Section 1 - Project Description

1.1 Project

This project will be called PlayPal.

1.2 Description

The project will be an application that allows users to input metrics by which to recommend games from. After this we will utilize an algorithm with SKLearn cosine similarity to recommend content to users based on their profile generated from games they interact with.

1.3 Revision History

Date	Comment	Author

Contents

Section 2 - Overview

2.1 Purpose

The focus of this project is to develop a software that solves a problem for people that play video games at multiple different frequencies. For people that play on weekends after a long work week or people that stream their video game playtime everyday for multiple hours. No matter what anybody that plays video games has a moment where they want to play a new game but aren't sure what game they want to commit to spending their hard earned money on.

2.2 Scope

This module will be developed with a front end tech stack utilizing React.js. The back end will be python flask. I will have to integrate the IGDB api which I already have a key for. It will consist of developing the front end to allow a user log on. Provide input fields to allow users to filter what type of games should be suggested to them. They can like or dislike video games based on past experience, reviews. The model will take in every dislike or like to try and draw patterns towards what games a user might be drawn to. I will utilize PostgreSQL to store user information and parameters necessary for the AI.

2.3 Requirements

User Authentication and Input (Front-End):

- Develop a user-friendly front-end interface allowing users to log in.
- Implement input fields to capture user preferences, such as favorite genres, play frequency, and preferred game features.

Data Collection and Processing (Back-End):

- Integrate the IGDB API to retrieve relevant game data.
- Design and implement a back-end system to process user input and prepare data for the PyTorch Neural Network.

Machine Learning Algorithm:

- Develop an algorithm that will tailor suggestions to a users positive and negative interactions with past games

User Feedback System:

- Implement a feedback mechanism allowing users to like or dislike game suggestions.
- Design a system to collect and incorporate user feedback to continuously improve the recommendation model.

Front-End Design and Interaction:

- Design an intuitive and responsive front-end interface using Python Django.
- Create a smooth user experience for inputting preferences, viewing recommendations, and providing feedback.

Database Management:

- Design a robust database schema to store user data, preferences, and feedback.
- Optimize database queries for efficient data retrieval and storage.

2.3.1 Estimates

#	Description	Hrs. Est.
1	Log in	.5# est
2	Log out	.25# est

3	IGDB Api Query	2# est
4	Machine Alg Input Format	1# est
5	Machine Alg	3# est
6	Training Feedback Retrieval	.5# est
7	Filtration	6# est
8	Encrypt Date	.25# est
9	Decrypt Date	.25# est
10	Database Query	1#est
	TOTAL:	14 Hours # est tot

2.3.2 Traceability Matrix

Cross reference this document with your requirements document and link where you satisfy each requirement

SRS Requirement	**SDD Module**						
	Log In/ Out	Machine Learning	IGDB Api	Input Formatting	Security	Database Query	React
Req 1	x				x	x	
Req 2			x	x			
Req 3		x	x	x			
Req 4		x		x	x	x	
Req 5					x	x	
Req 6							x

Section 3 - System Architecture

The system architecture will follow a client-server architecture. The client will be the front end system which provides input and outputs for filtering what game recommendations should be outputted. The backend will consist of a python flask app with the necessary endpoints + processing of data to provide recommendations to the user.

Section 4 - Data Dictionary

Brief description of each element in this module or a link to an actual data dictionary

(template of a database table description)

Table

User Table

Field	Notes	Type
UserID	Unique Identifier for users	INT
userName	The userName used to login	VARCHAR
passWord	The password used to login	VARCHAR
Email	Email used to login or get emails	VARCHAR
RegistrationDate	The date the users account was made	DATETIME
game_query	the saved query that was generated to run through IGDB	VARCHAR

Game Table

Field	Notes	Type
GameID	Unique Identifier for games	INT
Ratinge	A 0-100 rating of the game	INT
Genre	The genre of the video game	VARCHAR
ReleaseDate	The date the game released	DATETIME
GameModes	The game modes the game has associated with it	VARCHAR
Themes	The themese of the game	VARCHAR

User Game Interactions Table

Field	Notes	Type
InteractionId	Unique identifier for a preference the user provided	INT
UserID	Foreign Key to the User Table	INT
GameID	Foreign key to the games table	VARCHAR
InteractionType	Whether the user liked or disliked the game	VARCHAR
InteractionTimeStampe	When the user interacted with the game	DateTime

Filter Category Table

Field	Notes	Type
category_id	unique id	INT
category_name	name of the category	VARCHAR

User Filters Table

Field	Notes	Type
User Filter ID	Unique ID for the user filter	INT
UserID	Foreign Key to the User Table	INT
CategoryId	foreign key to the filter category table	INT
option_value	What options the user has saved for their filters	VARCHAR

Section 5 - Software Domain Design

5.1 Software Application Domain Chart

Describe / chart each major software application domain and the relationships between objects (UML, etc)

5.2 Software Application Domain

5.2.1 Recommendation Engine Domain

- High-Level Description:
  - The Recommendation Engine Domain is responsible for generating personalized game recommendations based on user preferences. It involves generation/ translation to one hot encoded matrices and utilizing a cosine similarity algorithm to generate similarity of games in comparison with a users already interacted with games

5.2.1.1 Recommendation Engine Component

- Definition:
  - The Recommendation Engine Component is the core module within the Recommendation Engine Domain. It encompasses the logic and algorithms for processing user data and generating game recommendations.
- Data Flow/Control at Component Level:
  - User preferences, including selected genres, are input into the Recommendation Engine.
  - Game data from the Game Catalog Domain is also fed into the Recommendation Engine.
  - The Recommendation Engine processes this data, and compares 500 randomly pulled games through this generated profile to generate similarity score so we can pull the most similar games
  - Output includes a list of recommended games personalized for the user.

5.2.1.1.1 Recommendation Generation Task of Recommendation Engine Component

- Definition:
  - The Recommendation Generation Task is a specific activity within the Recommendation Engine Component. It focuses on the actual generation of personalized game recommendations.
- Data Flow/Control at Task Level:

- The task receives user preferences and game data from the Component level.
- Applies content recommendation algorithm to select and recommend similar games to already interacted with content
- Generates a list of recommended games based on the analysis.
- The list is then provided as output to be displayed to the user.

Section 6 – Data Design

6.1 Persistent/Static Data

6.1.1 Dataset

IGDB Game Data - Game database that stores tons of metadata on almost every game thats ever been released, available via API.

6.1.2 Static Data

IGDB Game Data
Error Messages
Default Recommendation Presets

6.1.3 Persisted data

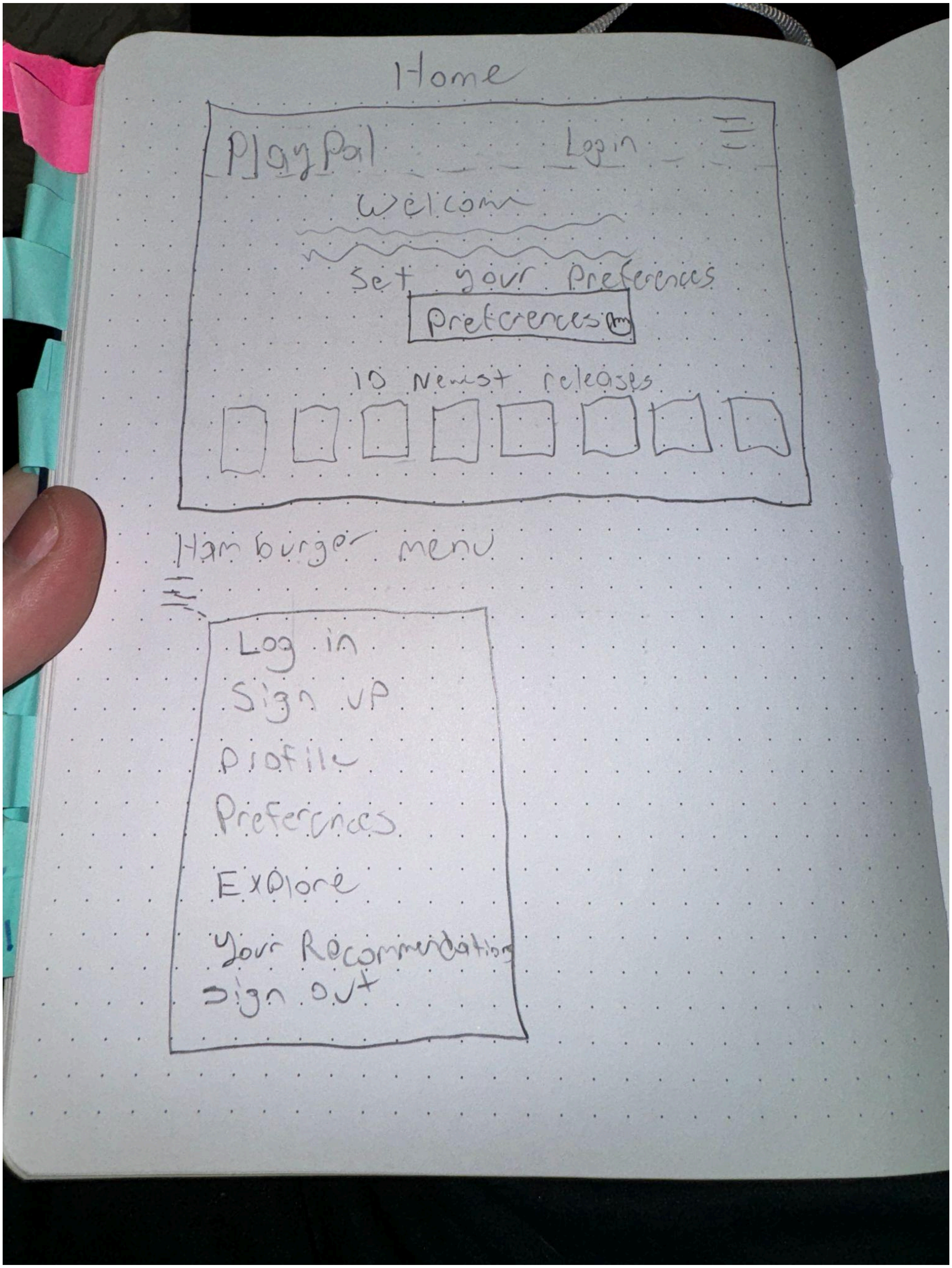
Recommended games the user hearts
User profile data
Machine learning profile specific input parameters
Game metadata

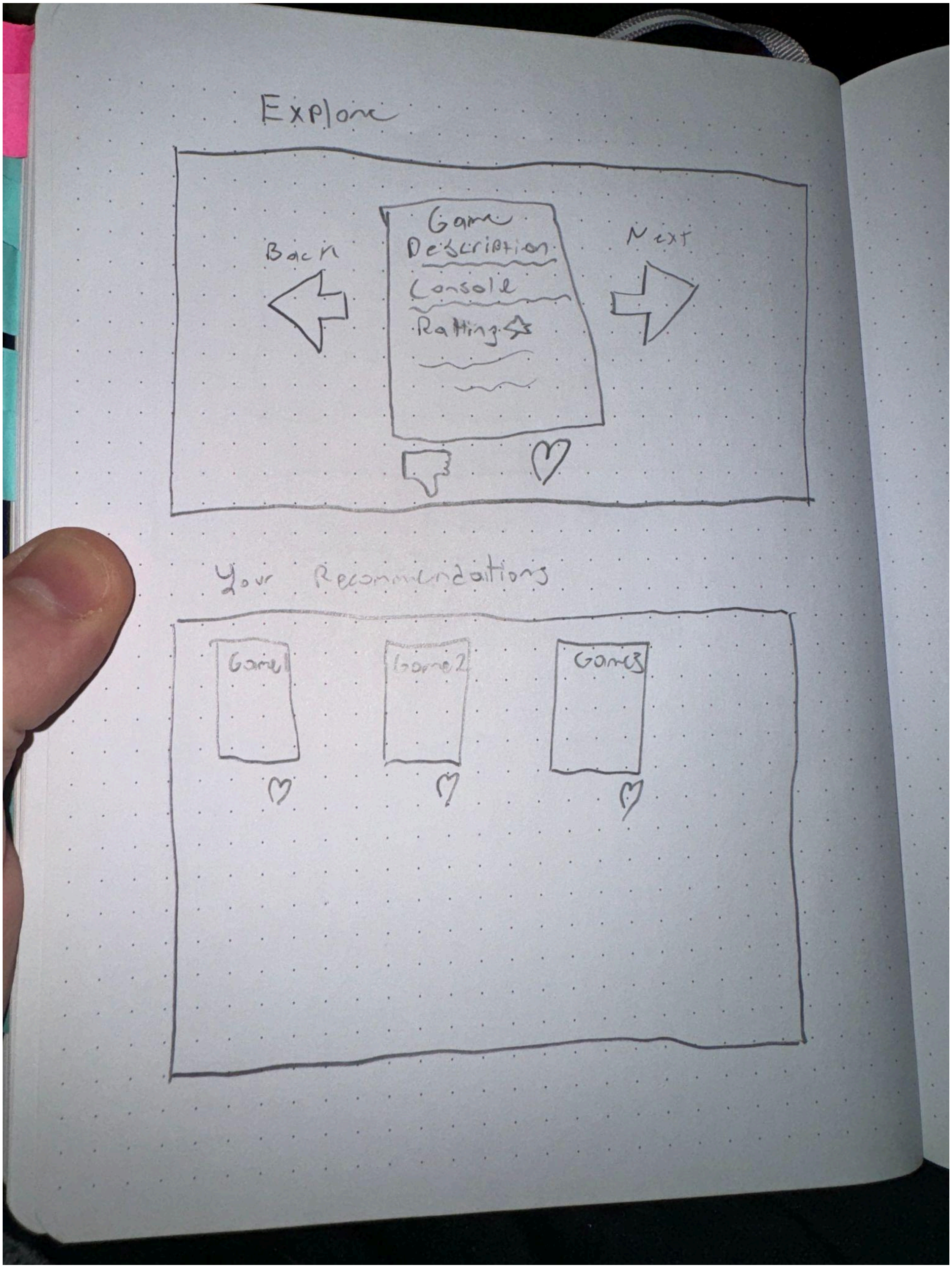
6.2 Transient/Dynamic Data

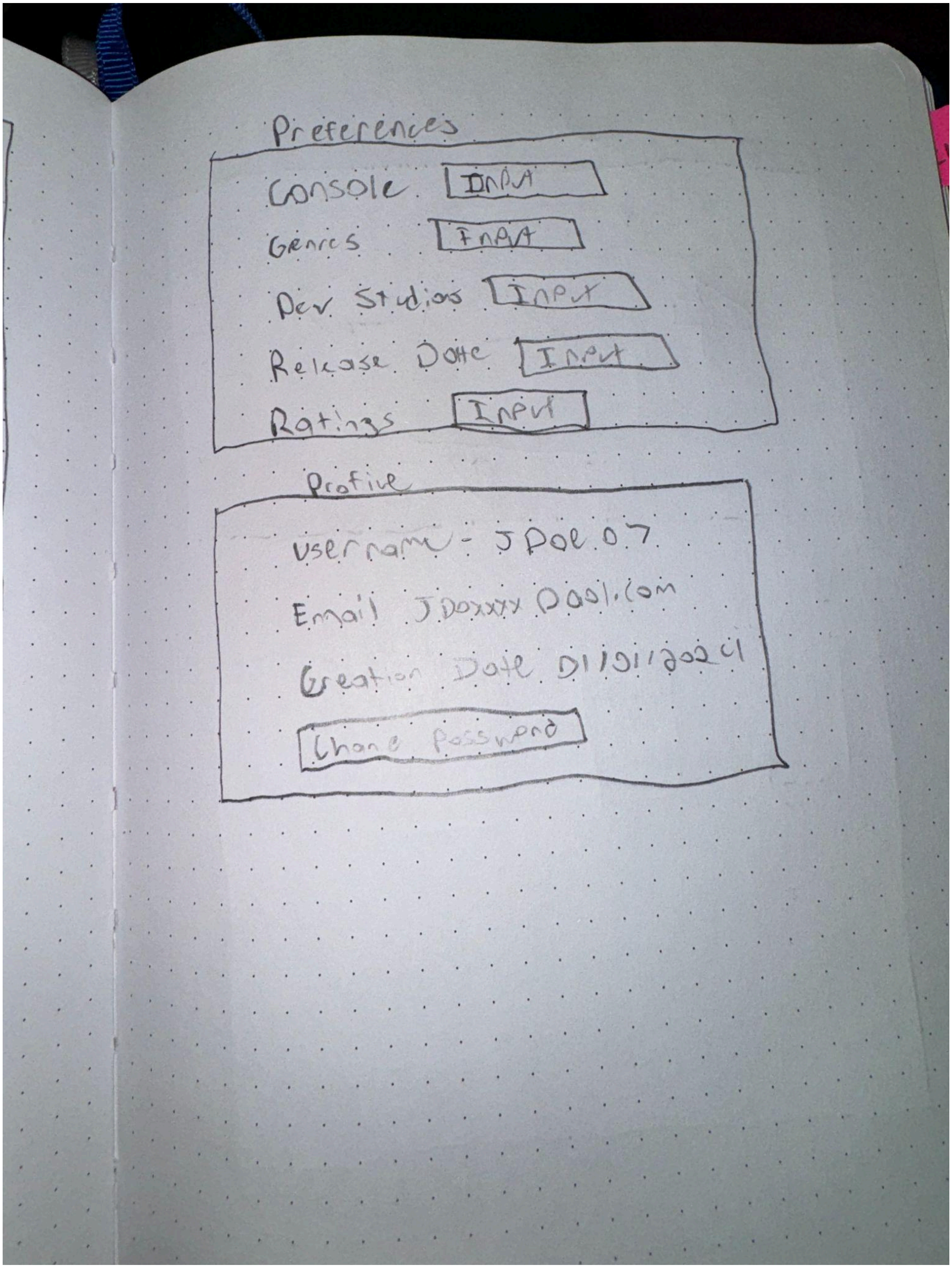
Database Queries - Setting up queries to filter on from IGDB API and our own database
User Session Data - Things like session ID, session log time
Recommendation Results - The engines generated recommendations are temporarily stored and all of them might not persist if the user does not heart them
Temp User Input Storage - When the user fills in their recommendation filters they will be temporarily stored for the session

Section 7 - User Interface Design

7.1 User Interface Design Overview


Home
PlayPal
Login
Welcom
Set your Preferences
Preferences
10 Newest releases
Hamburger menu
Log in
Sign up
profile
Preferences
Explore
Your Recommendations
Sign out


Explore
Back
Game
Description
Console
Rating
Next
Your Recommendaitions
Game1
Game2
Game3


Preferences
Console
Input
Genres
Input
Dev. Studios
Input
Release Date
Input
Ratings
Input
Profile
Username - JDoe07
Email JDoxxxx@aol.com
Creation Date 01/01/2024
Chane Password

7.3 Use Cases / User Function Description

1. User Registration and Login:

- Use Case:
  - New users can register for an account to receive personalized game recommendations. Existing users can log in to access their profiles.
- User Function:
  - New User:
    - Navigate to the "Sign Up" page.
    - Provide required information: username, email, password.
    - Click on the "Sign Up" button to register.
  - Existing User:
    - Navigate to the "Login" page.
    - Enter username and password.
    - Click on the "Login" button to access the account.

2. Inputting User Preferences:

- Use Case:
  - Users can input their preferences to receive accurate game recommendations.
- User Function:
  - Log in and navigate to the "Preferences" section.
  - Select preferred game genres, play frequency, and specific features.
  - Click on the "Save" button to update preferences.

3. Browsing Recommended Games:

- Use Case:
  - Users can explore game recommendations based on their preferences.
- User Function:
  - Log in and navigate to the "Recommended Games" section.
  - View a list of personalized game recommendations.
  - Click on a game to view details.

4. Providing Feedback on Recommendations:

- Use Case:
  - Users can provide feedback on recommended games.
- User Function:
  - Click on a recommended game to view details.
  - Rate the game and leave optional comments in the feedback section.
  - Click on the "Submit Feedback" button.

7. Viewing Game Details:

- Use Case:
  - Users can view detailed information about a specific game.
- User Function:
  - Click on a game from the recommended list or genre exploration.
  - View details such as title, genre, release date, and description.