Software Requirements Specification

for

Video Game Shop (V.G.S)

Version 1.0 approved

Prepared by Group 28

Waqai Asli Chaddi Buddys

10th October, 2020

Contents

Chapter 1: Introduction

1.1 Purpose

This document describes the software requirements of V.G.S (Video Game shop). It is intended to aid the developer, the designer and the maintainer of the software.

1.2 Scope

The function of V.G.S is to provide a wide array of games to be bought or rented by a cash on delivery method. The users can register an account for themselves, browse through the games and select one which they would like to buy/rent and also have be notified when they have a game due to be returned. This project will streamline the process of buying items and have the added convenience of having delivered straight at your door.

1.3 Definitions

- **Username:**

A unique combination of letters, numbers and symbols that will be used to represent and identify the user in the database and the shop.

- **Password**

Password is a combination of letters numbers and symbols associated with a unique username only known to the user. It is used to protect and allow access to the users account

- **Items**

Items is used to refer the products available for sale at our shop this includes the Games and the Consoles.

- **Shopping cart**

A shopping cart is a feature that acts as a catalog and ordering process. Typically, a shopping cart is the interface between our website and its deeper infrastructure, allowing consumers to select merchandise; review what they have selected; make necessary modifications or additions; and purchase the merchandise.

- **Category**

A category is a class or division of items regarded as having particular shared characteristics.

- **COD**

COD is an acronym for Cash on delivery, the user will pay for the product on his doorstep as it is being delivered to him.

- **Stock**

Stock refers to the number of products available for sale in the warehouse

- **Retailer**

Retailer is the person contacted to replenish the stock.

- **Penalty:**

It is a fine that would be included in the users next purchase if he fails to return the game within the due date or returns faulty product

Chapter 2: General Description

2.1 Product description:

Video Game Shop is an online video game website which provides its users with all sorts of video games and accessories. Users can purchase or rent a video game based on their needs. It allows to pay cash on delivery or online payment (whatever they like) and delivers the product to their home. So overall it's an easier way to buy branded gaming i.e. a one stop shop for all gaming related accessories.

2.2 Product Functionalities:

The store provides multiple functionalities, providing all sorts of gaming related products. It's an online website on which user can type his/her user details to login. The website keeps the user details, history and shopping cart in its database. The admin can manage the site to manage or update the incoming stock, change the interface or view ongoing trends of purchase. Admin also sends mail notifications to the user accordingly. From user's perspective, he/she can search for a particular game or accessory and add that product to his/her cart and finally purchase or rent that product. Users can also remove a product from his/her cart if he does not want it. Finally user can pay through credit/debit card or cash on delivery.

2.3 User characteristics:

Admin:

Admin is responsible for managing the website. He has direct access to database and determines things when the stock will be updated and when email notification will be sent to a particular user. It sorts the games and deletes the games which have low downloads. Its responsibility is to verify a user login details and its payment methods etc.

User:

A user is person looking for some gaming product. User logins to the website by adding its login details. User searches for its desired items and purchases or rents it.

Chapter 3: System Requirements

3.1 Functional Requirements

The functional requirements for the Video Game Shop website are divided into three sections i.e. User functional requirements, Admin functional requirements and system functional requirements.

3.1.1 User Functional Requirements

User Functional Requirement 1

- *Description*

The website will allow new users to sign up into the system and would then allow the users to purchase various items/games from the website. Upon signing up, the website email client will send email to the user telling him/her that his/her account has been successfully created.

- *Input*
  1. *User email address*
  2. *Unique username*
  3. *User password*
  4. *Permanent address*

- *Processing*

User details added to the customer table in the database

- *Output*

An email sent to the user telling him/her that account has been successfully created

User Functional Requirement 2

- *Description*

The website will allow existing users to sign in in order to purchase items, games etc. using their unique username and passwords. The user keys in his/her details and then he presses the log in button to sign in.

- *Input*
  1. *Unique Username*
  2. *Password*

- *Processing*

The keyed in username is searched for in the user table. If no match is found, message displayed "user does not exist". If match is found, the corresponding password is matched with the one just typed in. If they are the same, authenticated otherwise rejected with message "incorrect password".

- *Output*

User successfully signs in or in case has provided incorrect details, messages displayed e.g. incorrect password

User Functional Requirement 3

- *Description*

User will search for the item which he/she wishes to buy in order to see whether it is available or not.

- *Input*
  1. *Category*
  2. *Name of item*
  3. *Criteria*

- *Processing*

The keyed in name is searched for in the appropriate table (which is identified by category)and if it matches the criteria entered to see if a match is returned

- *Output*

If match is found, details about the item are displayed onto the user screen

User Functional Requirement 4

- *Description*

In order to purchase items from the website, the user selects the item which he/she wishes to buy and adds it to the shopping cart.

- *Input*
  1. *The item(s) which the user wishes to buy*
  2. *Number of that particular item user wishes to buy e.g. 2 games*

- *Processing*

Once an item is selected, the number is deducted from the stock field in the database

- *Output*

The item(s) have been added to the shopping cart and the sub total is displayed along with the details of item added. User can now checkout. Email sent to user containing receipt.

User Functional Requirement 5

- *Description*

The user will be able to rent games, if he/she does not wish to purchase the game at the full price. User can rent one game at a time

- *Input*
  1. *User selecting the game he/she wishes to rent*
  2. *User enters the duration for which he/she wishes to keep the game*

- *Processing*

The user's name along with the name of game are stored into the rent table. The date on which the game is rented along with the date on which it is due will also be stored to keep track of user.

- *Output*

Game details along with the total amount displayed. Email sent containing receipt and date of receiving

User Functional Requirement 6

- *Description*

This feature allows the user to view his/her transaction history till date and displays it on to the screen

- *Input*

User presses "Order History" button

- *Processing*

In the sales table, all records are selected where the username is equal to the name of the user through SQL query

- *Output*

User transaction till date displayed with name of video games purchased, order dates etc

User Functional Requirement 7

- *Description*

This feature allows the user to group the items with respect to genre of a game or the media on which the games are played upon.

- *Input*
  1. *User presses the group by button*
  2. *The user then selects the category on which he/she wishes to group the items by*

- *Processing*

The tuples in question are grouped by the category selected by the user.

- *Output*

The result is displayed i.e. items grouped by category

User Functional Requirement 8

- *Description*

This feature allows the user to change his/her account details..

- *Input*
  1. *User selects the account management option and he/she is required to type in the password again*
  2. *User selects the details which he/she wishes to change*
  3. *User then keys in new details in the form*

- *Processing*

The password entered is compared with the one stored in the database. If they are the same then user is allowed to change details otherwise denied. After filling out the form, the relevant fields are updated in the user table.

- *Output*

User has successfully changed her/his details

User Functional Requirement 9

- *Description*

This feature allows the user to gain in store credit upon transaction which can be later used to purchase games without any payment, store credits utilized

- *Input*
  1. *After filling shopping cart, the user checks out (presses checkout button)*

- *Processing*

Once user checks out, 20% of the total are added as in store credit for the particular user

- *Output*

User gains store credit and credit updated

3.1.2 Admin Functional Requirements

Admin Functional Requirement 1

- *Description*

The admin will be able to add or update new items as stock arrives.

- *Input*
  1. *Admin logs on to the admin page*
  2. *Admin fills out form related to the item*

- *Processing*

Appropriate table(s) will be updated once the form has been filled depending upon the category of the item

- *Output*

The admin has successfully added new items in the database

Admin Functional Requirement 2

- *Description*

This feature allows the admin to display the most bought game in a particular period of time to attract users into buying this particular game

- *Input*
  1. *The admin needs to be logged into his/her account*
  2. *The admin selects the duration in which he/she needs to find out the most purchased game*
  3. *He/she presses the find button*

- *Processing*

A query is executed to find the best selling items in a particular period of time as specified by the admin. The items are grouped by name and total revenue calculated for that period. The item with the greatest revenue returned

- *Output*

The item with the greatest revenue is displayed along with its detail. Now the admin can display this on the homepage.

Admin Functional Requirement 3

- *Description*

The admin will check whether new stock needs to be ordered or not.

- *Input*
  1. *The admin presses the check stock button*

- *Processing*

In the database, all items will be checked to determine which item is low in stock. If any of the item(s) are less than the threshold set by the admin (say 5), he/she decides to order the stock.

- *Output*

An email is sent to the retailer, demanding new stock and message is displayed on the screen telling whether email has been sent or not.

Admin Functional Requirement 4

- *Description*

The admin will check whether any game needs to be returned or if any charges need to be paid.

- *Input*
  1. *The admin presses the Check dues button*

- *Processing*

In the rent table, the due date of every game rented by every user is compared with the current date and if it is before the current date, then that user needs to return the game.

- *Output*

The details of every user who rented the game(s) in question are displayed on the screen. An email is then sent to each user reminding them to return the games that they rented or there will be a penalty.

Admin Functional Requirement 5

- *Description*

The admin can remove any old or outdated item or whether it is discontinued.

- *Input*
  2. *The admin enters the name of the item to be removed*

3. *Admin selects the category of said item*

- *Processing*

The item is searched in the category specified by the user. Once a match is found, the admin is asked whether he/she really wants to delete the item from said table.

- *Output*

The item has been successfully removed from the database.

Admin Functional Requirement 6

- *Description*

The admin can calculate monthly profit for every item sold

- *Input*
  1. *The admin enters the month for which profit needs to be calculated*

- *Processing*

All the items sold that particular month are selected, their total cost price is subtracted from the total sale price.

- *Output*

Profit is calculated, displayed on the screen as well as stored in the profits table

Admin Functional Requirement 7

- *Description*

The admin can view transactions for every day, week, month, year. He/she can calculate the total revenue for that particular period of time

- *Input*
  2. *The admin enters the date for which transaction history needs to be displayed*
- *Processing*

All the items sold that particular date are selected, grouped and displayed. If admin selects the sum option, then total sale for that date is calculated

- *Output*

Transaction details are displayed for that date as well as total

Admin Functional Requirement 8

- *Description*

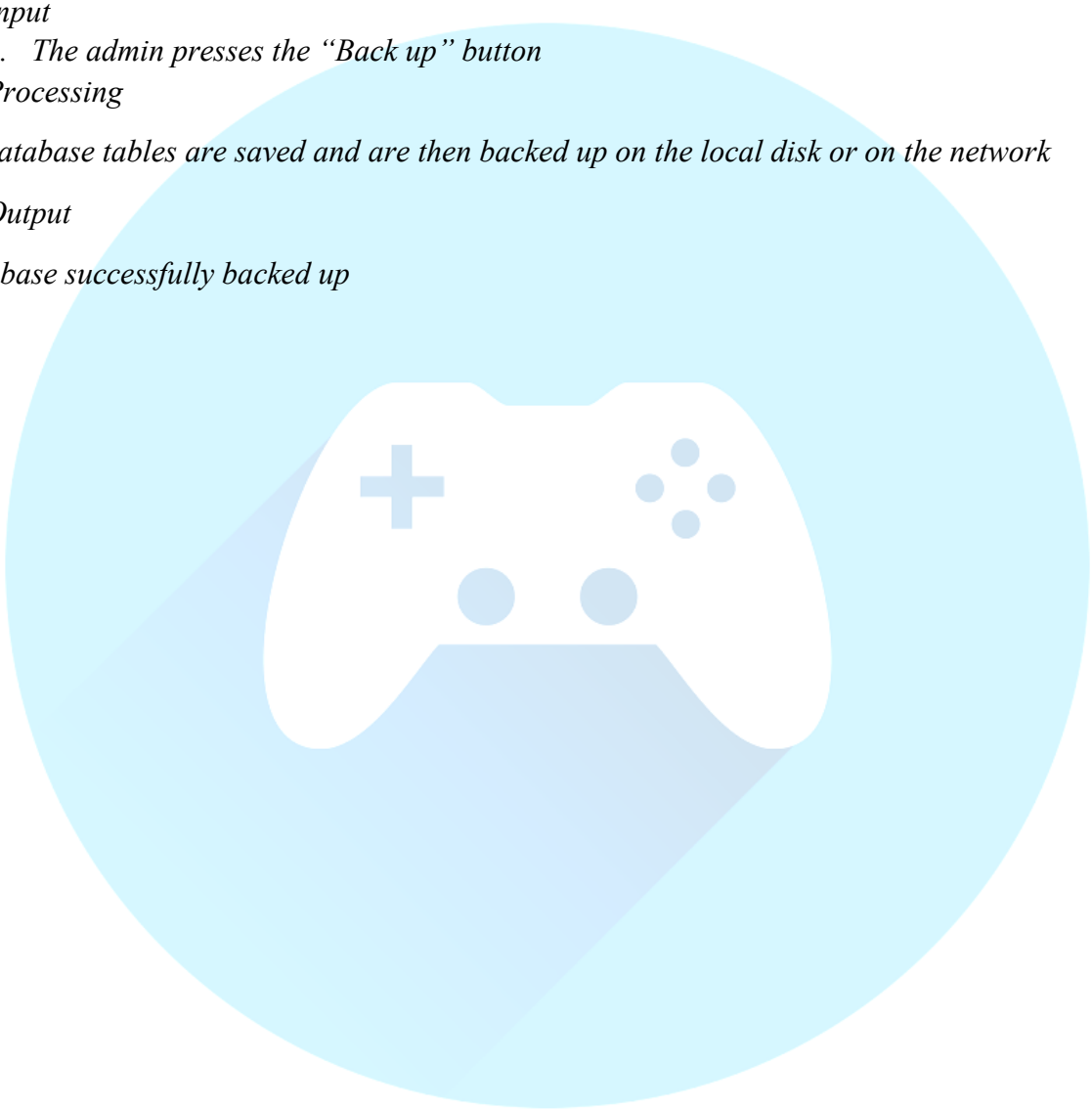
The admin can back up the database regularly so that in case of any faults, back up is present to jump back where he/she left off

- *Input*
  3. *The admin presses the "Back up" button*
- *Processing*

All database tables are saved and are then backed up on the local disk or on the network

- *Output*

Database successfully backed up



3.2 External Interface Requirements

3.2.1 User Interface Requirement

The system should provide an easy to use interface with a web portal that could allow features such as browsing through the available games, their prices and access to a form order them. And a simple system for the Admin to keep an eye out for stock, profits, etc. A simple interface at the POS terminal that uses a keyboard and a mouse integrated with the local area network.

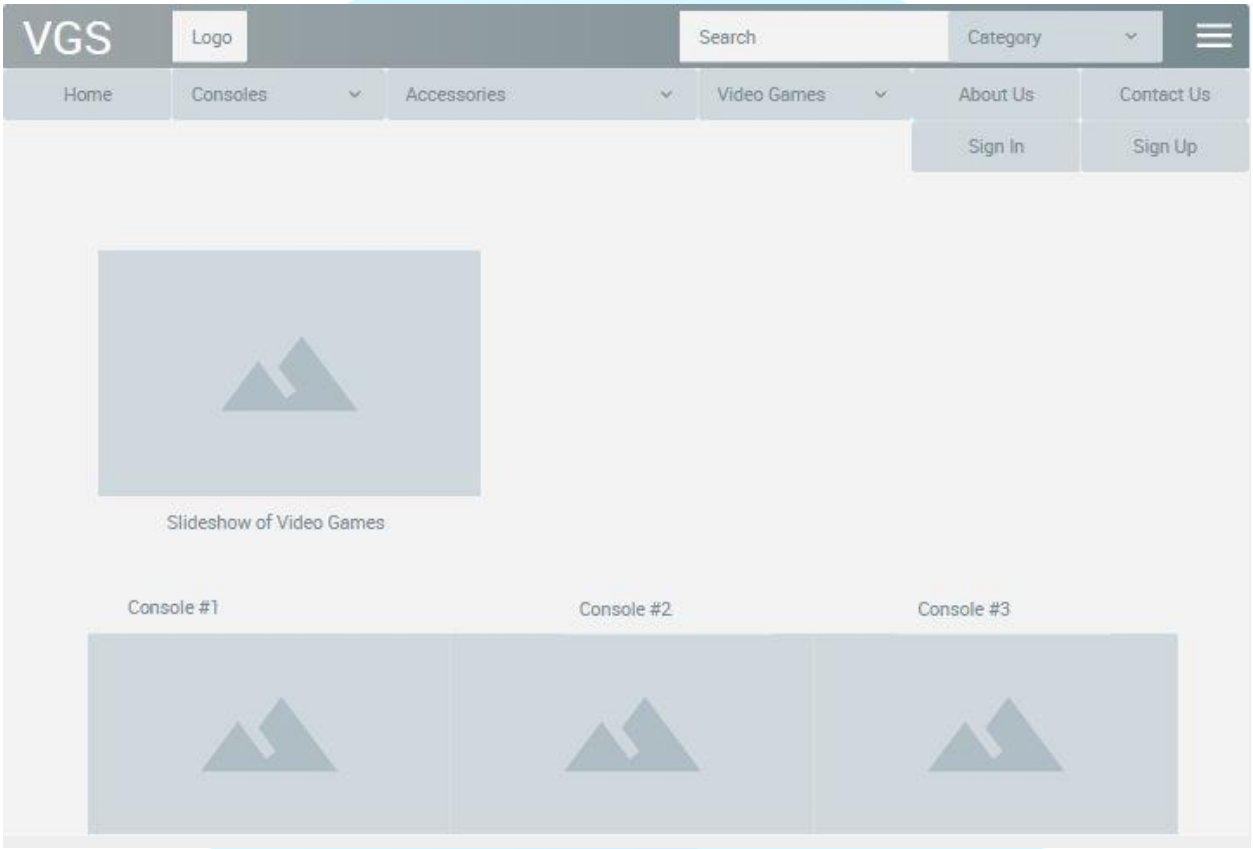


Figure (a). Initial layout of homepage

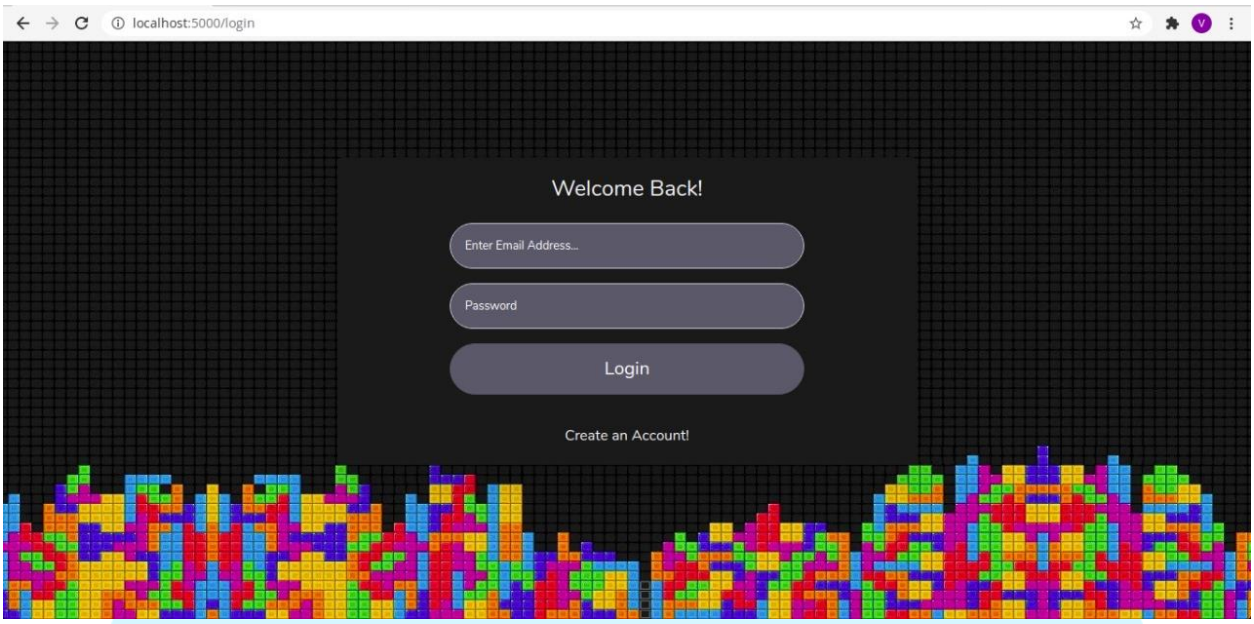


Figure (b). Layout of user login page

3.2.1 Software Interface Requirement

The website should be accessible through any of the latest web browser e.g. Google Chrome, Mozilla Firefox. Other than that, it should work on any Operating System e.g. Linux, Windows etc.

3.2.3 Hardware Interface Requirement

Since the application must run over the internet, all the hardware shall require to connect to the Internet will be hardware interface for the system i.e. Modem, WAN – LAN, Ethernet. Moreover, keyboard and mouse required to browse the website and a monitor to view the UI.

3.3 Security Requirements

- *Every user and admin will have a unique username*
- *Every password would be greater than 8 characters and should at least one special case character*
- *Only people listed in the admin table will be able to log in to the admin page, no unauthorized*
- *Users will be required to be logged in to their accounts before ordering/purchasing any item from the website*

- *Admin page will only be available to the logged in admin, public users cannot view the page*
- *Only one user/admin can be logged in from the same browser/system at one time*

3.4 Performance Requirements

- *SQL language would be used to manage the databases as well as to perform the required functions*
- *There would be at least 10 tables in the database and these tables would be linked with one another through a relationship*
- *Each table (except admin table) would consist of at least 1000 tuples, so a total of 9000 plus tuples*
- *Each table would be updated in real time, as the user/admin is entering data*
- *The notification emails will be prepared and sent automatically as soon as an event occurs*
- *User and admin authentication would be very fast*
- *Great number of users can access the website at the same and perform their intended tasks*
- *Aiming for robust execution of various functions*
- *Pictures of each item would be displayed when the user/admin browses*

Chapter 4: Use Cases

4.1 Use cases for V.G.S

Revision History

Name	Date	Reason for Update	Version
VGS	09/10/2020	Initial draft	1.0

The various user classes identified the following use cases and primary actors for the Video Game Shop:

Primary Actor	Use Case
User	1. Register User 2. Sign in 3. Search Item 4. Shopping Cart 5. Rent 6. User History 7. Grouping 8. User Account Management 9. Store credit
Admin	10. Add Items 11. Selection of Most Bought Games 12. Stock Update 13. Games due / Charges due 14. Remove Items 15. Profit calculation 16. View Transactions 17. Back up

Use Case ID:	1		
Use Case Name:	Register User		
Created By:	Ali	Last Updated By:	Ali
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	The user creates an account for himself.
Trigger:	
Preconditions:	1. The User has a valid Email Address
Post conditions:	1. The user has been added to the database
Normal Flow:	1. The user clicks on register account 2. The systems displays a screen with multiple demands 3. The user enters a username, email and password 4. The systems check's if the email is already registered 5. The system check's if there is a similar username in the system, 6. The systems stores the username ,password and email
Alternative Flows:	After 4; if the email is already registered an error box shows on the screen saying so After 5; if the username is already registered an error box shows on the screen saying so
Exceptions:	
Includes:	None
Priority:	High
Frequency of Use:	Every time a new user wants to browse through website
Business Rules:	-

Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Sign in		
Created By:	Haseeb	Last Updated By:	Haseeb
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	The user logs into his/her account to purchase various items from the shop.
Trigger:	
Preconditions:	1. The user has a unique user name 2. The user knows his/her password
Post conditions:	1. The user has signed in
Normal Flow:	1. The user presses the sign in button 2. The user enters his/her username 3. The user then enters his/her password 4. The system checks the username whether is exists or not, is correct or not 5. Once username is authenticated, password is compared with the one stored in database.
Alternative Flows:	The users log in details are invalid Message is displayed telling the user that either username or password is invalid
Exceptions:	
Includes:	Authenticate the user and allow him/her to purchase on the website
Priority:	High
Frequency of Use:	Every time the user wishes to purchase anything from the website.
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Search Item		
Created By:	Raahem	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/05/2020

Actors:	User
Description:	User enters a query in the search bar and appropriate results are returned
Trigger:	
Preconditions:	None
Post conditions:	None
Normal Flow:	1. User enters query(e.g. name of game) in the search bar 2. The name of game is searched throughout the game table. 3. If found, the result is returned i.e. game details
Alternative Flows:	None
Exceptions:	If game does not exist in the database, message displayed e.g. not found
Includes:	None
Priority:	High
Frequency of Use:	High
Business Rules:	---
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	4		
Use Case Name:	Shopping cart		
Created By:	Sameer	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	Contains the information of the items the user wants to purchases.
Trigger:	The user clicks on add to cart on a particular item
Preconditions:	1. The user is interested in a particular item
Post conditions:	1. The item is added to user's cart and stock is updated accordingly
Normal Flow:	1. The user clicks on the item he/she is interested in. 2. The user selects add to cart and item is added to his cart.
Alternative Flows:	
Exceptions:	Item is out of stock.
Includes:	
Priority:	High
Frequency of Use:	Every time the user clicks on add to cart
Business Rules:	-
Special Requirements:	- the item must be in stock If user logs out and logs in, contents of cart should remain intact
Assumptions:	
Notes and Issues:	- the item is out of stock

Use Case ID:	5		
Use Case Name:	Rent a game		
Created By:	Haseeb	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	The user rents a game from the shop
Trigger:	
Preconditions:	1. The user must be ready to pay on Cash on delivery
Post conditions:	2. The game stock would be updated
Normal Flow:	1. The User selects the game he wants to buy from search or from trending section 2. The system updates the stock 3. The system notifies the admin through email what game to ship
Alternative Flows:	
Exceptions:	
Includes:	The stock of the game is updated
Priority:	High
Frequency of Use:	Every time the user wishes to purchase anything from the website.
Business Rules:	-
Special Requirements:	-The user can leave the rent game page at anytime
Assumptions:	The only system for payment available is cash on delivery
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	User History		
Created By:	Sameer	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	The user views his history
Trigger:	
Preconditions:	1. The user has been logged into his account
Post conditions:	History displayed to the user
Normal Flow:	1. The user clicks on view history 2. The system checks the history corresponding to the username, Games rented, due date, consoles bought 3. The history is displayed to the user
Alternative Flows:	None
Exceptions:	
Includes:	None
Priority:	Medium
Frequency of Use:	Every time the user wishes to view his/her history.
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Grouping		
Created By:	Ali	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	Grouping of the items with respect the genre, number of sales etc.
Trigger:	
Preconditions:	The user selects the category on which the items will be grouped
Post conditions:	The items are then displayed with respect to category selected
Normal Flow:	The system takes the data and performs the grouping of the table Based on sales or genre.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	Medium
Frequency of Use:	Medium
Business Rules:	-
Special Requirements:	Keeps track of the sales of each item.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	User account management		
Created By:	Sameer	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	User
Description:	The user changes his info
Trigger:	
Preconditions:	1. The user has a unique username 2. The user knows his/her password 3. The user is logged in
Post conditions:	The users log in information has been changed
Normal Flow:	1. The system demands the user's username and passwords 2. The system verifies the username and password 3. The system asks what to change Email/Username/Password 4. In case of Username and email the system verifies if a similar email or password is already in the system
Alternative Flows:	The users log in details are invalid Message is displayed telling the user that either username or password is invalid
Exceptions:	
Includes:	None
Priority:	Medium
Frequency of Use:	Every time the user wishes to change his/her details
Business Rules:	-

Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	**9**		
Use Case Name:	Store Credit		
Created By:	Ali	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	**User**
Description:	The user gets store credit after purchase of games
Trigger:	When two or more items are purchased
Preconditions:	1. User completes his/her shopping and proceeds to checkout
Post conditions:	Store credit is added to user account
Normal Flow:	1. User presses checkout button. 2. After pressing checkout button, if number of items purchased are greater than or equal to 2, user is liable for store credit 3. 20% of total is calculated and added as credit to user account
Alternative Flows:	
Exceptions:	Number of items are less than 2, so nothing happens, no change to store credit
Includes:	None
Priority:	High
Frequency of Use:	Every time the user completes shopping
Business Rules:	-
Special Requirements:	The number of items purchased should be greater than or equal to 2
Assumptions:	-
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	Add Items		
Created By:	Haseeb	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	Admin
Description:	Admin adds new, upcoming titles to the database
Trigger:	New stock arrival
Preconditions:	1. The admin needs to be logged in
Post conditions:	1. The game database is updated with the new tuples
Normal Flow:	1. Admin logs into his account 2. Admin enters the new games through a form into the database 3. Database is updated with the new titles
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	Whenever new games arrive (monthly)
Business Rules:	None
Special Requirements:	The Admin can leave the add games feature at any time.
Assumptions:	None
Notes and Issues:	None

Use Case ID:	11		
Use Case Name:	Selection of Most Bought Game		
Created By:	Haseeb & Sameer	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	08/10/2020

Actors:	Admin
Description:	Admin presses navigates to the highest sale page and the game with the greatest revenue is displayed
Trigger:	
Preconditions:	None
post conditions:	Most sold game is displayed on website
Normal Flow:	Once button is clicked, sql query is executed to select game with highest revenue from sales table. Game is found and displayed on the website homepage
Alternative Flows:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Occasionally (once a week)
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	12		
Use Case Name:	Charges due / Games due		
Created By:	Ali	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	Admin
Description:	Checks which games are overdue (not returned after renting)
Trigger:	Pressing check charges button
Preconditions:	1) The user has unpaid dues. 2) The user has not returned games
post conditions:	1) The user returns rented games
Normal Flow:	The system creates a mail based on the situation(arrival of stock ,unpaid dues etc.) The mail is sent to articular user.
Alternative Flows:	
Exceptions:	If no item is due, nothing happens
Includes:	
Priority:	Medium
Frequency of Use:	Variable frequency
Business Rules:	
Special Requirements:	
Assumptions:	Email is sent to the user(s) in question
Notes and Issues:	

Use Case ID:	13		
Use Case Name:	Remove Items		
Created By:	Sameer	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	Admin
Description:	Admin removes old obsolete items
Trigger:	Low Sale or items outdated
Preconditions:	1. The admin needs to be logged in
post conditions:	1. The game database is updated once the obsolete items are removed
Normal Flow:	1. Admin logs into his account 2. Admin enters the item to be deleted from the database 3. Database is searched to find item 4. Once found, admin deletes the tuple after confirming
Alternative Flows:	None
Exceptions:	The item to be deleted does not exist or is not found in the database, message displayed and nothing happens
Includes:	None
Priority:	Medium
Frequency of Use:	Low
Business Rules:	None
Special Requirements:	The Admin can leave the remove feature at any time.
Assumptions:	None
Notes and Issues:	None

Use Case ID:	14		
Use Case Name:	Stock Update		
Created By:	Haseeb	Last Updated By:	Raahem
Date Created:	05/10/2020	Date Last Updated:	07/10/2020

Actors:	Admin
Description:	The system notifies admin that limited stock for an item is available
Trigger:	The stock for an item falls below 5 units
Preconditions:	1. The stock of all items is above 5
post conditions:	2. Email regarding stock update is sent
Normal Flow:	The stock of an item falls below 5 units The system emails the retailer that stock of a particular item is low
Alternative Flows:	
Exceptions:	If items are in stock, nothing happens
Includes:	
Priority:	High
Frequency of Use:	Every time the stock of items falls below 5 units
Business Rules:	-
Special Requirements:	-
Assumptions:	Email is sent and received by the retailer
Notes and Issues:	-

Use Case ID:	15		
Use Case Name:	Profit Calculation		
Created By:	Raahem	Last Updated By:	Raahem
Date Created:	10/10/2020	Date Last Updated:	10/10/2020

Actors:	Admin
Description:	The admin wishes to calculate the profit made after a particular period of time
Trigger:	None
Preconditions:	1. Sales table exists with all the relevant fields required for calculation
Post conditions:	2. Profit is calculated and stored
Normal Flow:	1. Admin wishes to calculate the monthly profit 2. The total cost price of all items is subtracted from the total sale price of all items for that duration of time
Alternative Flows:	
Exceptions:	None
Includes:	
Priority:	High
Frequency of Use:	Medium
Business Rules:	-
Special Requirements:	-
Assumptions:	None
Notes and Issues:	-

Use Case ID:	16		
Use Case Name:	View Transactions		
Created By:	Raahem	Last Updated By:	Raahem
Date Created:	10/10/2020	Date Last Updated:	10/10/2020

Actors:	Admin
Description:	The admin wishes to view the transactions carried out for a particular period of time
Trigger:	None
Preconditions:	1. Admin needs to be logged in
Post conditions:	2. Transaction details for a particular period of time are displayed
Normal Flow:	1. Admin presses the view transactions button 2. He/she selects the date 3. Transaction details of that period displayed
Alternative Flows:	
Exceptions:	If there are no sales on a particular day, message displayed telling the admin that no transactions were carried out on this day
Includes:	
Priority:	Medium
Frequency of Use:	High
Business Rules:	-
Special Requirements:	-
Assumptions:	None
Notes and Issues:	-

Use Case ID:	17		
Use Case Name:	Back up		
Created By:	Raahem	Last Updated By:	Raahem
Date Created:	10/10/2020	Date Last Updated:	10/10/2020

Actors:	Admin
Description:	The admin wishes to back up the databases after an interval of time
Trigger:	None
Preconditions:	1. Admin needs to be logged in
Post conditions:	2. Database has been backed up on the network or on a local disk
Normal Flow:	1. Admin presses the backup button 2. He/she selects the tables that are to be backed up 3. Transaction details of that period displayed
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Weekly or monthly
Business Rules:	-
Special Requirements:	-
Assumptions:	None
Notes and Issues:	-

4.2 Use Case Point Calculation Sheet

Unadjusted Use Case Points

Item	Item Description	Complexity	Count	Weight	Weighted Count
1	Number of Actors	Simple		1	
		Average		2	
		Complex	2	3	6
2	Number of Use Cases	Simple	7	5	35
		Average	6	10	60
		Complex	4	15	60
Unadjusted Use Case Points (UUCP)					**161**

Complexity Factor

Factor	Description	Rating 0=Irrelevant 5=Essential	Weight	Weighted Rating
T1	Distributed system	3	2	6
T2	Response performance objectives	4	1	4
T3	End-user efficiency	4	1	4
T4	Complex internal processing	5	1	5
T5	Code must be reusable	3	1	3
T6	Easy to install	5	0.5	2.5
T7	Easy to use	5	0.5	2.5
T8	Portable	5	2	10
T9	Easy to change	4	1	4
T10	Concurrent	5	1	5
T11	Secure	4	1	4
T12	Access to 3rd parties	2	1	2
T13	User training facilities	3	1	3
Technical Factor (TF) = sum of weighted ratings				**55**
Technical Complexity Factor (TCF) = 0.6 + (0.01 x TF)				**1.15**

Environmental Factor

Factor	Description	Rating 0=Lowest 5=Highest	Weight	Weighted Rating
F1	Familiar with Rational UP	2	1.5	3
F2	Application experience	2	0.5	1
F3	Object-oriented experience	3	1	3
F4	Lead analyst capability	2	0.5	1
F5	Motivation	2	1	2
F6	Stable requirements	1	2	2
F7	Part-time workers	0	-1	0
F8	Difficult programming language	2	-1	-2
Environmental Factor (EF) = sum of weighted ratings				**10**
Environmental Value (EV) = 1.4 - (0.03 * EF)				**1.1**

Use Case Points

Use Case Points (UCP) = UUCP * TCF * EV	**203.665**