

# Functional Requirements Document

## Introduction

This document serves as the Functional Requirements Document (FRD) for the **GAMEDROP** web application. Its purpose is to provide a detailed description of the system's functionalities from a user perspective, ensuring clear communication between the development team and other stakeholders. This FRD details all the core functionalities expected in the **GAMEDROP** application based on the business requirements found in the Project Rubric file, providing a comprehensive reference for development. The functionalities are described from the viewpoint of the primary user roles: Members (Borrowers) and Administrators.

## Identified Functionalities

The following core functionalities will be detailed in this document:

- ❖ User Registration and Authentication
- ❖ Browse Game Catalog
- ❖ View Game Details
- ❖ Search and Filter Games
- ❖ Shopping Cart Management
- ❖ Checkout Process
- ❖ View Order History
- ❖ Product Management (Admin)
- ❖ Order Management (Admin)
- ❖ User Account Management (Admin)
- ❖ Product Review/Rating

## ❖ Inventory/Stock Management (Admin/System)

# Detailed Functional Requirements

## 1. User Registration and Authentication

**Actor(s):** New Visitor, Registered User (Customer or Administrator)

**User Goal/Objective:** To create a customer account or log in securely to access shopping features and order history.

### Detailed Steps/Scenarios:

#### a) Registration (New Visitor):

- Visitor navigates to the "Register" page.
- Visitor enters required information (e.g., username, email, password, password confirmation).
- Visitor submits the registration form.
- System validates the input (e.g., unique username/email, password complexity, matching passwords).
- If validation fails, system displays specific error messages.
- If validation succeeds, system creates a new User account (default role: Member), securely hashes and salts the password, and stores the user record.
- System logs the new user in automatically (generates JWT).
- System displays a success message and redirects the user to their dashboard or the homepage.

#### b) Login (Registered User):

- User navigates to the "Login" page.
- User enters their registered email and password.
- User submits the login form.

- System verifies the credentials against stored (hashed) passwords.
- If verification fails, system displays an error message ("Invalid credentials").
- If verification succeeds, system generates a JWT session token.
- System redirects the user to their appropriate dashboard (Member or Admin).

**c) Logout (Logged-in User):**

- User clicks the "Logout" button/link.
- System invalidates the user's session/token.
- System redirects the user to the homepage or login page.

**Involved Content/Information:** User Account Record (Username, Email, Hashed Password, Role), JWT Token.

**Constraints/Conditions:** Username and Email must be unique. Password must meet complexity requirements. User must provide valid credentials to log in.

## 2. Browse Game Catalog

**Actor(s):** Customer, Administrator, New Visitor

**User Goal/Objective:** To view the selection of video games available for purchase.

**Detailed Steps/Scenarios:**

- User navigates to the "Game Catalog" or "Browse Games" section of the application.
- System retrieves a list of all canonical Game Items from the database.
- System displays the list of games, typically showing key information like Game Title, Cover Art, Price, and potentially Platform/Genre per item.
- System may include pagination if the number of games is large.

**Involved Content/Information:** Game Item Records (Title, Price, Cover Art, Platform, Genre).

**Constraints/Conditions:** Only products marked as 'active' or 'visible' by Admin are shown.

### 3. View Game Details

**Actor(s):** Customer, Administrator, New Visitor

**User Goal/Objective:** To see detailed information about a specific video game before deciding to purchase.

**Detailed Steps/Scenarios:**

- User clicks on a specific game title or cover art from the Game Catalog page (or search results).
- System retrieves detailed information for the selected canonical Game Item (Title, Description, Platform, Genre, Price, Cover Art/Images, Stock Status (e.g., "In Stock", "Out of Stock"), Average User Rating).
- System displays an "Add to Cart" button if the product is in stock.
- System displays user reviews/ratings for the product (See Functionality #11).

**Involved Content/Information:** Specific Game Item Record (all details), Stock Status, Average Rating, User Reviews.

**Constraints/Conditions:** "Add to Cart" button visibility depends on Stock Status.

### 4. Search and Filter Games

**Actor(s):** Customer, Administrator, New Visitor

**User Goal/Objective:** To find specific games within the catalog more efficiently based on search criteria or filters.

**Detailed Steps/Scenarios:**

- User accesses the Game Catalog page.
- System displays search input fields and filter options (e.g., text search box, dropdowns/checkboxes for Platform, Genre, Price Range).
- **Search:** User enters keywords into the search box and initiates the search (e.g., clicks "Search" or types). System queries the database for Game Items matching the keywords in title or description. System updates the displayed list to show only matching games.
- **Filter:** User selects filter options (e.g., Platform="PS5", Genre="RPG"). System queries the database for Game Items matching the selected filters. System updates the displayed list to show only matching games.

**Involved Content/Information:** Game Item Records (Title, Description, Platform, Genre, Price), Search Keywords, Filter Criteria.

**Constraints/Conditions:** None.

## 5. Shopping Cart Management

**User(s) / Actor(s):** Customer (Logged-in)

**User Goal/Objective:** To add desired games to a temporary list (cart) before purchase, review the list, and make quantity adjustments.

**Detailed Steps:**

- **Add to Cart:** On a Product Detail page (See Functionality #3) or Catalog listing, Customer clicks "Add to Cart" for an in-stock item. System adds one unit of the selected Product to the Customer's current shopping cart session/record. System provides visual feedback (e.g., cart icon updates, success message).

- **View Cart:** Customer navigates to the "Shopping Cart" page. System displays all items currently in the cart, showing Product Title, Price, Quantity, and Subtotal per item, plus the overall Cart Total.
- **Update Quantity:** In the Cart view, Customer changes the quantity for an item. System updates the quantity, item subtotal, and overall Cart Total. System may check against available stock (See Functionality #12).
- **Remove Item:** In the Cart view, Customer clicks a "Remove" button next to an item. System removes the item from the cart and updates the Cart Total.
- **Proceed to Checkout:** Customer clicks the "Checkout" button. (See Functionality #6).

**Involved Content/Information:** Shopping Cart Record (associated with User session/ID), Cart Item Records (Product ID, Quantity), Product Records (Price, Stock Status), Cart Total (Calculated).

**Constraints/Conditions:** User must be logged in to have a persistent cart. Cannot add more items than available stock.

## 6. Checkout Process

**User(s) / Actor(s):** Customer (Logged-in)

**User Goal/Objective:** To finalize the purchase by providing necessary information and confirming the order.

### **Detailed Steps:**

- Customer proceeds from the Shopping Cart page.
- System displays the checkout page, potentially broken into steps (Shipping Info, Payment Info, Review Order).
- **Shipping Info:** Customer enters or confirms their shipping address details. System saves/updates this information associated with the user profile or the specific order.

- **Payment Info (Simulated):** System displays placeholder payment options (e.g., "Credit Card," "Cash on Delivery"). Customer selects an option. **No actual payment processing occurs.**
- **Review Order:** System displays a final summary: items ordered, quantities, prices, calculated total (including simulated tax/shipping), shipping address.
- Customer clicks "Place Order" / "Confirm Purchase".
- **Order Placement:** System performs the following atomically:
  - Validates order details (e.g., stock availability one last time).
    - a. Creates a new Order record in the database with status 'Pending' or 'Processing', associating it with the customer, items, quantities, total price, and shipping address.
    - b. Updates the stock count for the purchased items (See Functionality #12).
    - c. Clears the user's shopping cart.
    - d. Triggers a notification (e.g., in-app/email) to the customer confirming the order (See Functionality #7).
    - e. Redirects the user to an "Order Confirmation" / "Thank You" page displaying the order number and details.

**Involved Content/Information:** Order Record (User ID, Order Items, Total Price, Shipping Address, Status='Pending'/'Processing'), Product Records (Stock Count update), Shopping Cart (cleared), User Profile (Address). Calculation Service (for totals).

**Constraints/Conditions:** User must be logged in. Cart must not be empty. All items must be in stock at the time of confirmation.

## 7. View Order History

**User(s) / Actor(s):** Customer (Logged-in)

**User Goal/Objective:** To view past orders and their current status.

**Detailed Steps:**

- Customer navigates to their "Account" / "Order History" page.
- System retrieves and displays a list of all past Orders placed by the logged-in customer.
- For each order, the list shows key info: Order Number/ID, Date Placed, Total Amount, Current Status (e.g., 'Processing', 'Shipped', 'Delivered', 'Cancelled').
- Customer can click on an order to view full details (items purchased, quantities, prices, shipping address).
- System may display notifications related to status updates here.

**Involved Content/Information:** Order Records (filtered by logged-in User ID).

**Constraints/Conditions:** User must be logged in.

## 8. Product Management (Admin)

**User(s) / Actor(s):** Administrator

**User Goal/Objective:** To manage the catalog of games available for sale.

**Detailed Steps:**

- Admin navigates to the "Product Management" section.
  - a. System displays a list of all products (games). Admin can search/filter this list.
  - b. **Add Product:** Admin clicks "Add New Product". System displays form for Product details (Title, Description, Platform, Genre, Price, Stock Quantity, Cover Art Upload/URL, etc.). Admin submits. System validates and creates new Product record.
  - c. **Edit Product:** Admin selects a product. System displays details in an editable form. Admin modifies and saves. System validates and updates Product record.



- d. **Delete Product:** Admin selects a product and confirms deletion (consider soft delete). System removes/flags product.
- e. Admin can manage categories (Platform, Genre) used for filtering/organization.

**Involved Content/Information:** Product Records, Category Records.

**Constraints/Conditions:** User must be logged in as Admin.

## 9. Order Management (Admin)

**User(s) / Actor(s):** Administrator

**User Goal/Objective:** To view incoming orders and update their status as they are processed and fulfilled.

### **Detailed Steps:**

- Admin navigates to the "Order Management" section.
- System displays a list of all customer orders, filterable by status (e.g., show only 'Processing' orders).
- Admin selects an order to view its full details (customer info, items ordered, shipping address, current status).
- Admin can update the order status using predefined options (e.g., 'Processing' -> 'Shipped', 'Shipped' -> 'Delivered', or 'Cancelled').
- When status is updated (e.g., to 'Shipped'), the system may trigger a notification to the customer.
- System logs status changes and timestamps.

**Involved Content/Information:** Order Records (Status, all details).

**Constraints/Conditions:** User must be logged in as Admin. Status transitions may follow predefined rules.

## 10. User Account Management (Admin)

**User(s) / Actor(s):** Administrator

**User Goal/Objective:** To manage customer and potentially other admin accounts.

**Detailed Steps:**

- Admin navigates to "User Management".
- System displays list of users, searchable/filterable.
- Admin selects a user to view details.
- Admin can perform actions like: Edit Role (Customer/Admin), Disable/Enable Account, Trigger Password Reset.

**Involved Content/Information:** User Account Records.

**Constraints/Conditions:** User must be logged in as Admin. Cannot delete self or last admin.

## 11. Product Review/Rating

**User(s) / Actor(s):** Customer (Logged-in)

**User Goal/Objective:** To provide feedback on purchased games to help other customers.

**Detailed Steps:**

- Customer navigates to the Product Detail page (See Functionality #3) of a game they have previously purchased (or navigates via their Order History).
- System displays a section for writing a review and/or providing a star rating (e.g., 1-5).
- Customer enters their review/rating and submits.
- System validates input (e.g., requires rating, checks review length).

- System saves the Review record, associating it with the Product and the Customer.
- System may update the average rating displayed for the Product.
- Other users viewing the Product Detail page can see submitted reviews and the average rating.

**Involved Content/Information:** Review Records (User ID, Product ID, Rating, Comment Text, Date), Product Record (Average Rating).

**Constraints/Conditions:** User must be logged in.

## 12. Inventory/Stock Management (Admin)

**User(s) / Actor(s):** Administrator

**User Goal/Objective:** To track the quantity of physical game copies available for sale.

**Detailed Steps:**

- **Admin Input:** When adding/editing a product (See Functionality #8), the Admin sets the initial **STOCK\_QUANTITY**. Admin can manually update this quantity later if needed (e.g., receiving new stock).
- **System Update (Checkout):** When an order is successfully placed (Functionality #6), the system automatically decreases the **STOCK\_QUANTITY** for each purchased item by the quantity ordered.
- **System Update (Cancellation/Return):** If an order is cancelled *before shipping* or an item is returned, the system increases the **STOCK\_QUANTITY** accordingly.
- **Availability Display:** The system uses the **STOCK\_QUANTITY** to determine the "In Stock" / "Out of Stock" status displayed to users (See Functionality #3) and to control whether the "Add to Cart" button is active.

**Involved Content/Information:** Product Record (**STOCK\_QUANTITY** field).

**Constraints/Conditions:** Stock quantity cannot go below zero.