

[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: Bloody-Badboy

Game Database

Description

Using this app it easier for game lovers to find popular, top rated and upcoming games.

The app is mobile friendly and easy to use with the following feature:

- Explore games and see their information in details.
- Watch game videos and images.
- Get the latest news about your favorite game on the home screen.
-

Intended User

- This app is for everyone for who loves games and want to keep up-to-date with the latest game information.

Features

- The app is written solely in the Java Programming Language
- The app utilizes stable release versions of all libraries, Gradle, and Android Studio.
- The App refer all the hardcoded strings from the strings.xml file.
- The app enables RTL layout switching to support accessibility on RTL supported languages.
- Explore games by different categories like Popular, Coming Soon, Latest Releases.
- See the detailed information about any game by clicking.
- See the game videos, images, latest news, and other information.
- Add games to your favorite and get the latest news about them.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    // Android Support
    implementation 'androidx.appcompat:appcompat:1.0.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.recyclerview:recyclerview:1.0.0'
    implementation 'com.google.android.material:material:1.0.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'

    // LeakCanary
    debugImplementation 'com.squareup.leakcanary:leakcanary-android:1.6.1'
    releaseImplementation 'com.squareup.leakcanary:leakcanary-android-no-op:1.6.1'

    // Timber
    implementation 'com.jakewharton.timber:timber:4.7.1'

    // Glide
    implementation 'com.github.bumptech.glide:glide:4.8.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.8.0'

    // Retrofit
    implementation 'com.squareup.retrofit2:retrofit:2.4.0'
    implementation 'com.squareup.retrofit2:converter-moshi:2.4.0'

    // RecyclerView SnapHelper
    implementation 'com.github.rubensousa:gravitysnaphelper:1.5'

    // Stetho
    implementation 'com.facebook.stetho:stetho:1.5.0'

    // Android architecture component
    implementation "androidx.lifecycle:lifecycle-extensions:2.0.0"
    implementation "androidx.lifecycle:lifecycle-viewmodel:2.0.0"

    implementation "androidx.room:room-runtime:2.0.0"
    annotationProcessor "androidx.room:room-compiler:2.0.0"

    // Firebase
    implementation 'com.google.firebase:firebase-core:16.0.4'
    implementation 'com.google.firebase:firebase-ads:17.0.0'
    implementation 'com.crashlytics.sdk.android:crashlytics:2.9.5'

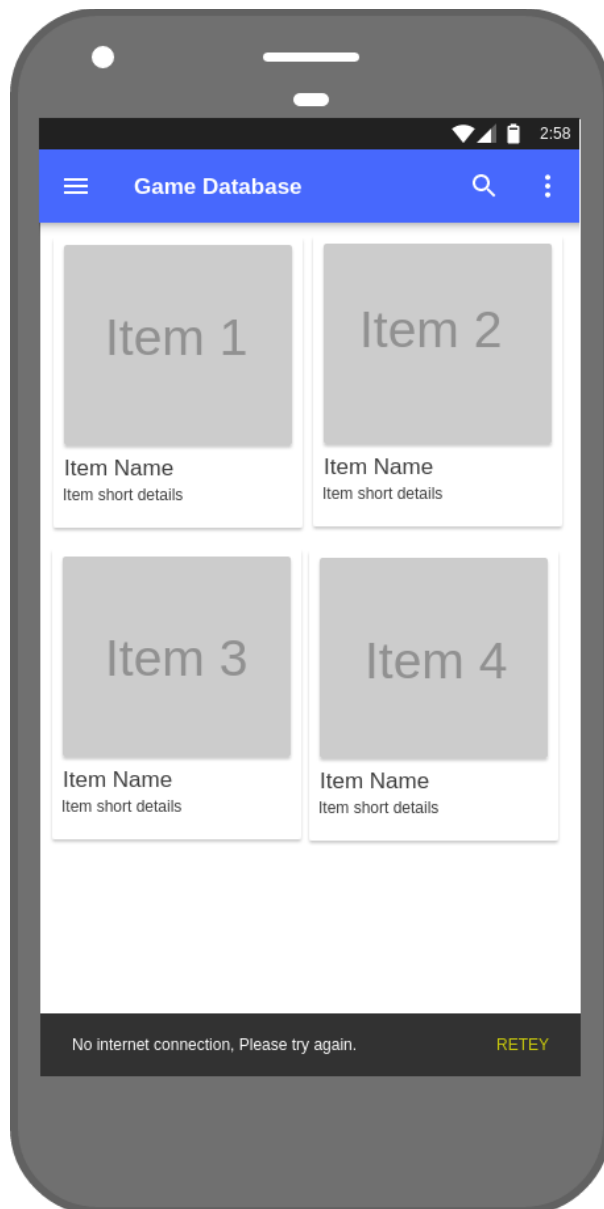
    // Shimmer
    implementation 'com.facebook.shimmer:shimmer:0.3.0'

    // Testing
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test:runner:1.1.0'
    androidTestImplementation 'androidx.test:rules:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-contrib:3.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-intents:3.1.0'

    implementation 'androidx.test.espresso:espresso-idling-resource:3.1.0'
}
```

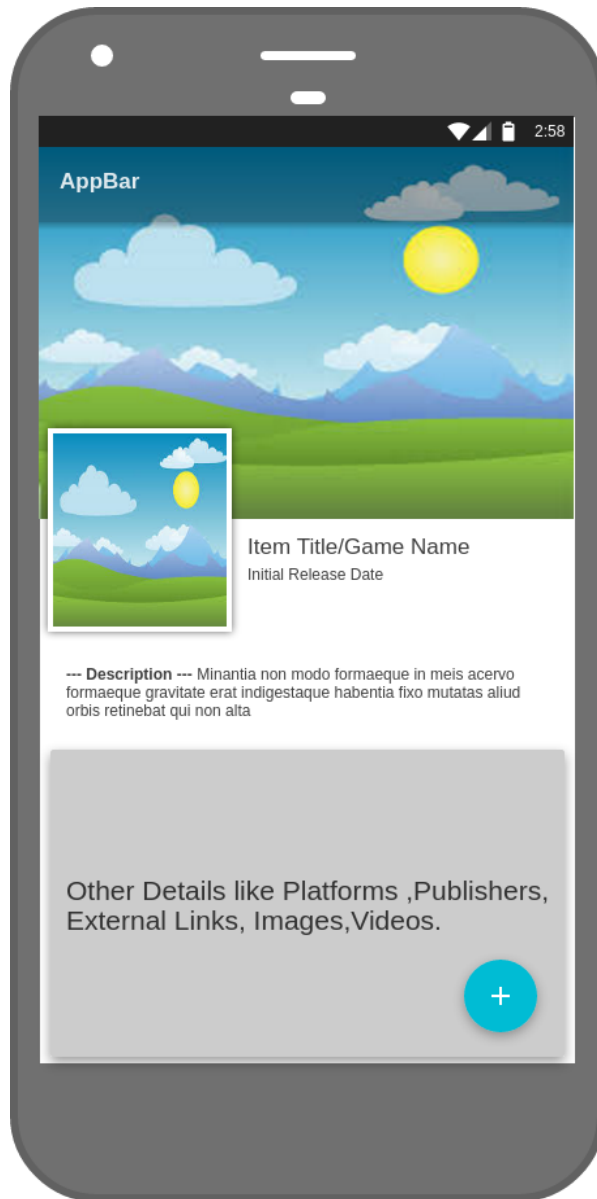
User Interface Mocks

Screen 1

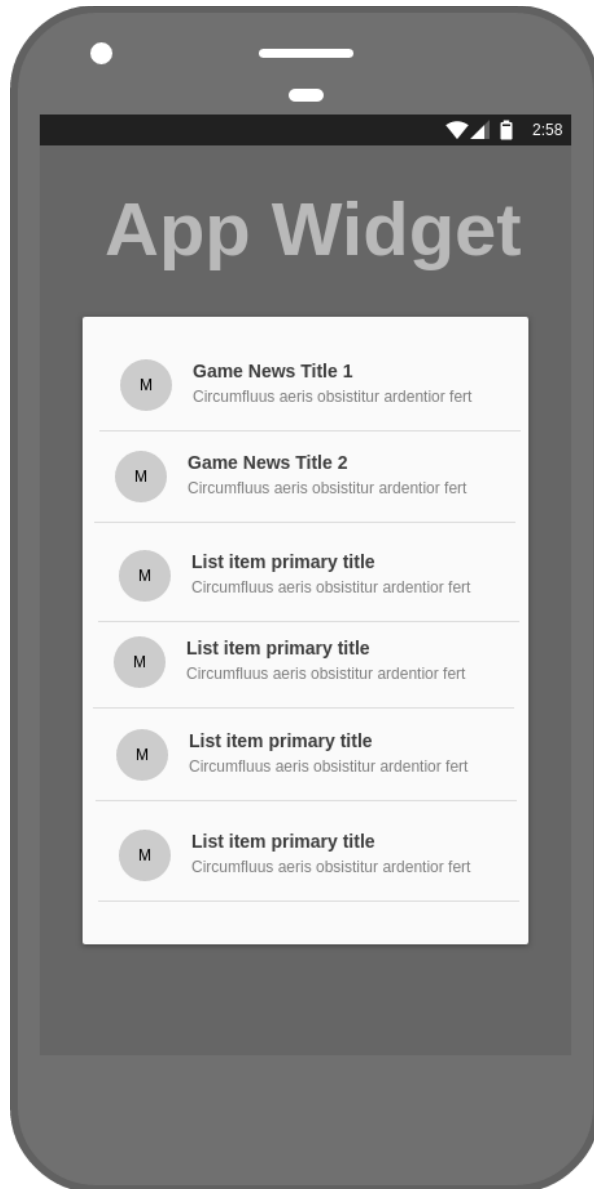


This is the home screen where the user can view the list of games, from the menu user can switch between Top Rated, popular, upcoming games list. More item will be displayed as users scrolls down. Using the search button the user can search game. On clicking on any list item used will be navigated to the details screen.

Screen 2



The user is navigated from the home screen to this screen. The screen shows detailed information about the selected item. The user can scroll down and view game videos images, release date, and other various information.



App widget shows the latest news about users favorite games.

Since this is the early stage of the app, The color scheme may be changed. I will try to add more features like Individuals who are in the video games industry.

Key Considerations

How will your app handle data persistence?

- The app will use the Room persistence library for data persistence.
- The app will use JobScheduler to schedule data update from web service to a local database.
- The locally cached data will be displayed to the user only when he/she is offline. This data will be also used to populate app widget.

Describe any edge or corner cases in the UX.

- What will happen if there is no internet connection?
A snack bar will be displayed saying “No internet connection, please try again”, with action “Retry”.
- How will the app respond if the data from the API fails to load?
A snack bar will be displayed saying “Unexpected error occurred” with action “Refresh”.

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

- **Android architecture component** for managing your UI component lifecycle and handling data persistence
- **Glide** for image loading
- **Android design support** libraries
- **GravitySnapHelper** for RecyclerView snapping
- **LeakCanary** for memory leak detection
- **Retrofit** as a rest client
- **Moshi** for parse JSON into Java Objects
- **Timber** for logging
- **Stetho** as an interceptor
- **Shimmer** for loading Shimmer effect
- **Firebase crashlytics** for crash analysis

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

- Interstitial Ads when the user navigates from splash screen to the home screen.

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

- Setup libraries.
- Setup MVVM architecture using Android Architecture Components
- Setup different layers, like the data layer, domain layer.
- Setup a Remote and Local data source inside the data layer.
- Design the flow of the project.
- Implement Google Ad Services.

Task 2: Implement UI for Each Activity and Fragment

- Build the splash screen.
- Build UI for each activity.
- Build a fragment(view) for each activity.
- App uses a LiveData to moves its data to its views.
- Implement shared element transition and other layout transition.

Task 3: Implement App Widget

- Implement App Widget.

Task 4: Implement Google Ad service

- Implement Interstitial Ad in UI.

Task 5: Design for Tablets

- Create detail activity layout for the tablet.
- Enable layout mirroring.
- Check compatibility with older devices.

Task 6: Testing and Debugging

- Design Test cases and perform each test case accordingly.
- Analyze the bug.

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**”