

Game Listings

The system shall display multiple categories of games on the homepage:.

- The **Trending score** of a game is its number of buyers in the last 7 days divided by the highest number of buyers for any game in the last 7 days.

1. Recommended Games

- The system shall include games that share **tags** already prefer using the following weights:
 - **50% tag score** based on the user's own **favourite tags**.

Favourite tags= The top 3 tags the user has purchased games of ordered by how many games each tag has. The **tag score** is 0.333 times the number of **favorite tags** the game has.

- **50% based trending score**
- If the user has no preferred tags, their respective values will be 0. So the algorithm will show new users, with no preferred tags, games based solely on their **trending score**.

2. Trending Games

- The system shall list the games with the highest **trending score** for each category (Top 10 for each category)

3. Games on Sale

- The system shall display the most popular(highest **trending score**) games that are currently discounted.

Filtering and Search

The system shall provide the following filtering and search features:

Filtering

Users shall be able to refine displayed games based on:

- **Price** : The user will be able to set a lower and an upper bound for price to filter by. For now there will be a single, general currency type supported and taken into consideration. Games that have prices within the bounds will be shown
- **Review Score**: The user will be able to see games with a minimum review score for games. The review score is an "out of 10" score which is the average of all the scores of community reviews from other users who have already purchased the game

Search

- The system shall provide a **search bar** that allows users to find games by **name**. The system will try to search for games that have as a name a superset of what the user typed in the search bar. If there is no game that match this condition, the system will display no game, telling the user there are no results matching their search
-

Navigating to different pages from the Homepage

Navigating to a Game Page

- The system shall enable users to go to the store page of any game that is shown on the homepage

Wishlist

- The system shall provide a way to take the user to their User Wishlist.

Points

- The system shall provide a way to navigate the user to the Points Shop
- The system shall provide a way for the user to see their points

Wishlist:

A **wishlist** is a feature that allows users to save and track items they are interested in but are not ready to purchase yet. In the context of a **game store**, a wishlist helps users keep a **personalized list of games** they may want to buy in the future. ([games](#) that they have a interest in)

Wishlist Management

- Users shall be able remove games from their wishlist.
- The system shall display all wishlisted games, including their current price, discount (if applicable), and rating score.
- Users shall have the ability to sort wishlisted games by criteria such as:
 - Price (ascending/descending)
 - Discount percentage (how much a game is discounted)
 - User rating (how well the games is viewed in the eyes of the players - user rating is explained in the homepage requirement above)

Integration with Store & Recommendations

- The system shall allow users to navigate directly from their wishlist to the game's store page. (A button in the corner of the wishlist page that says "Back" and restores the user to the previous page)

Implementation Example:

- The wishlist will have an "Add to Wishlist" option on every game's store page.
- Once added, the game appears in the user's Wishlist section, displaying its price, discount status, and review score.
- The user can remove a game at any time by clicking a "Remove" button next to it.

Minimum Rating Threshold

The minimum rating threshold allows users to filter games based on their overall review score. This is useful for avoiding poorly reviewed games.

- Example rating categories:
 - **Overwhelmingly Positive** (9+ rating score)
 - **Very Positive** (8-9 rating score)
 - **Mixed** (4-8 rating score)
 - **Negative** (<4 rating score)

Implementation Example:

- Users set a minimum rating threshold (e.g., only show games with a "Very Positive" rating or higher).
- The wishlist updates dynamically, hiding games that do not meet the threshold.

- Example: If a user sets the minimum rating to “Overwhelmingly Positive,” a game rated “Mixed” will not appear in search results.

Search

- **Name Search:** Users shall be able to search for games by [name](#) in both the store and wishlist.

Name Search Works in a Wishlist System

Overview

The **Name Search** feature allows users to quickly find a specific game by typing its title (or part of it) in the search bar. This feature is essential for both the **store** and the **wishlist**, enabling users to locate games efficiently without scrolling through long lists.

How It Works (Step-by-Step)

1. User Input

- The user types a **keyword** or **full game title** in the search bar.
- Example: If a user searches for "CyberQuest", they will see all games that match "CyberQuest " in their wishlist or store.

2. Search Processing

- The system scans **game titles** in the user's wishlist or store database.
- It matches:
 - **Exact Matches** – If the user types "Elden Ring", the system finds "Elden Ring".
 - **Partial Matches** – If the user types "Elden", it still finds "Elden Ring".
 - **Case-Insensitive Search** – "elden ring" is the same as "Elden Ring".
 - **Fuzzy Matching (Optional)** – The system can suggest close matches in case of typos (e.g., searching "Zelda Breathe" could suggest "The Legend of Zelda: Breath of the Wild").

Game Page:

The system shall provide a dedicated page for each game, having the following features:

Game Description

- The system shall display a comprehensive description of the game as a UI text, where the developer can specify its genre, gameplay mechanics, storyline, and unique selling points.
- The system shall allow developers to format descriptions with headings, bullet points, bold text for better readability.

Game Specifications

- The system shall list both the **minimum** and **recommended** system requirements necessary to run the game.
- The system shall ensure that specifications include details such as **operating system, processor, RAM, graphics card, storage requirements**.
- Informative example : Minimum/Recommended requirements: Operating system Windows 10, Intel I5 7200-u processor, 8 GB RAM, NVIDIA RTX 3090 graphics card, 16 GB available storage.

User Reviews & Ratings

- The system shall display an aggregated **overall rating**(average value of all review scores by users) based on user reviews, shown as a star-based system(From 1 star minimum to 5 maximum, with half stars being available).
- The system shall ensure that reviews contain both **text feedback(which is imported from the community section)** and an optional **rating score(star-based system)**.

Game Media (Images & Videos)

- The system shall display a **characteristic capture** on the top of the page, near the game title. Each game has a single image that should represent the game.
- The system shall also have **hyperlinks displayed** under the description, which redirect the users to the official trailers and gameplay showdowns of the game.

Navigation & Additional Features

- The system shall provide a **clear purchase option**, allowing users to buy or add the game to their wishlist(two separate buttons at the top of the page, one representing the buy option which then adds the game in the user's cart, and a way that allows the user to add the game to the wishlist).
- The system shall include exactly 3 hyperlinks to related games, such as **expansions or similar titles**. Similarities are defined by genre (a game with the same genre as another is similar to it), with priority taken by the games which have the highest aggregated rating (Example: For Assassin's Creed Unity a first recommendation should be Dota2 because its rating is 4/5 , and after Dota there should be the next 2 similar games by genre ordered descendingly by rating : like Elden Ring which has a

rating is 3,5/5(because it is the second highest rated in the adventure genre) and then Sea Of Thieves with a rating of 3/5 (the third highest rating in the adventure genre).

Transaction Requirements

1. Cart Management

- Users can remove items from the cart before completing the transaction.
- The cart should persist across user sessions.

2. Payment Method Selection

- Users must be able to choose a payment method from the following options:
 - Steam Wallet
 - PayPal or Credit/Debit Card
- Users can change their selected payment method before confirming the transaction.

3. Payment Validation

- If the user selects Steam Wallet:
 - The system must verify that the current balance is greater or equal than the price of the game.
 - If the balance is insufficient, the transaction cannot proceed.
- If the user selects PayPal or Credit/Debit Card:
 - The system must validate the format of the payment credentials before proceeding.
 - For Credit/Debit Card: It checks if the CVV is exactly 3 digits, the card number is 13 digits, and the expiration date follows the correct format.
 - For PayPal: It verifies that the email is a valid Gmail address and that the password is at least 8 characters long.
 - The system does not check if the provided details exist or are correct, only that they meet the required format.
 - If validation fails, the user must be notified of the specific reason and prompted to correct the invalid details.

4. Transaction Processing

- Upon successful validation:
 - The system deducts the purchase amount from the selected payment method. The payment can be done just with the store's currency described at the homepage requirement above.

5. Post-Transaction Actions

- If the payment is successful:
 - The purchased items are marked as owned by the user.
 - A confirmation message is sent via an in app notification.
- If the payment fails:
 - The user is notified of the failure reason and asked to change the details that led to the specific failure.
 - The transaction is canceled, and no funds are deducted from the actual steam wallet.

Developer Page:

This section provides an overview of how developers manage their games on the platform. It covers creating, updating, and deleting games created by the developer, as well as the process by which games become visible in the store once they are approved. Every developer will have its own personal page, where they can manage the following:

Game Creation

- Developers have access to a “Create New Game” feature, allowing them to submit basic game information (such as title, price, genre, requirements and initial game release discount), along with media (one screenshot) in order to add a new game to their page, pending approval (will be specified later in [Validation and Publication](#))
- Any newly submitted game begins in an unvalidated state and requires approval from other developers (will be specified later in [Validation and Publication](#)) before it is visible to the public and on the store page.
- Each game’s status is displayed (e.g., **Pending, Approved, Rejected**) along with the game name on the main page, so developers always know whether a game is live or awaiting review.

Game Updating

- Developers have access to a “Update Game” feature, allowing them to update information about a game created by them inputting new information, such as title, description, price, genre, tags, the current discount that will apply to the listing on the store or the media (screenshots and trailers) that comes along with it.
- Once updated (i.e. the developers enter the new data about each game), the game returns to a pending state until the changes are reviewed and approved by other developers (will be specified later in [Validation and Publication](#)) .
- If an update is not accepted by other developers, the developer sees a brief explanation (e.g., “Rejected with reason”), to help them address any issues, and then try again.

Game Deletion

- A developer may only delete games that they created, and not interfere with any of the games from the other developers.
- When a developer confirms deletion, after being asked if he’s sure, the game is removed from public listings on the store, such that it is no longer visible to any user, and no longer appears in the developer’s portfolio.
- If a game already has active players that have purchased the game, the system may display a message advising the developer that the game has been previously bought by others, and the developer will be asked for reconfirmation, as the game will disappear completely from every game owner’s library

Validation & Publication

- Newly created or updated games require review by one other developer, after that the game will disappear from the validation page and no other reviewer will be able to accept or reject it. Once approved, they become visible in the store for each user
- The developers who have a game rejected will receive a message from the developer that has rejected their game, where the reason (e.g. The name is inappropriate) for the refusal will be clearly stated, and the game will be put in a “Rejected” state, such that the developer

that has programmed the game, will know what changes he needs to address in order for the game to be accepted on the market.

- Developers can review and validate **other** developers' games, providing feedback or approval, as stated above. However, they **cannot** self-approve their own games; those must be validated by a different reviewer.

Points Shop:

General requirements:

- The Points Shop system must be integrated into the store section of the platform as a sub-tab of the homepage.
- Users must have a separate balance for points (points are a distinct type of in-game currency and user can buy specific items only with them) , distinct from their monetary currency.
- The system must display the user's current points balance on the store page and in the Points Shop tab at all times.

Earning Points

- Users earn 10% of the total purchase price of any store transaction as points and points must be added to the user's balance immediately after a successful monetary purchase.
- Points should not be awarded for purchases also made with points, they should only be awarded for purchases made with real money currency.
- Users should receive a confirmation notification when the points has been added to the balance.

Spending Points

- All items in the Points Shop must be exclusive to points purchases and cannot be bought with real money.
- There is no cart, you can directly buy an item from the Point Shop page and it will be displayed at the bottom of the page along with all other already bought items.
- When an item is purchased with points, the corresponding amount of points must be deducted from the user's balance and the item should be deleted from the buy list.
- If the user does not have enough points, the purchase should be prevented with an appropriate message.
- Users should receive a confirmation notification when an item is purchased with points.

Points Shop UI & Storefront

- The Points Shop must clearly indicate the point cost of each item, its photo and the name.
- Users must be able to see item details(such as name, description, how much it costs) before purchasing it, these details will be shown below the game's picture.
- The User should be able to see its points currency at all times in the UI.

Transactions & Verification

- Users should be able to view their points transaction history in a sub-tab (earnings and spending).
- The system must prevent fraudulent point gains (e.g., canceling purchases after earning points) by checking to see if the transaction was made successfully.