

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: [mohamedMRG7](#)

Party Photos

Description

Party photos allows you to create a party room and save all the party photos u taken at this room so any of your party friends can access it without need to send the photos to every one of your friends

-party photos allows you to make the room private so just who have the room password can access or make the party room open so any one can access it

-you can discover other people parties photos

Intended User

-Friends group

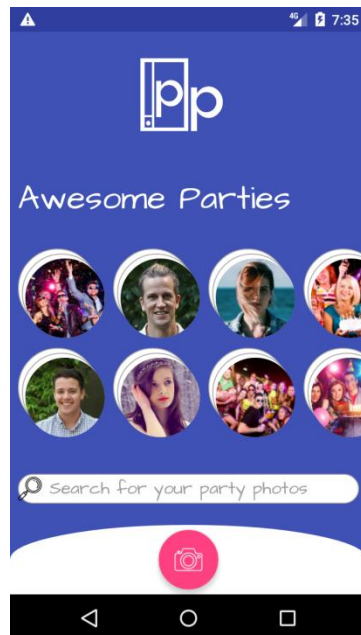
Features

- * takes pictures
- * share pictures
- * search for party
- * download pictures

User Interface Mocks

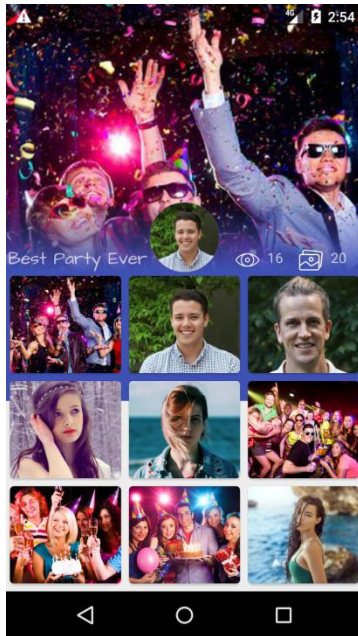
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



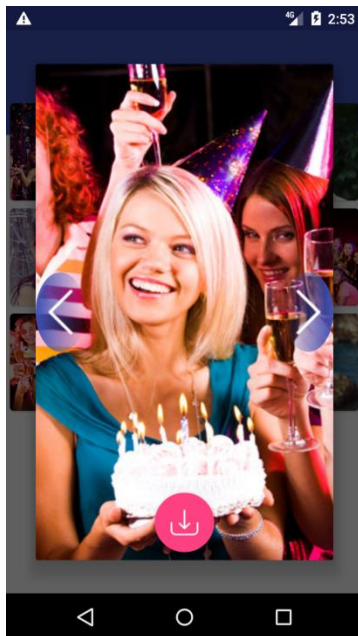
Main screen u can search for your party , discover other people parties or create a new party room to save your party photos at it

Screen 2



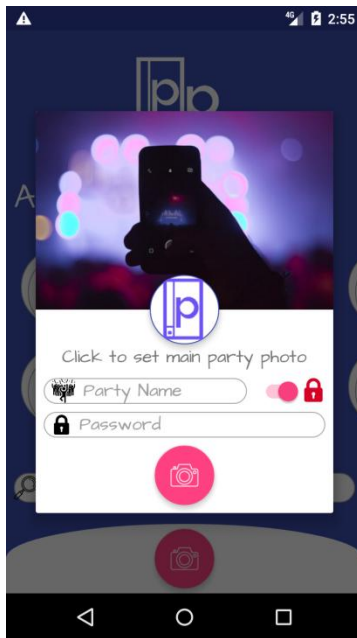
Show all the party photos and details

Screen 3



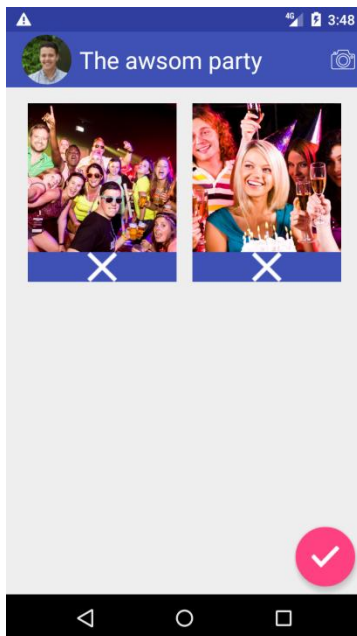
Discover photos in bigger size and download the photo you like

Screen 4



Create new party room and choose its name and password if private

Screen 5



After declaring the party name and password will open the camera and start to take photos and after finish the photos taken will appear in this screen for delete or take more photos or save and start sharing with friends

Screen 6



Downloaded photo widget which show all the downloaded parties photos

Key Considerations

How will your app handle data persistence?

-Use Firebase real time database to store the party data

-Use Firebase Storage to save the photos

Describe any edge or corner cases in the UX.

-User will return to main screen if he hits the back button

- in screen 5 when hit back button will show dialog to ask user to out without sharing the current party

Describe any libraries you'll be using and share your reasoning for including them.

- `com.google.firebase:firebase-database:16.0.1` to save the party details
- `com.google.firebase:firebase-storage:11.8.0` to upload the party photos at it and reuse it by the links saved in the database
- `com.android.support:recyclerview-v7:27.0.2` to show the lists of photos and parties
- `com.jakewharton:butterknife:8.8.1` to use it instead of `findViewById` for less code
- `com.github.bumptech.glide:glide:4.7.1` to load photos from fire base storage

Describe how you will implement Google Play Services or other external services.

Will use google play services by add

- `classpath 'com.google.gms:google-services:3.2.0'` to build.gradle project level
- `apply plugin: 'com.google.gms.google-services'` to build .gradle app level
- add `google-services.json` file in app dir

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Configure libraries
- Setup firebase api
- Add permissions
- Modify app them
- Download required font
- Add required photos and icons to drawable folder

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity (screen 1)
- Build UI for partyDetailsActivity (screen 2)
- Build UI for LargPhotosGallaryActivity (screen 3)
- Build UI for StartPartyDialog (screen 4)
- Build UI for PartyManageActivity (screen 5)
- Build UI for the widget
- Implement transitions and animation between the activities

Task 3: Manage Firebase storage and database

- Create class PartyData which contain objects like partyName,id,numOfViews ,List of photos links, and other objects and its getter and setter functions
- Create class DataBaseUtilies which manage the insert and get the data from database
- Create class StorageUtilies which manage uploading photos to firebaseStorage and get its links
- Using asyntask to download the photos from fireBaseStorage

Task 4: Implement OnClickLisner for the views

- Implement onclicklisner for adapters to open and send the required data to the target activity
- Implement onClickLisner for the the buttons to open camera , insert the party data to the database and download photo to specific location in sdCard

Task 5: Manage adapters and recyclerviews

- Create adapter for the list of parties in (screen 1) and attach it to the recyclerview
- Create adapter for the autocompleteTextView (in screen 1) and attach it
- Create adapter for the party photos (in screen 2) and attach it to the recyclerview
- Get the data from database by using DatabaseUtilies and insert the required data to the adapters

Task 6: setup widget to show downloaded photos

Describe the next task. List the subtasks. For example:

- Create required files for widget (widgetProvider ,widgetProviderLayout and widgetProviderInfo) and Add widget provider to Androidmanifest
- Create the adapter of girdview by create gridViewRemotService and gridViewRemoteFactory
- Create widgetService to update the gridView content