

Film Friend Application Design

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Login Screen](#)

[Register Screen](#)

[Home Screen](#)

[Group Screen](#)

[Movie Details Screen](#)

[Group Detail String](#)

[Key Considerations](#)

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

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

[Describe how you will implement Google Play Services or other external services.](#)

[Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement object models](#)

[Task 3: Implement API Calls](#)

[Task 4: Implement UI Layouts](#)

[Task 5: Implement account features](#)

[Task 6: UI and Data Linking for Home Activity](#)

[Task 7: Implement Movie Activity Features](#)

[Task 8: Implement Group Feature](#)

[Task 9: Implement Saved Features](#)

[Task 10: Implement Transitions](#)

[Task 11: Implement Testing](#)

GitHub Username: izhang1

Description

MovieMate is the social media platform for movie lovers locally and around the world. Find other who enjoy the same genre of movies, join groups and setup outings together.

If you ever felt lonely because your genre of movies is different then all your friends? This is the app to help you connect with others who share the same passion for action films or love of horror movies. Watch, rate and connect with others.

Intended User

Movie lovers, people who enjoy watching film and going to the movies. It's for the nerdy guy who wants to find someone else to debate about the newest marvel movie. It can also be for the social people who wants to find a group of friends to watch drama or horror with.

Features

Main Features

- Create an account
- Find and join local movie groups
- Find movies and information about their ratings and when local theater's playtime
- Rate and comment on movies
- Use GPS hardware to find local movie theaters and movies

User Interface Mocks

Login Screen

Activity asking for user input and allowing them to login with an email and password.

A hand-drawn sketch of a login screen. At the top right, it says "< LOGIN PAGE". In the center, there is a square box containing the text "Film Friends". Below this box are two horizontal lines for input, labeled "EMAIL" and "PASSWORD". Under the password field is a small rectangular button labeled "LOGIN". At the bottom right corner, the word "REGISTER" is written.

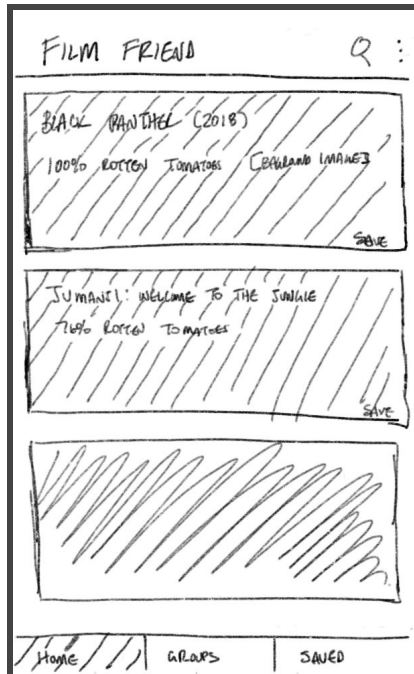
Register Screen

Allow users to register for an account using their email, username and password.

A hand-drawn sketch of a register screen. At the top right, it says "< REGISTER PAGE". In the center, there is a square box containing the text "Film Friend". Below this box are three horizontal lines for input, labeled "EMAIL", "USERNAME", and "PASSWORD". Under the password field is a small rectangular button labeled "REGISTER". At the bottom right corner, the word "LOGIN" is written.

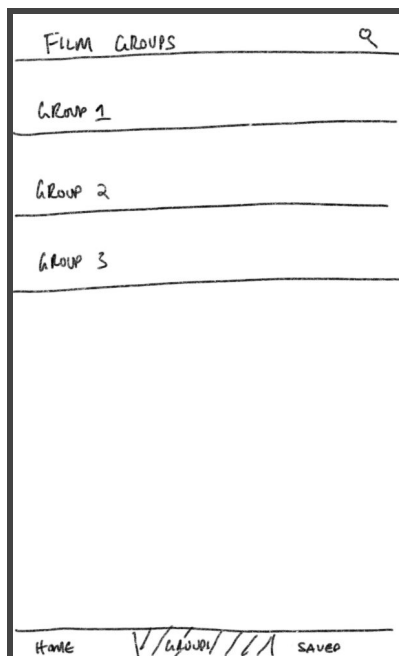
Home Screen

First view the user sees. Includes a feed of the newest movies and the ability for users to search and move to other screens such as groups or saved.



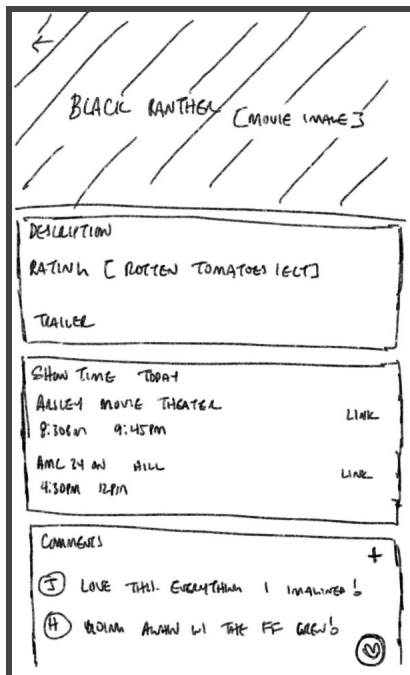
Group Screen

Shows a list of groups that the user can click to start talking with others. Also allows the user to search for specific groups.



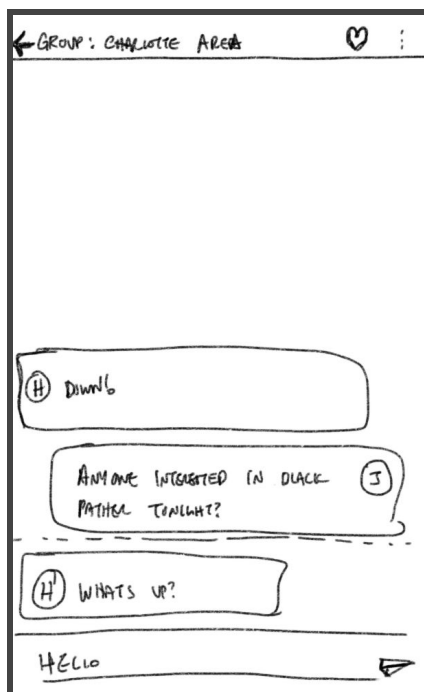
Movie Details Screen

Shows details regarding the specific movie the user is interested in. Allows user to interact with others via a rating, comment and saving button. Also will pull in data regarding the users location and show local theaters with the showtime.



Group Detail String

Group chat specific to an area based on the location provided by the user. Favorite button to save this group as a favorite within the saved screen.



Key Considerations

How will your app handle data persistence?

Most of the movie and theater data will be pulled from the Fandango API directly. Some of this data will be saved to a Firebase realtime database. In addition, account related information and group messages will all be saved within Firebase. Shared preferences will save settings related data including the user's ZIP code, most recent location, and settings preferences.

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

Glide

- Image loading to easily load images from URLs parsed from the Fandango API.
- <https://github.com/bumptech/glide>

Retrofit

- HTTP client to easily request data from the Fandango API.
- <http://square.github.io/retrofit/>

GSON

- Data object conversion easily convert between JSON and object models, used together with Retrofit.
- <https://github.com/google/gson>

Timber

- Testing and logging purposes.
- <https://github.com/JakeWharton/timber>

Espresso

- UI automated testing to test the interactions between the UI and different components.

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

Firebase

- Realtime database, storage and authentication.

Google Places

- Used to determine the ZIP code of the user based off of coordinates.

Required Tasks

Task 1: Project Setup

Setting up the application and required libraries and services.

- Create base application and upload to github
- Setup required libraries in the app
- Setup Firebase and Google Places services
- Create Fandango dev account

Task 2: Implement object models

Designing and developing models that'll be used.

- Design models for messaging, movies and theater
- Implement models
- Implement serialization for these models

Task 3: Implement API Calls

Develop interface to interact with API using Retrofit

- Design interface to pull data from the Fandango endpoint
- Create NetworkUtil class act as the coordinator class for network requests
- Create methods to pull search, theater, and movie data.
- Design database layout for Firebase to save account, group, messaging and saved information
- Implement FirebaseUtil class so that the app can utilize one central object to access Firebase data

Task 4: Implement UI Layouts

Design the layout associated with sketch design and utilize fake data.

- Implement TabLayout as the main interface with search bar
- Implement fragment views within the tab layout including the Home, Group and Saved fragment
- Implement view of Messaging Activity
- Implement view of Movie Activity
- Implement view of Login Activity
- Implement view of Register Activity
- Implement view of Loading Activity

Task 5: Implement account features

Account login and register features

- Implement the ability for users to login and sign up (register)
- Prevent users from certain tasks such as commenting, rating and creating groups without an account

Task 6: UI and Data Linking for Home Activity

Link between the Home Activity fragments and API/Firebase data.

- Show cardview UI of recent movies
- Enable searching in the toolbar of movies
- Link each cardview to the MovieActivity

Task 7: Implement Movie Activity Features

Feature set within the movie activity

- Allow users to rate the movie
- Request location information or ask user to put in ZIP code if that's not already available
- Show nearby theater time and information about the movie to user
- Implement feature to allow users to comment on the movie
- Implement feature to save the movie so the user can access this later

Task 8: Implement Group Feature

Feature set within the group activity

- Implement messaging feature for anyone to join a group and post comments
- Implement feature to allow users to create group chats for anything and save them
- Implement feature to allow users to save groups so they can access them easily

- Implement search feature to allow users to find groups that they are interested in

Task 9: Implement Saved Features

Saved fragment features

- Pull down information from Firebase on the saved movies and groups
- Show them in CardView using the same layout within the Home Fragment and Group Fragments layout.

Task 10: Implement Transitions

Create transitions going between activities and fragments

- Design and implement sliding transition between fragments on the home page
- Design and implement transition from fragment to activity (Home Fragment -> Movie Activity and Group Fragment -> Group Activity)
- Design and implement transition for searching

Task 11: Implement Testing

Test the application and create test cases to confirm the data and UI is functioning as expected.

- Setup testing to test HTTP request data, confirm error handling
- Setup testing to confirm proper intents from the Home Fragment and Group Fragment
- Setup testing to confirm that data is being saved within the Firebase database as expected