How to Use this Template
1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: joselukeolivares

Gamers IGBD

Description

Gamers IGDB is the app for everyone that loves games. If you love games as well, we are the gaming app for you. You can search info of your favorite video games : age_ratings, category, cover, dlcs, game_engines,popularity, rating, release_dates and more! All the data is from IGDB a video game database, intended for both game consumers and video game professionals alike. IGDB is operated by Twitch.

Intended User

Gamers IGDB is the app for everyone that loves games and love search data for games.

Features

List the main features of your app. For example:
- Saves information of the videogames marked like favorites.
- Widget with the image of the last video game viewed.
- Search with video game' name.
- Video Games info from IGDB API: description, videos, reviews and screenshots.
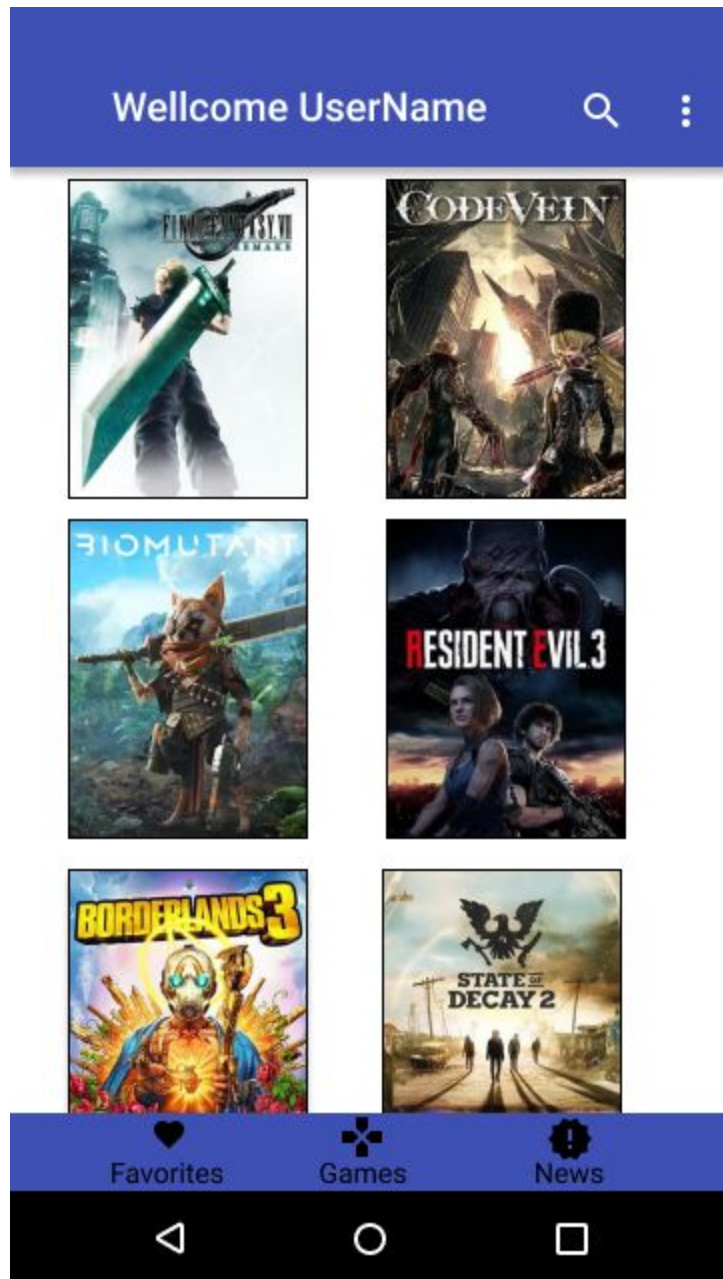
User Interface Mocks

Screen 1: Sign in with Google account
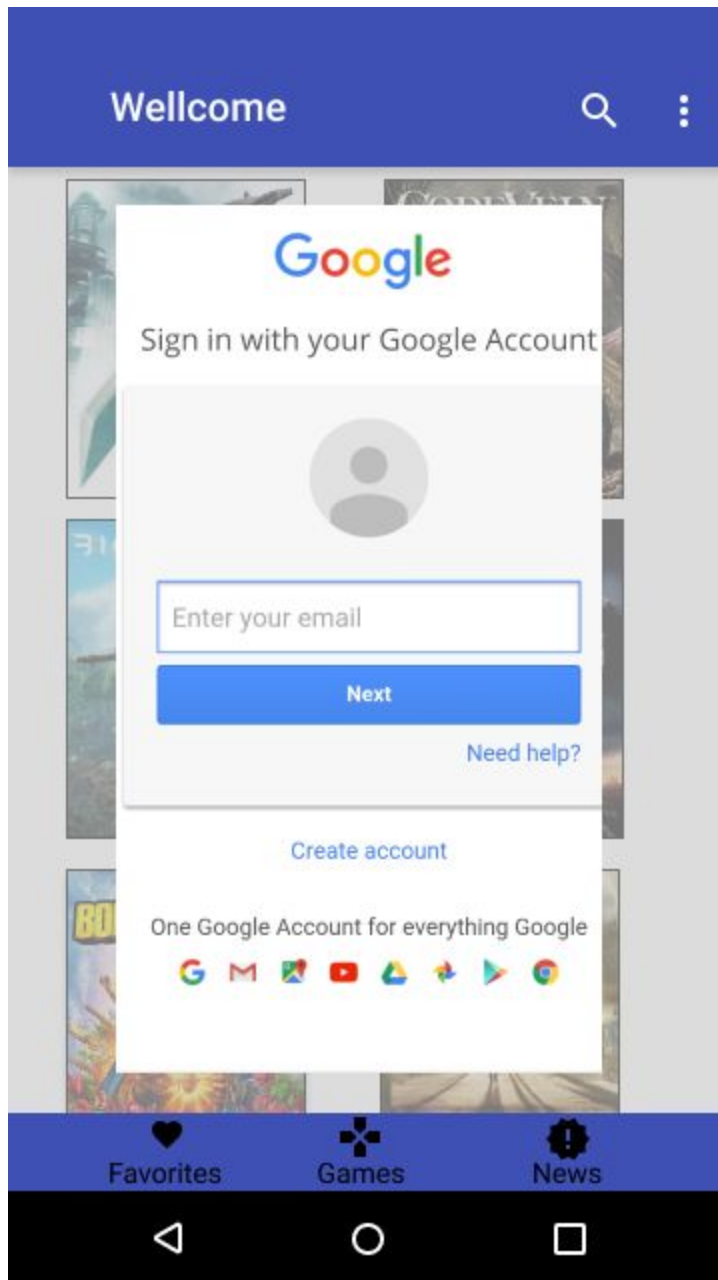
For use of this App the user must be logged with a Google's account. If the user is not logged before, the app will prompt the Sign in for data of the user. Once the user is logged the user can use the app.

## Screen 2: Main Activity with list of video games

Once the user is logged the app will show a posters of popular video games (Fragment Master) from IGDB API. The user can select a video game poster to open an Activity where the app show the description of the game.

The user can select in the bottom menu the differents options for show data:
- Favorites: Will show only the posters and screenshots of the games marked liked favorite by the user and removing thes rest of the games.

- Games: Will show all the games marked liked favorite by the user and the rest of the games from the IGDB API.
- News: The IGDB share all kinds of news registered by the users like new releases,coming soon, new_trailer and more.
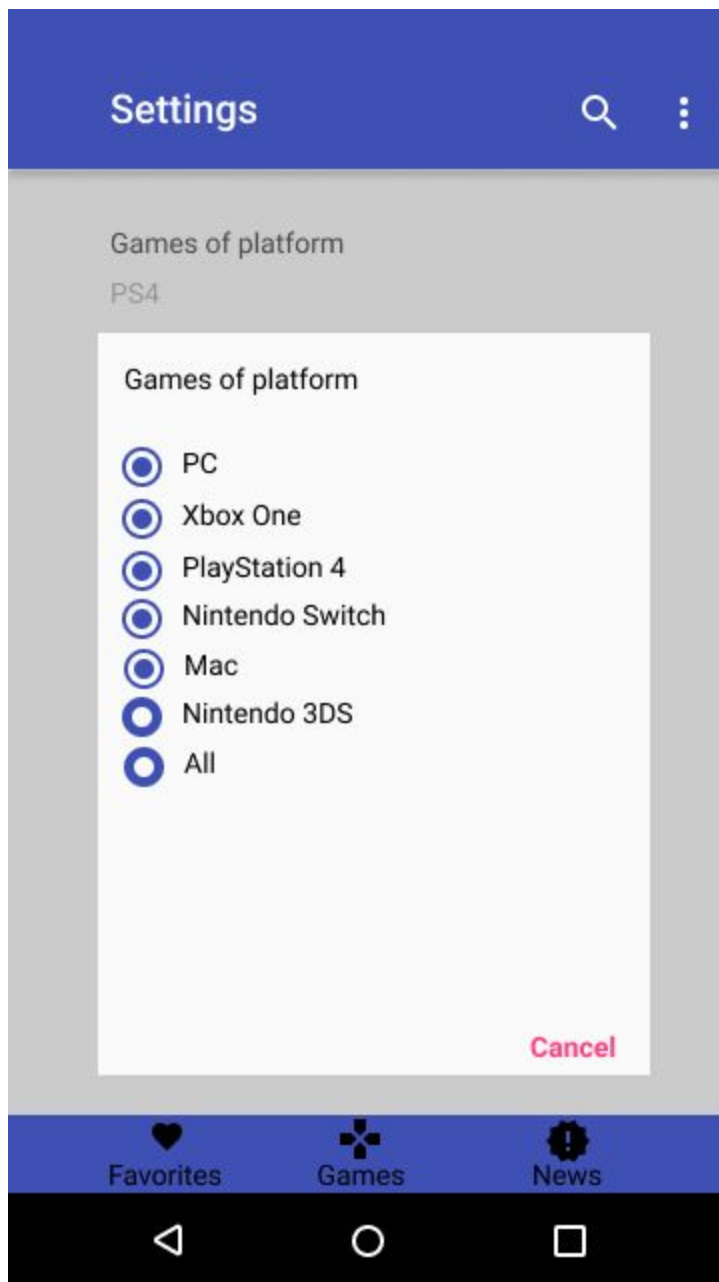
Screen 3: Settings
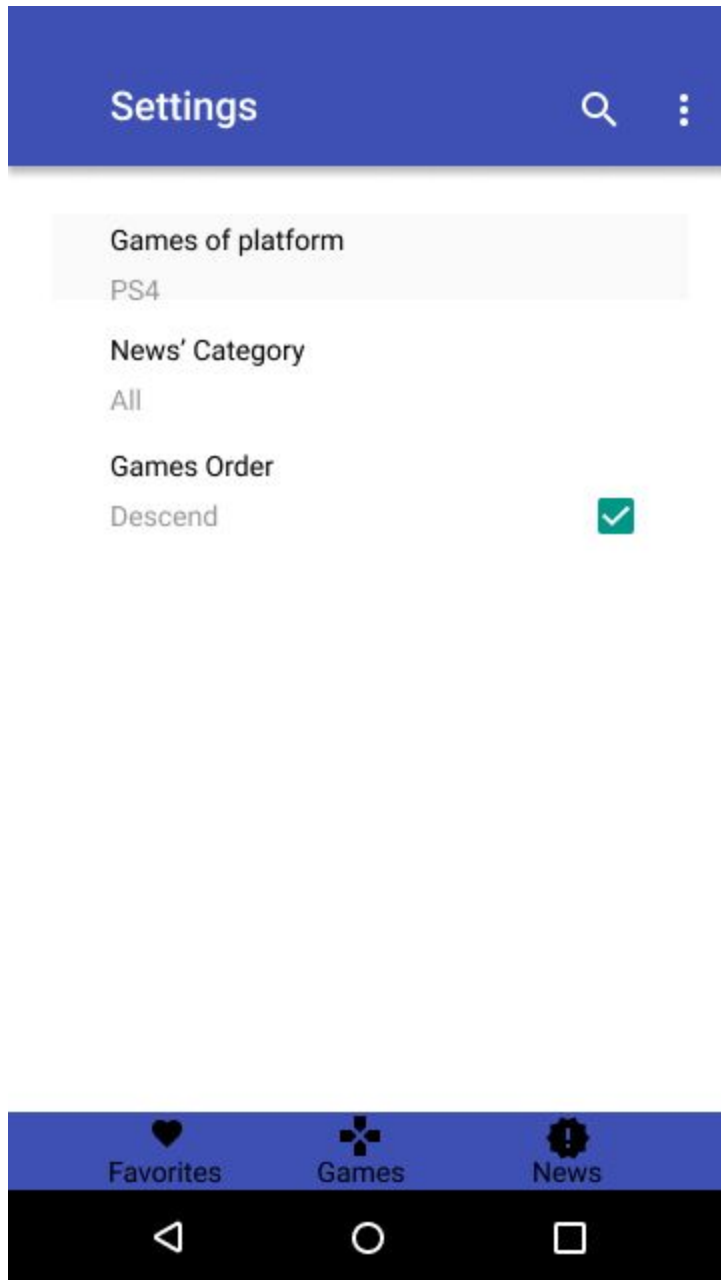
The user can set the settings for limit the result of search of the video games every time that the app make a request to the API:

- Games Platform: The user can select the platform for specific games that he need get data.
- News Category: The app make a request for feeds for show news the video games.
- Game's Order: The sort of the video games showed can be ascendant or descendent by the popularity.

Screen 4: Setting Games of Platform

Screen 5: Setting News Category
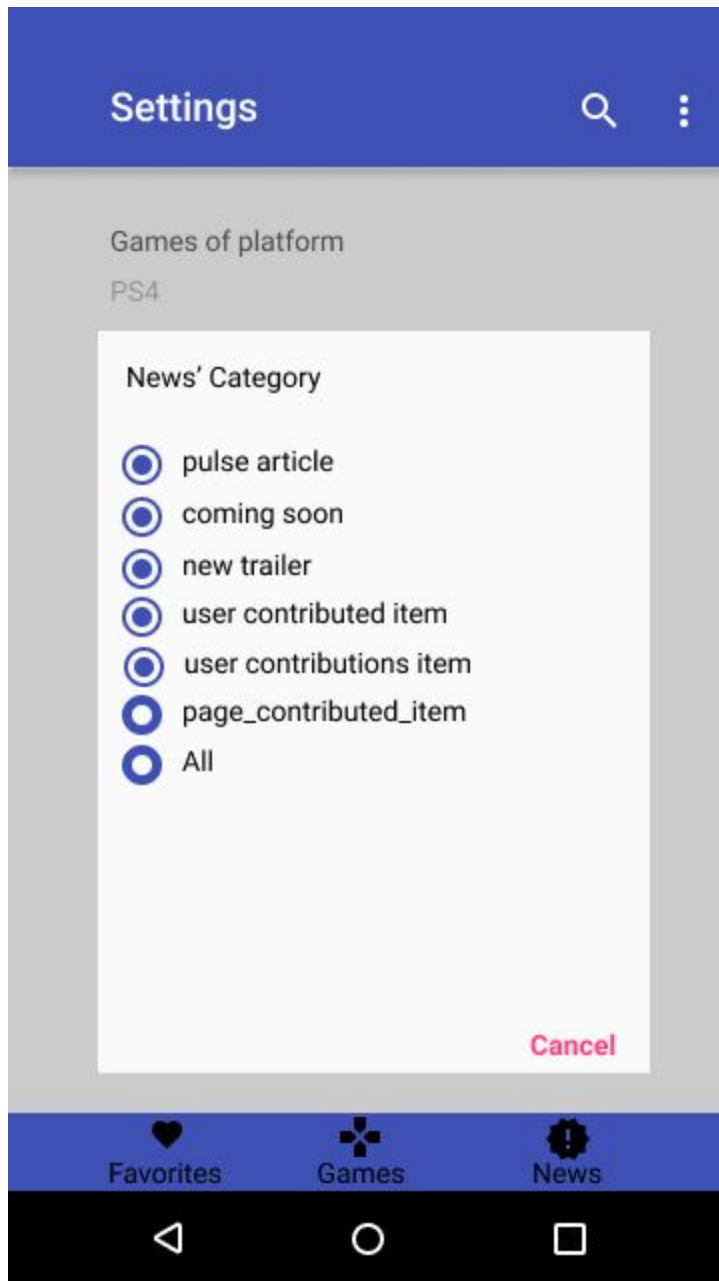
Screen 6: Game Detail


Once the user select a movie poster from the Main Activity or other, the app shows a detail data of the video game:

- Title of the game.
- Date of release.
- Company of development.
- Members ratings.

- Critic ratings.
- Genre.
- Platforms.
- Description of the game.
- Number of members want to play this game.
- Number of members are playing this game.
- Number of members played this game.

The user can indicate with fab button if want store the data related of the game and view it in the Activity Favorites where only can see posters and screenshots of the favorite games.

Screen 7: Game Detail Scrolled
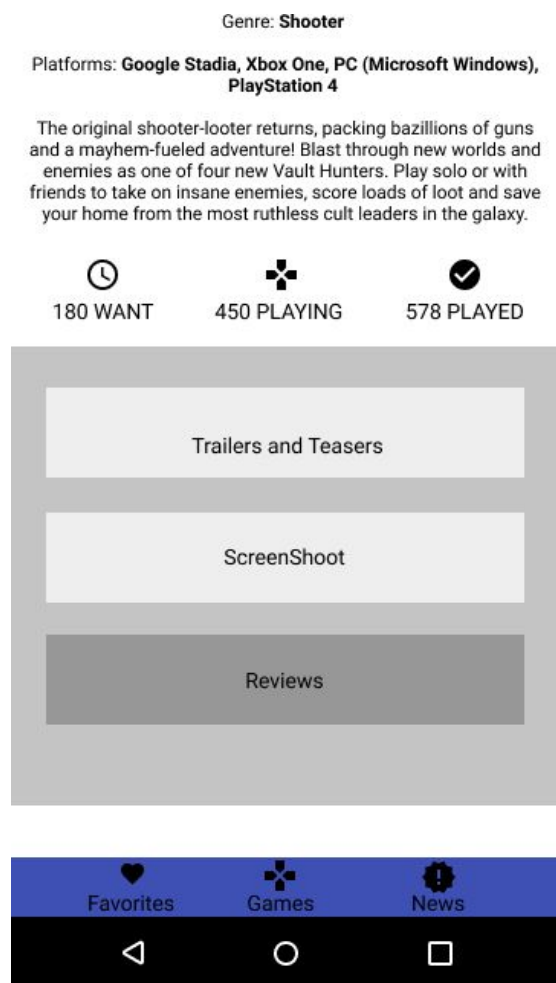

The user can scroll in the screen 6 to down for select if he want to see other content related:
- Trailers and Teasers: When user selects this button the will open an Activity where recyclerview with the list of the videos related with game.
- ScreenShoot: When user selects this button the will open an Activity where recyclerview with the list of the images (screenshots) related with game.

- Reviews: When user selects this button the will open an Activity where recyclerview with the list of the reviews written by members about the game.

Screen 8: Related Videos
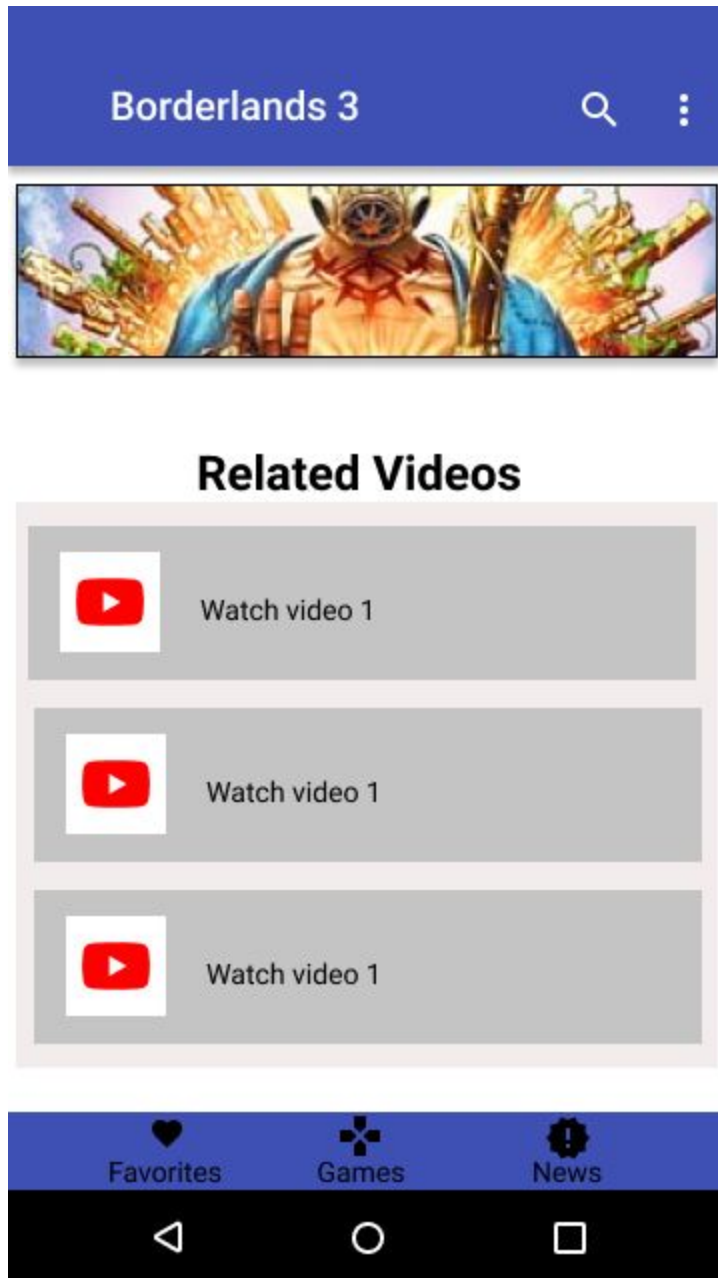
It the user select the option "trailers and teasers" the app will start this activity, where recyclerview show the list of the videos related with Youtube icon for indicate the user that the video will be showed in a Youtube App.

Screen 9: Screenshoot
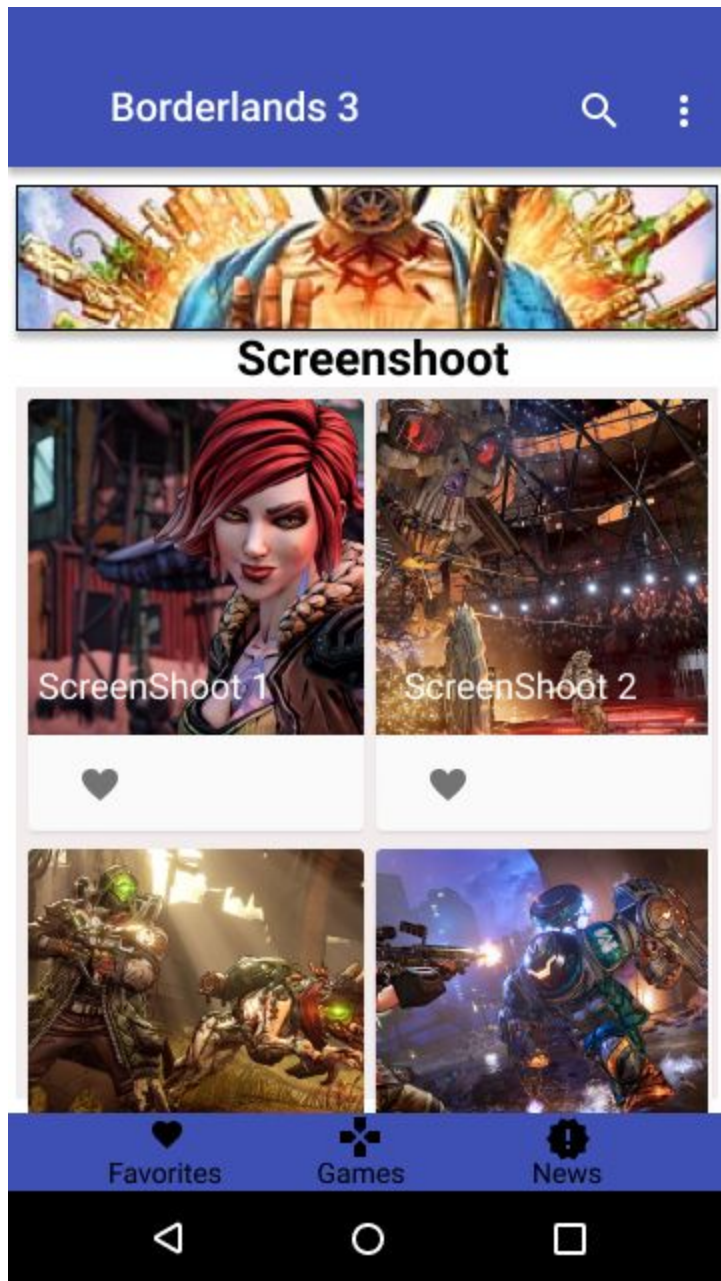
It the user select the option "Screenshoot" the app will start this activity, where recyclerview show the list of Cardviews with the images of the screenshots.The user can indicate the images that users like, the app will store the data related with the image (URL) and can show it the images in the Activity Favorites when user select the option in the bottom menu.

Screen 10: Reviews

If the user select the option "Reviews" in the Screen 6 (Game Detail) the will show a recyclerview with the Reviews that members of IGDB have written about the game. Each viewholder will show the date was published, the rate by the member owner of the review and rating translated by IGDB. The user can select a Review for more detail.
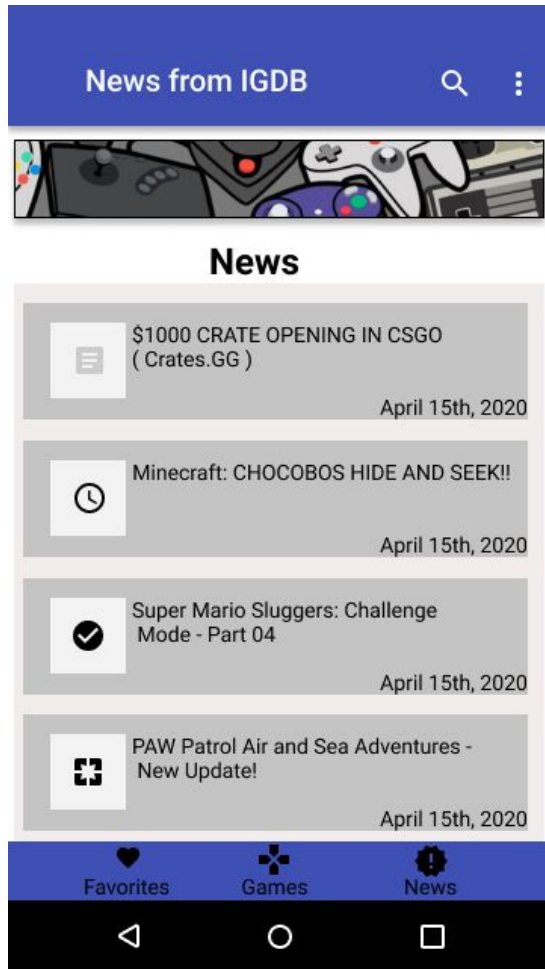
Screen 11: Reviews Details

If the user select one review from the list in Screen 10 (Reviews), the app will show more details of the review:

- Rate with numbre by the member of IGDB owner of the review.
- Rate with text by the member of IGDB owner of the review.
- Published date.
- The main content of the review.
- Positive points (if exist).
- Negative points (if exist).

Screen 12: News

If the user use the News option in the Bottom Menu the app will open an Activity with RecyclerView where each View Holder will show icon of the category of the Feed, title and date when the feed was published. If the user select one New will be redirect to the official web site where can find all the details of the feed.

Screen 13: Main Activity (Tablet version)
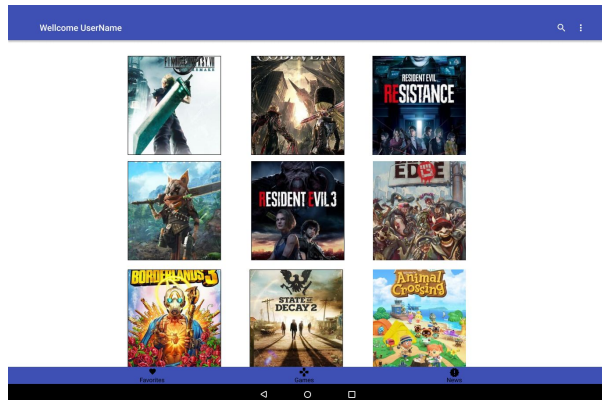
In tablet version, the recyclerview that show the posters of the video games has 3 columns.

Screen 14: Game Detail (Tablet version)
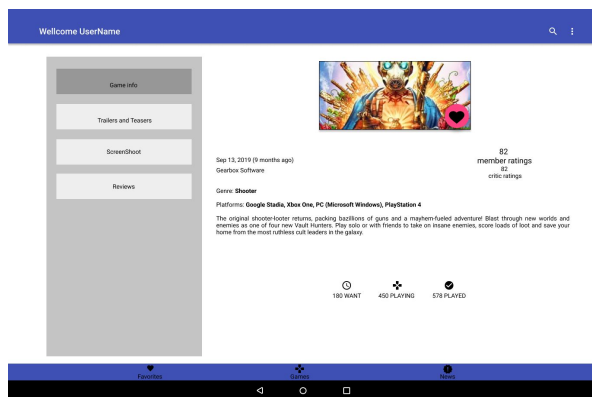
In tablet version, once the user select a video game detail the app will show similar content like Screen 6, the principal difference is that the options for more information about the game allow the user use the same Activity for display in Fragment the videos, screenshots and reviews:

Screen 15: Related Videos (Tablet version)
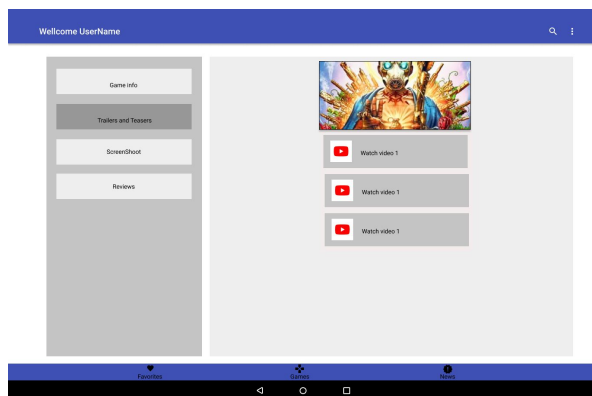
Screen 16: Screenshoot (Tablet version)
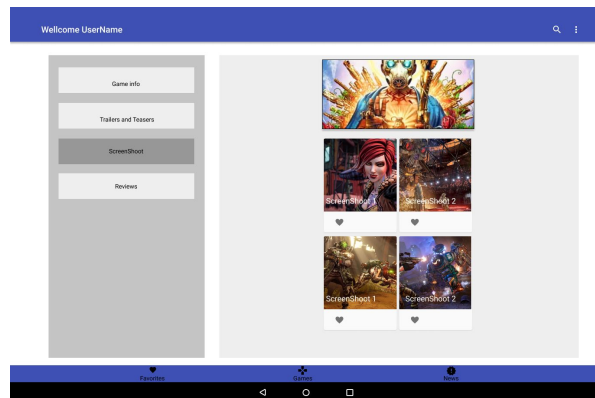
Screen 17: Reviews(Tablet version)
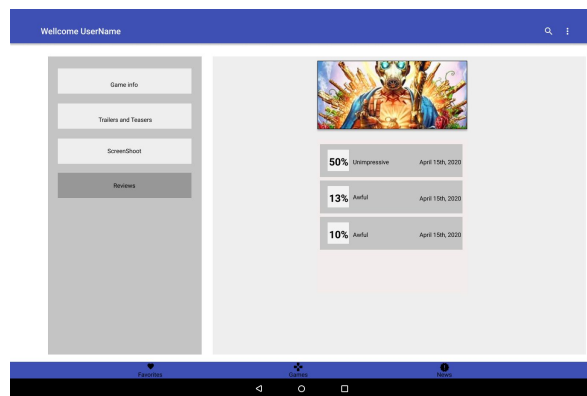
Screen 18: Reviews Detail (Tablet version)

Screen 19: News (Tablet version)
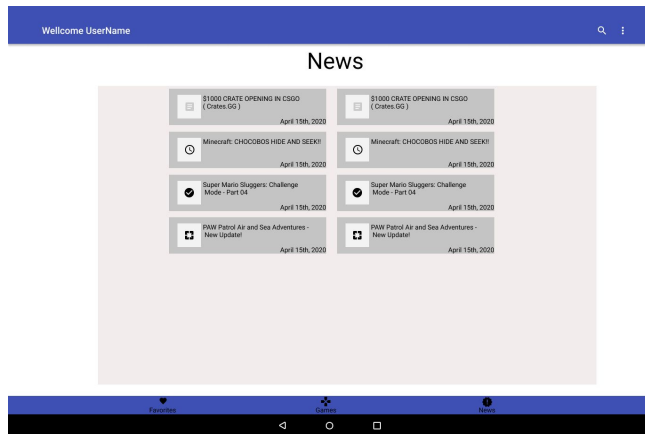
Screen 19: Favorites (Tablet version)
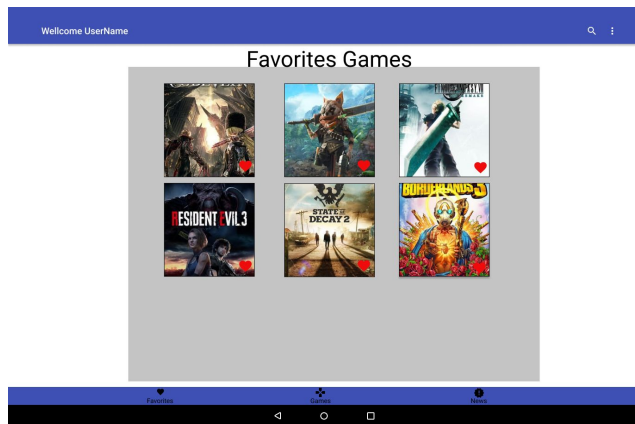
If the user select the option Favorites in Tablet version the will display in three columns the favorite video games marked by the user. In mobile version the recyclerview will have two columns.

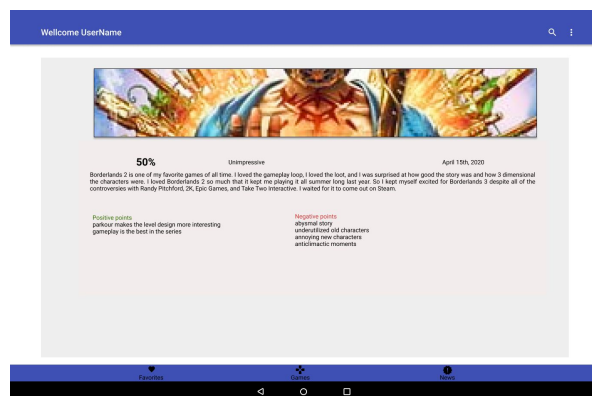**Screen 20: Widgets on Home Screen(Mobile and Tablet version)**
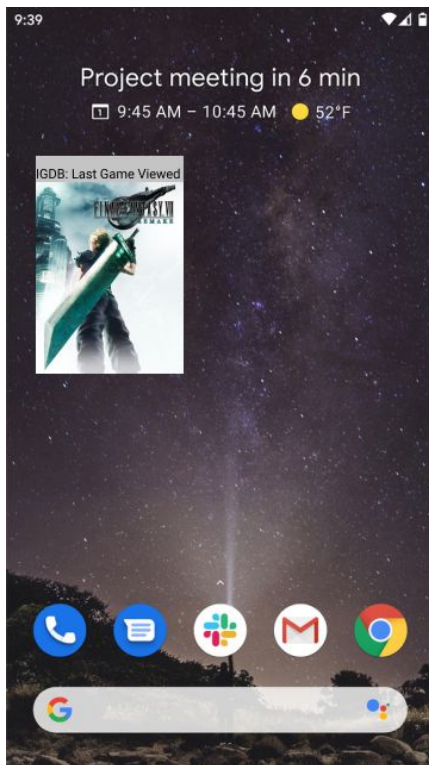
The widget will show the poster of the last video game viewed by the user.

Key Considerations

How will your app handle data persistence?

The app will use Content Provider and Live Data.

Describe any edge or corner cases in the UX.

- The content hasn't loaded yet.
- After loading content, there are no results.
- Error while loading content.
- Buttons are pressed, selected, or disabled.
- Content is too heavy or the content is lacking.
- No network is available.

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

- Picasso: Display images in imageview.
- Volley: Easy way of get data from IGDB API.
- Firebase: Easy way to Sign in for the users with Google Accounts.
- Play-services-ads: Free version will have ads and paid version will be free of ads.
- Material Design: better experience for use with the components.
- Room Persistence Library: Save data of games in a local database using Room.
- RecyclerView: Dynamic list for show info of the games.
- Fragment: Ensure the app looks its best across all device types & screens.

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

- Play-services-ads: Free version will have ads and paid version will be free of ads.
- Firebase Auth: The users can sign in with the help of this service.

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

First you need to get an Api key for use the request of data to IGDB API:

- https://www.igdb.com/api

Once you get the API key store it in Resource Strings file in the element called: API_key:
<string name=API_key> YOUR_API_KEY </string>

- Use add Volley library and use it for a request of data of the video games.
- Use asyncTaskLoader to ask for request the data with Volley.

Once you added API key and make a request to the API add the Firebase service for Sign in with Google Account:

- Add Firebase service to the app gradle.

- If you have not registered before in Firebase, get a free account for register this Android app, follow all the steps for registering the app and download the Json file that the project need for the correct use of the Auth with Google.
- Main Activity, Games Detail, Videos Related, ScreenShot, Reviews and News Activities will use AsynctaskLoader for use Volley and get the data from the API of IDGB.
- From Main Activity must send data of Game like extra in the intent, in that way the Activity Game Detail will make a request with Id of the game.
- Once Firebase notify you that the app connect successfully with Firebase service, add the following code and make the required changes to the MainActivity: https://firebase.google.com/docs/auth/android/custom-auth?hl=en
- You must be sure that the login will be the first work that the app will do to validate the identity of the user.

.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:
- Once the user is logged he could see the posters of the video games in a RecyclerView. The order of the posters depends of the popularity of the games. By default must be descendant, and the user could change that with settings in the top menu.
- The recyclerview that will contain the posters of the games will be contained in Master Fragment.
- When the user select the poster of the game in the viewholder the app must apply effect transition to the Activity where the app show game detail.

Task 3: Add Flavors

Implement Google Play Services:

- Create Build Variant: Add Free and Paid Flavors to the app, where Free version will show ads with Play-services-ads, and Paid version will not show ads.

Task 4: Add Permissions

The app will need Internet connection and Handle Error Cases:
- If the device has not Internet connection you must notify the user retry again or later.

- Some request to the IGDB API will give empty data or some properties will be null, you must handle empty or null error avoiding the app crash.

Task 5: Add Intent for Youtube and Web Page

Game details, news (feed) and video relates to the video games allow user use the list for show the elements of the detail of the game, so you must guest it the content allow user open a video on Youtube app or Web Browser for more detail like official site where the feed comes:

- In the Fragment where the recyclerview contains video related, the app must use an Intent to open a Youtube app.
- In the fragment where the recyclerview contains news (feed), the app must use an Intent to open a Web Browser and be redirected to the site.

Task 6: Save data of games in a local database using Room

The Game Detail Activity allow the user mark a video game like favorite, when the user indicate like favorite or dismark the app must use Room for store data of the video game:

- The information available about the game: name, summary, rates, platforms, numbers of members who want to play, are playing and played the game.
- Only save the data downloaded with the API and not store the images, only the url for access to screenshots.

Task 7: Widget

The app must save the url of the poster of last video game viewed by the user, and allow the user use a widget where the will show the last video game viewed.

Task 8: All Activities with Fragment

For better experience to the user, must use fragments when the app show:

- Related videos.
- Screenshoot.
- Reviews.

Add as many tasks as you need to complete your app.

Submission Instructions
- After you've completed all the sections, download this document as a PDF [File →
  Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:
- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"