



GroupThink

By Ali Shammout, Marius Goffart, Griffin Haymes, and Robert Gargaro



GroupThink

What is GroupThink?

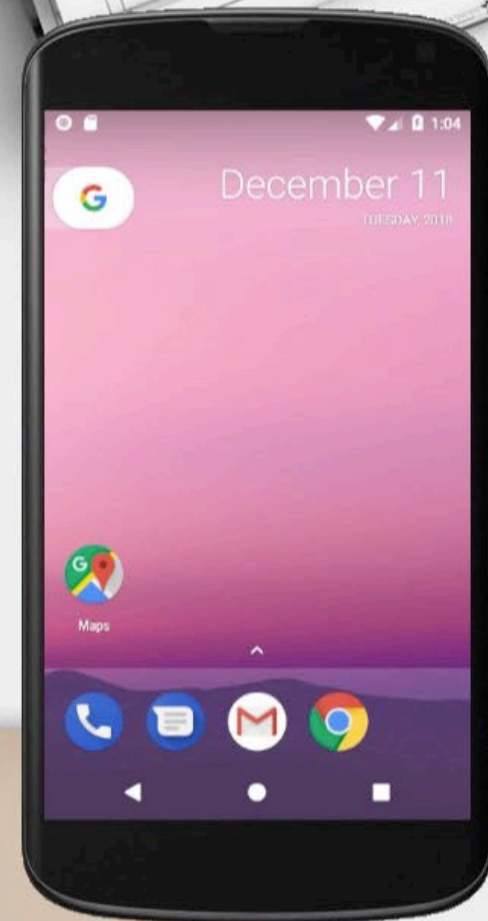
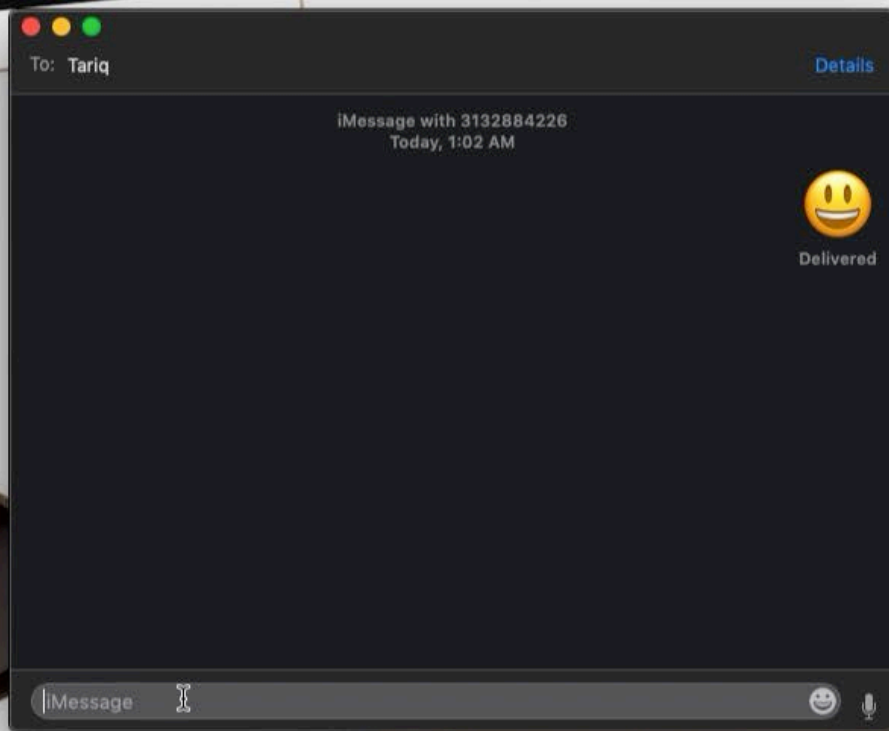
- GroupThink is a mobile application designed to:
 - Organize Group meetings
 - Create Agendas
 - Facilitate meetings for smoother group dynamics



GroupThink

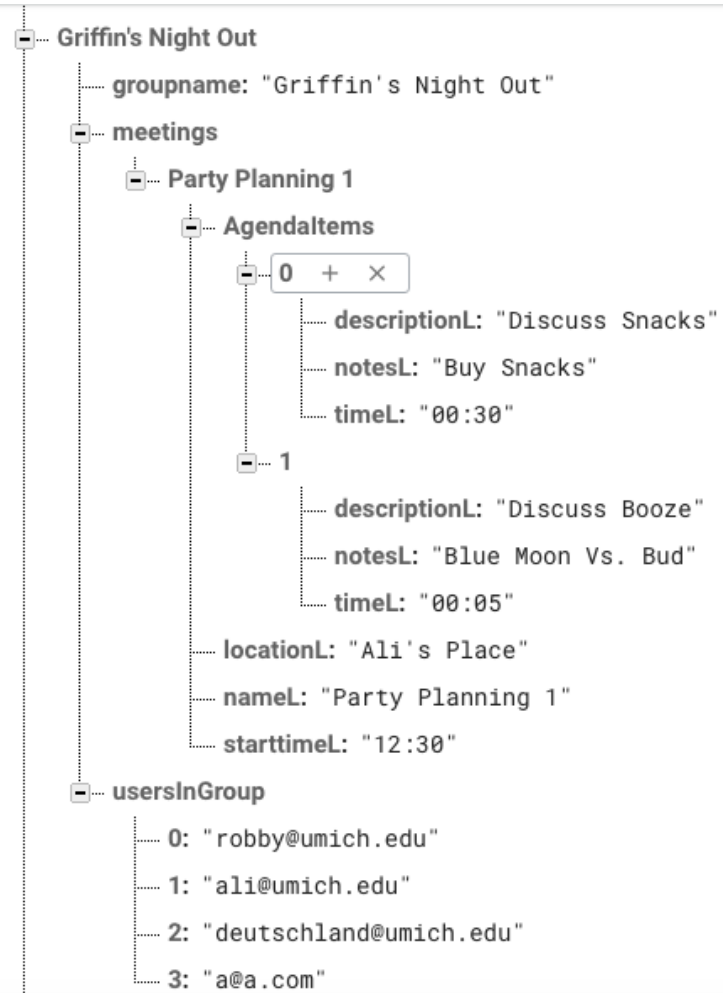
Demo Video

- Tariq is requesting Ali set up a GroupThink group for the organization Books for Kids and plan their mass meeting
 - Tariq specified a list of members, meeting time, and meeting location
 - Ali has free rein to create the meeting agenda



Current State

Class and Data Structure



Current State

```
groupthink
├── AddGroup
├── AddPeople
├── AgendaCreation
├── AgendaItemsClass
├── GroupClass
├── GroupOverview
├── MainActivity
├── MeetingClass
├── MeetingCreation
├── MeetingLayout
├── MeetingSelection
├── RecyclerViewAdapterAC
├── RecyclerViewAdapterAddGroup
├── RecyclerViewAdapterAddPeople
├── RecyclerViewAdapterGroupOverview
├── RecyclerViewAdapterMeetingLayout
├── RecyclerViewAdapterMeetingSelection
├── RecyclerViewAdapterWIG
├── SplashScreen
├── UserClass
└── WhosInGroup
```

```
RecyclerViewAdapterGroupOverview(ArrayList<String> groupsForUser2, Context nContext) {
    this.groupsForUser2 = groupsForUser2;
    this.nContext = nContext;
}

@NonNull
@Override
public ViewHolderGroupOverview onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
    View view = LayoutInflater.from(viewGroup.getContext())
        .inflate(R.layout.layout_recycler_groupoverview, viewGroup, attachToRoot: false);
    ViewHolderGroupOverview viewHolderGroupOverview = new ViewHolderGroupOverview(view);
    return viewHolderGroupOverview;
}

@Override
public void onBindViewHolder(@NonNull ViewHolderGroupOverview viewHolderGroupOverview, final int i) {
    viewHolderGroupOverview.textViewSelectGroup.setText(groupsForUser2.get(i));
    viewHolderGroupOverview.parentLayoutGroupOverview.setOnClickListener(v -> {

        Intent intent = new Intent(nContext, MeetingSelection.class);
        intent.putExtra( name: "passed_group", groupsForUser2.get(i));
        nContext.startActivity(intent);

    });
}

@Override
public int getItemCount() { return groupsForUser2.size(); }

public class ViewHolderGroupOverview extends RecyclerView.ViewHolder {

    TextView textViewSelectGroup;
    RelativeLayout parentLayoutGroupOverview;
    public ViewHolderGroupOverview(@NonNull View itemView) {
        super(itemView);

        textViewSelectGroup = itemView.findViewById(R.id.textViewSelectGroup);
        parentLayoutGroupOverview = itemView.findViewById(R.id.parentLayoutGroupOverview);
    }
}
```

Current State

```
//updates Member List
//still needs to display the Array
//maybe a single textViewShowMember isn't the best solution? (also in terms of looks)
myRef.child("usersInGroup").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for(DataSnapshot child : dataSnapshot.getChildren()) {
            String member = child.getValue(String.class);
            Members.add(member);
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
});

private void getAgenda() {
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference(currentGroup);
    myRef.child("meetings").child(currentMeeting).child("AgendaItems").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for(DataSnapshot child : dataSnapshot.getChildren()) {
                AgendaItemsClass agendaItem = child.getValue(AgendaItemsClass.class);
                wholeAgenda.add(agendaItem);
            }
            recyclerViewAdapterMeetingLayout.notifyDataSetChanged();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
        }
    });
}
```

```
public void getGroups () {
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference();

    myRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                for (DataSnapshot snapshot2 : snapshot.child("usersInGroup").getChildren()) {
                    String myUser = snapshot2.getValue(String.class);
                    if (mauth.getCurrentUser().getEmail().equals(myUser)) {
                        groupsForUser.add(snapshot.getKey());
                    }
                }
            }
            recyclerViewAdapterGroupOverview.notifyDataSetChanged();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
        }
    });
}
```

Current State

```
//method for checking valid email format
public static boolean isEmailValid (String email) {
    String expression = "^([\\w\\.\\-]+)@([\\w\\.\\-]+\\.)+[A-Z]{2,4}$";
    Pattern pattern = Pattern.compile(expression, Pattern.CASE_INSENSITIVE);
    Matcher matcher = pattern.matcher(email);
    return matcher.matches();
}
```

```
//method for checking valid time format
public static boolean isValidTime (String time) {
    String tester = "[0-9]{2}+:+[0-9]{2}";
    Pattern pattern = Pattern.compile(tester, Pattern.CASE_INSENSITIVE);
    Matcher matcher = pattern.matcher(time);
    return matcher.matches();
}
```

```
Intent intent = new Intent( packageContext: MeetingLayout.this, WhosInGroup.class);
intent.putExtra( name: "groupnameMT", currentGroup);
intent.putExtra( name: "meetingnameMT", currentMeeting);
startActivity(intent);
```


Current State

```
public class SplashScreen extends Activity {

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

        ImageView myImageView = findViewById(R.id.logo1);
        myImageView.setImageResource(R.drawable.finallogogroupthink);

        new java.util.Timer().schedule(
            new java.util.TimerTask() {
                @Override
                public void run() {
                    startActivity(new Intent( packageContext: SplashScreen.this,
                                            MainActivity.class));
                }
            }, delay: 6000
        );
    }
}
```





What Brought Us Joy?

- Learning the vast amount of things that code can do
- Designing the application with the intent of having a product which addresses a certain need
- Finishing our first prototype!



What Gave Us Nightmares?

- Working with Firebase to properly store all of our information
- Figuring out the Data Structure and how to effectively layer our nested class objects
- Setting up Recycler Views



Future Development

- Calendar functionality
 - Automatic mutual-availability finder
- Ability to add images, graphs, tables to agenda items
- Improved User Interface for more of a Modern-Design



Thank You for
Listening

GroupThink[®]