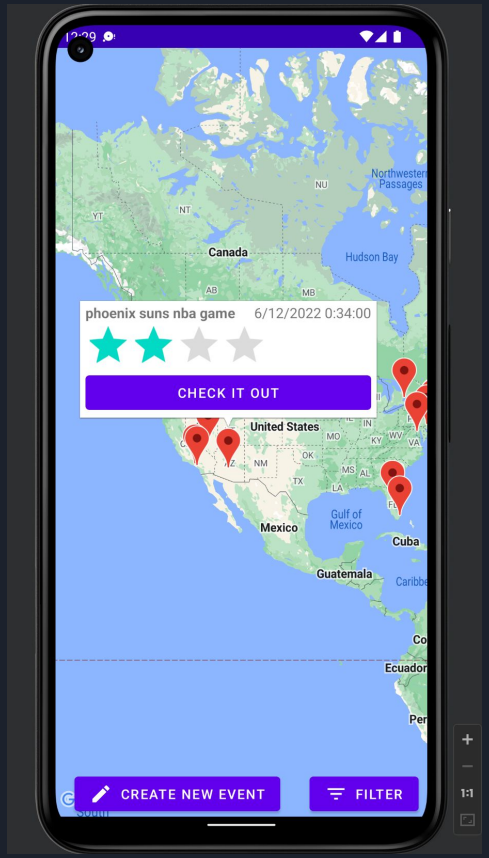
MVVM Diagram



New Skill Sets

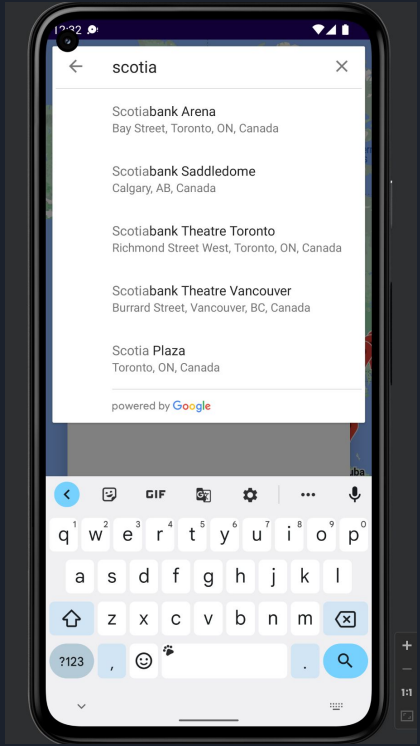
1. Design Google Map marker icon and GoogleMap.InfoWindowAdapter
2. Google Places API
3. Rating Bar
4. runBlock{} and await
5. Suspend functions
6. Kotlin filter library
7. Recyclerview

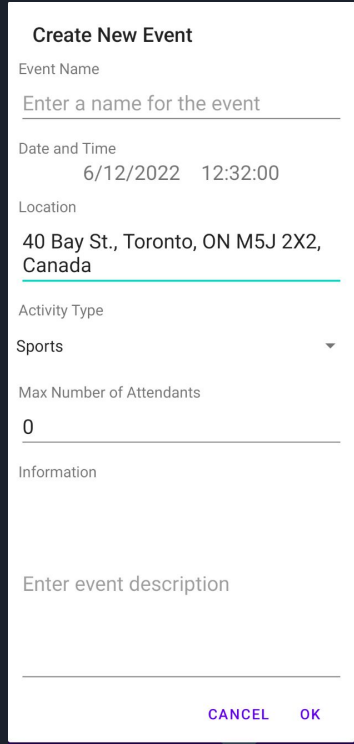
Google Maps Marker



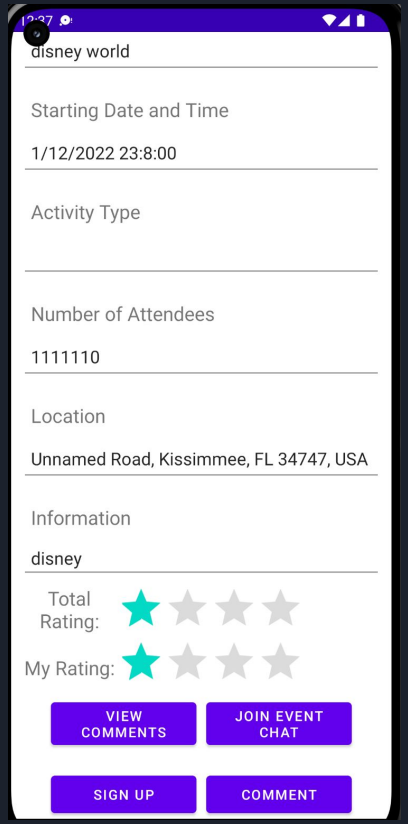

```
mMap.setOnInfoWindowClickListener {  it: Marker
            val name = it.title
            val info = DBHashMap.get(name);
            val intent = Intent(applicationContext,EventInfoActivity::class
            intent.putExtra(NAME,info!!.name);
            intent.putExtra(LOCATION,info.location);
            intent.putExtra(STARTINGDATE,info.startingdate);
            intent.putExtra( name: "Attendants",info.attendess);
            intent.putExtra( name: "info",info.information);
            intent.putExtra( name: "key",info.id)
            intent.putExtra( name: "info",info.information);
            intent.putExtra(ACTIVITYTYPESTR,info.activitytypes)
            startActivity(intent);

}
mMap.setOnMarkerClickListener(object : GoogleMap.OnMarkerClickListener{
    override fun onMarkerClick(p0: Marker): Boolean {
        p0.showInfoWindow()
        return true;
    }
```

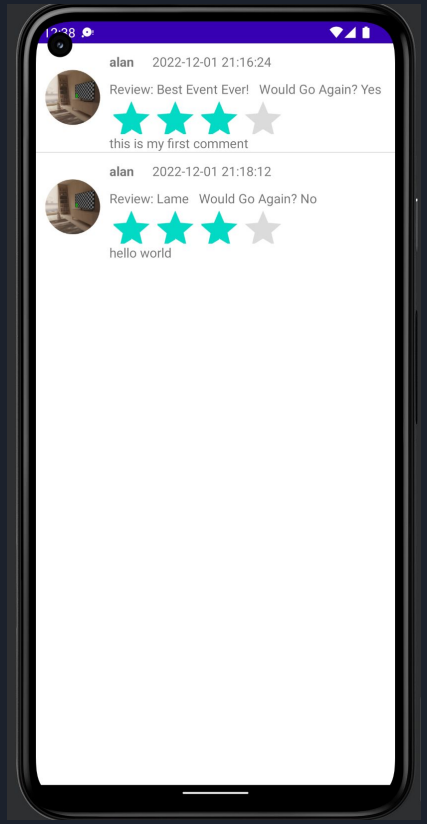
Google Places Api





Rating Bar


disney world
Starting Date and Time
1/12/2022 23:8:00
Activity Type
Number of Attendees
1111110
Location
Unnamed Road, Kissimmee, FL 34747, USA
Information
disney
Total Rating:
My Rating:
VIEW COMMENTS
JOIN EVENT CHAT
SIGN UP
COMMENT


alan
2022-12-01 21:16:24
Review: Best Event Ever! Would Go Again? Yes
this is my first comment
alan
2022-12-01 21:18:12
Review: Lame Would Go Again? No
hello world

Runblocking{} and Await()

```
runBlocking {  this: CoroutineScope
         infowindowview =jobA();
        val data = totalrating() as HashMap<String,Long>
        var total :Long= 0;
        var count = 0;
         for((key,value) in data){
            total += value;
            count ++
        }
        if(count > 0){
            println("3.${total/count}");
            infowindowview.findViewById<RatingBar>(R.id.InfoWindowtotalratingbar).rating = (to
        }

}

return   infowindowview;
```

Suspend

```
}
suspend fun totalrating():HashMap<String,Long> {
    val db = FirebaseDatabase.getInstance().reference.child( pathString: "events1").child(p0.title
          .child("ratings")

    val data = db.get().await()
    val eventdata = data.value as HashMap<String,*>
    if(!eventdata.containsKey("ratings")){
        val tmp = HashMap<String,Long>();
        return tmp
    }
    val tmp = eventdata.get("ratings") as HashMap<String,Long>
    return tmp;
}
```

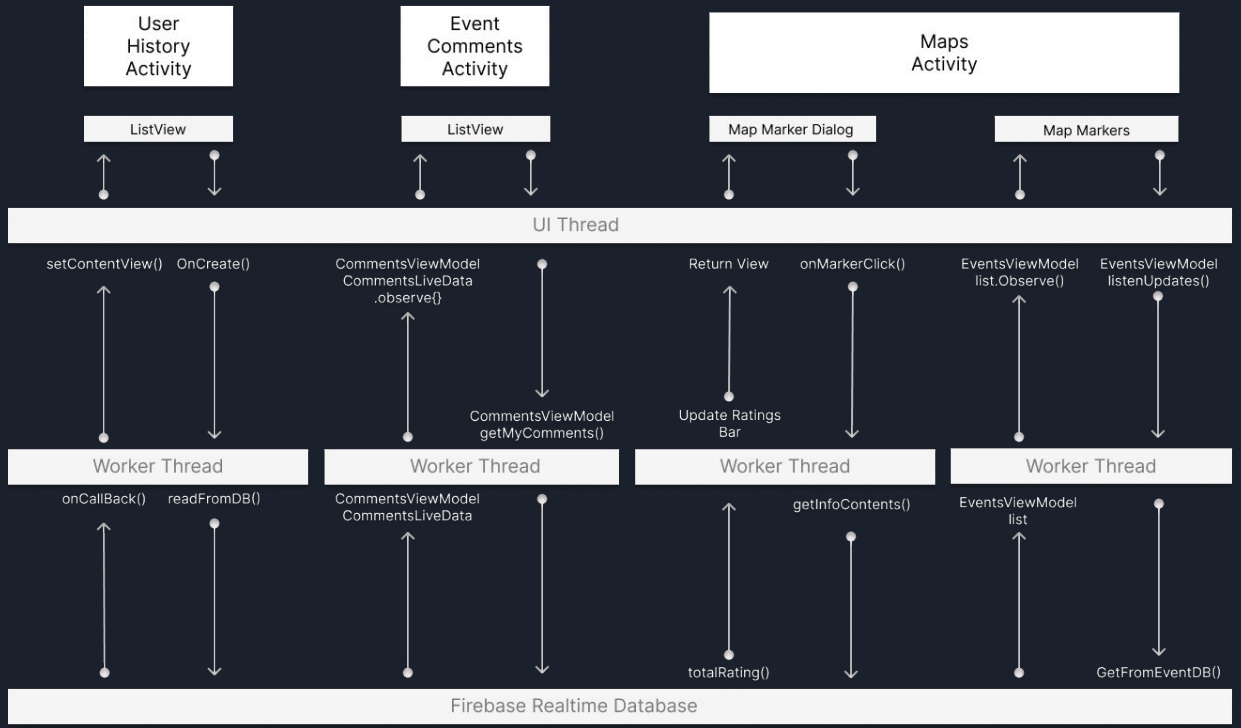
Filter

```
fun SortEvents(activitytype:String,numofppl:String){

    val curr = lst.value;
    if(curr == null) return;
    when(numofppl){
        ">100"->{
             val filtered = curr!!.filterNot{ info-> info.attendess > 100 && info.activitytypes =
            filteredList.value = ArrayList(filtered);
        }
        ">1000"->{
            val filtered = curr!!.filterNot { info->info.attendess>1000 && info.activitytypes ==
            filteredList.value = ArrayList(filtered);
        }
        "All"->{
            val filtered = curr!!.filterNot { info->info.activitytypes == activitytype }
            filteredList.value= ArrayList(filtered);
        }
    }

}
```

Recyclerview

```
package com.example.projectlfg                                    ⚠ 6 ⚠ 4 ⚠ 3

import ...

class MsgAdapter(val context: Context, val msgList: ArrayList<Message>): RecyclerView.Adapter<Recycl


    class SentMsgHolder(itemView: View): RecyclerView.ViewHolder(itemView){
        val msg = itemView.findViewById<TextView>(R.id.sentmsgtext)
        val imgview = itemView.findViewById<ImageView>(R.id.chatsendimg)
    }

    class ReceiveMsgHolder(itemView: View): RecyclerView.ViewHolder(itemView){
        val msg = itemView.findViewById<TextView>(R.id.receivemsgtext)
        val imgview = itemView.findViewById<ImageView>(R.id.chatrcvimg)
    }

    // return proper view type to determin sent/receive msg type
    override fun getItemViewType(position: Int): Int {

        val currentTxt = msgList[position]
        val curruserid = FirebaseAuth.getInstance().currentUser!!.uid.toString()
        // sent msg
        if(curruserid == currentTxt.sender){
            return 1

        //  receive msg
        }
        return 2
    }


    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {

        lateinit var view: View

        // if it's a sent msg
        if(viewType == 1){
```

Thread Design Diagram



LFG Demo

// For slides hand in should have link or embed video demo of application

// we can edit the same video into the final video presentation

Members Contributions

Allen Wei:

- Google Map - Designed the creation, filter, and marker information window on the Google Map
- Event Information - Designed the UI of this activity, fill the data in each box, and ratings
- Comment - Designed the frontend and backend
- Scheduled Activity - Designed the frontend and backend
- Config Activity - Designed the frontend and backend

Chien An Chen (Edward):

- Implement chat room and contact list
- Main menu activity
- Firebase initialization
- Pitching design ideas

Zachariah Delft:

- Branch merge and conflict resolution
- Initial map activity implementation
- Supporting role including bug fixing

Lessons Learnt

1. Positive
    a. Coroutines - Able understand what coroutines do and different ways to implement it
    b. Able to start a project from scratch (UI Design, MVVM model, etc) and produce something users can user daily
2. Negative
    a. Should start the project earlier and be better with time management because each group member needs to distribute his energy to all their courses and other obligations
    b. Unfortunate circumstances and illness can arise over the course of the project, and better communication with team members about ones progression and struggles is important