

SDA Project

Steam

Introduction:

A Steam replica software designed to manage a library of games as a project for Software Design & Analysis course.

About the Database:

This project uses MongoDB Cloud as its database, storing users, game information and the user-game mappings. The store is a pre-set document in the database that contains the

information of games which a user interacts with. Since the database stores JSON objects, they are mapped into POJO at each fetch with respect to each class. For dependency inversion, there are classes which provide exclusive layer functionalities and act as getters and setters of that layer. For the text database, we used .json files that store information similar to the other database. These JSON objects are serialized/deserialized as well into Java Objects using Google's GSON library.

Roles

- **Waleed Ahmad (18L-1282)** *Front-end GUI Logic and Database Interaction Logic*
- **Zulfiqar Chaudhry (18L-1037)** *Business Layer Logic and Class Logic*
- **Daud Mazhar (18L-0919)** *Class Diagram, Front-end and Documentation*
- **Sameer Ali (18L-1150)** *Business Layer Logic*

Use Cases

- ViewWallet
- BuyGame
- DownloadGame
- LaunchGame
- RemoveGame
- ViewStore
- ViewLibrary
- GetGameInfo
- CreateAccount
- UpdateWallet
- BrowseGames

- DisplayUserInfo

Complications:

- 1. JSON to POJO and POJO to JSON Conversion (Resolved):** We faced an issue of the mapping of JavaScript Object Notation to Plain Old Java Object and the other way around. We resolved this issue by using PojoCodecProvider registry for the Mongo Database and Google's GSON for the text-based database.
- 2. Collecting Game Data (Resolved):** The issue was to gather game data for building the store database. This was resolved using a game DB API (known as RAWG.io)

Class Diagram:

