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: addison-lima

Gaming Backlog

Description

With holiday sales, subscription services and bundles aplenty, it's easy to end up with *hundreds* of games you haven't even installed yet, much less played.

Sitting with a massive video game backlog can be frustrating and feel like a real problem, but there are constructive ways to deal with this first-world problem, so you spend more time playing games than agonizing over how many games you still have to play.

With Gaming Backlog app, you will be able to register games you own, discover new games, select games that you are currently playing and games that you already beat.

Intended User

For people who love games and want to manage their current gaming backlog.

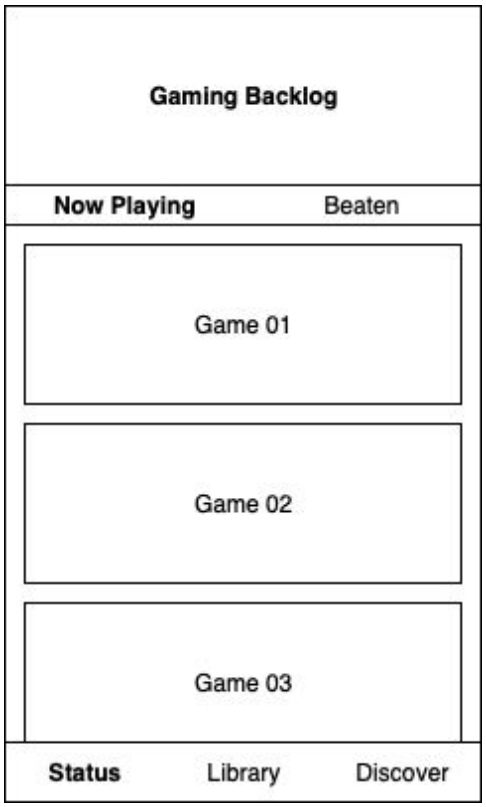
Features

- Add games to your Library;
- Select games you are currently playing;
- Select games you already beat;
- Discover new games.

User Interface Mocks

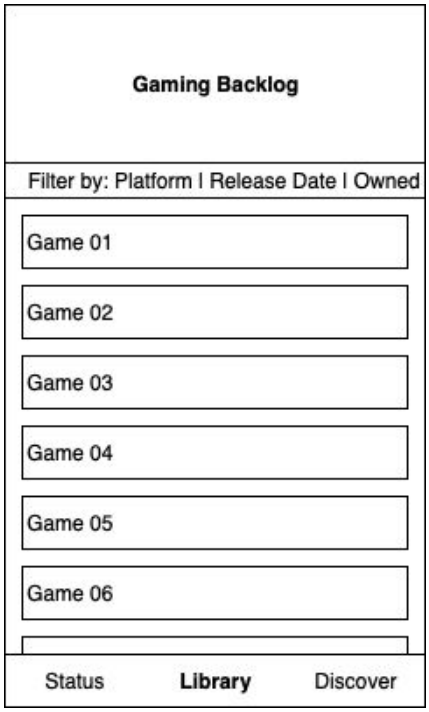
Screen 1

Status Tab lists *Now Playing* and *Beaten* games:



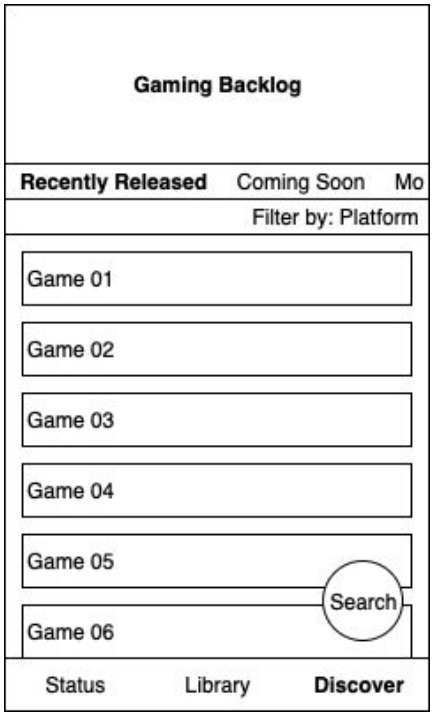
Screen 2

Library Tab lists *Owned* games. It provides filter options:



Screen 3

Discover Tab lists Recently Released, Coming Soon and Most Anticipated games. It provides filter options and a search button:

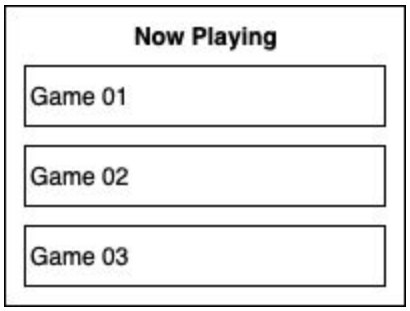


Screen 4

Game details with trailers, *Own*, *Playing* and *Beat* buttons:

Widget

A list of current playing games:



Key Considerations

Which programming language will be used for this project?

Java language will be used for development.

Will localization and accessibility support be considered?

App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts. App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

How will your app handle data persistence?

Use of shared preferences for filter options. Use of SQL database for games library.

Describe any edge or corner cases in the UX.

The user can play or beat a game that is not owned by him. In this case, this game is shown in *Status* tab as *Now Playing/Beat* game, but does not show in *Library* Tab. A not released game cannot be set as *Own*, *Playing* or *Beat*, as it is not available yet.

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

Use Room for Database transactions, along with Lifecycle Extensions, such as LiveData and ViewModel. Glide to handle the loading and caching of images.

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

Use Admob Google Play Services. Use Retrofit to retrieve games from IGDB (https://www.igdb.com/discover).

App utilizes stable versions of all libraries, Gradle and Android Studio:

Admob Google Play Services	Version 18.3.0
Android Studio	Version 3.4.2
Glide	Version 4.8.0
Gradle	Version 3.4.2
Lifecycle Extensions	Version 1.1.1
Retrofit	Version 2.4.0
Room	Version 1.1.1

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

- Implement Game API service to retrieve games from https://api.igdb.com;
- Create GameModel.

Task 2: Implement UI for Discover Tab

- Create GameItem layout;
- Retrieve and list *Recently Released* games;
- Retrieve and list *Coming Soon* games;
- Retrieve and list *Most Anticipated* games;
- Add filter options to the results.

Task 3: Implement Search

- Create a dialog to input keywords;
- List the results.

Task 4: Implement UI for Game Details

- Retrieve details from a single game;
- List trailers and send intents to play selected trailer;
- Add buttons to set current game as *Own*, *Playing* and *Beat*.

Task 5: Implement UI for Library Tab

- List games owned;
- Add filter options.

Task 6: Implement UI for Status Tab

- List *Now Playing* games;
- List *Beat* games.

Task 7: Implement Widget

- Create NowPlayingWidgetProvider;
- Update widget by sending Intents when Now Playing list is updated.

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**"