



- Gamer hub

Analysis and Design Document

Team 2

**Sean Corrigan
Rogelio Zamudio Jr.**

Contents

1 Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms, and abbreviations
- 1.4 References
- 1.5 Overview

2 System Architecture

- 2.1 System Overview
- 2.2 Architectural Overview
- 2.3 Decomposition Description
- 2.4 Design Rationale

3 Human Interface Design

3.1 Login and Registration

- 3.1.1 Input & Output
- 3.1.2 Actions
- 3.1.3 Pre and Post Conditions
- 3.1.4 Validation

3.4 Guides/Walkthroughs

- 3.1.1 Input & Output
- 3.1.2 Actions
- 3.1.3 Pre and Post Conditions
- 3.1.4 Validation

3.5 Reviews

- 3.1.1 Input & Output
- 3.1.2 Actions
- 3.1.3 Pre and Post Conditions
- 3.1.4 Validation

4 Component Design

4.1 Entities

5 Data Design

5.1 Forums/Discussions

5.1.1 Data Description

5.1.2 Registration and Authentication

5.1 So on and so forth

6 Database Directory / SCHEMA

6.1 Registration / Authentication

7 Full Database Model

8 Requirements Matrix

1 Introduction

1.1 Purpose

The purpose of this document is to outline the technical design of Geekd and provide and over for the site and its administrator functions.

Its main purpose it to:

- Provide links between functional and technical aspects
- Details the functionality which be provided by each component or group of components and show how the various components interact
- Provide a basis for Geekd's detailed design and development

1.2 Scope

The initial modules discussed from the SRS are:

- Login/Registration
- Search Engine

- News/RSS Feed
- Guides/Walkthroughs
- Reviews
- Forums/Discussions
- Support

The current scope of the project will focus on the following:

- Login
- Walkthroughs
- Reviews

1.3 Definitions Acronyms and Abbreviations

| Term | Definition |
|---------------------|---|
| User | Someone who interacts with the website |
| Admin/Administrator | System administrator that is given permission for managing and controlling the system |
| Thread | Created when a user creates a new post on the message board. Results in a new entry to the database that can branch into other threads. |
| Tag | Created by both users and admins but only approved by admins. This allows content to become searchable by specific terms. When a user or admin uses the search function, the terms in the search bar will first compare terms against tagged content. |
| Review | User or admin generated document that can be modified and approved by administrators and meant to be viewed by all parties using the website |
| Walkthrough | User or admin generated document that can be modified and approved by administrators and meant to be viewed by all parties using the website |
| Validation | The system will validate information by comparing it to its database as well as checks to ensure that items such as password length and uniqueness are correct |

1.4 References

Pressman, Roger S., and Bruce R. Maxim. *Software Engineering: A Practitioner's Approach*. 8th ed. New York: McGraw-Hill Education, 2015. Print.

1.4 Overview

The remainder of the ADD will feature the following components to be addressed: System Architecture, Human Interface Design, Component Design, Data Design, Database Directory, Full Database Model, Requirements Matrix. These sections will be ordered as listed with additional content categorically organized.

The System Architecture is organized into a System Overview, Architectural Design, Decomposition Description, and Design Rationale.

The Human Interface Design will feature detailed definitions of the system requirements specification for each use case. Uses cases will be mapped to pages with expected inputs and outputs, as well as actions taken as part of component executions and pre-conditions. There will be one section per module.

The Component Design section defines classes, attributes, methods, and relationships of classes. This will feature a subsection for Entities used in the design of Geekd. These entities will generally be described as: **Visitor**, **User**, **Admin**.

The Data Design is divided into Data Description and Registration / Authentication for each individual model.

The Database Directory / SCHEMA is defined in this section of the ADD. Fields, relationships, tables, and table fields will be well defined for individuals referencing this document.

The Full Database Model will be provided. This will include but not be limited to full entity relationships for all module described in the ADD.

The Registration Matrix section will provide a cross-reference that traces components and tables to requirements present in the SRS. These references will be provided via a table format to help identify each requirement with a design component.

2 System Architecture

2.1 System Overview

In order to create a functional website, some of the system architecture as it relates to infrastructure must be addressed. The server, storage, and network structure components will be assessed in order to provide an early analysis to the readers so that design risks can be mitigated before implementation.

2.2 Architectural Overview

System architecture defines the various components and their interactions in the context of the whole system. System architecture is the critical component that bridges the architectural gap between the application server and the application's business logic.

The primary function will be to use a server-side PHP application and a SQL based database. This server component will be comprised of a PHP interface, which will manage access to content of the site as well as the user's ability to generate content for the site. The SQL database will provide centralized storage for the synchronized data. The server will receive data from the site and will then store data in the database and synchronize it.

2.3 Decomposition Description

2.4 Design Rationale

Geekd primarily provides targeted results from user queries. A Client/Server model perfectly describes the typical user interaction.

3 Human Interface Design

3.1 Login and Registration

3.1.1 Registration

3.1.1.1 Input & Output

The registration page will allow users to sign up for a account to log in to the system. The following image describes a basic approach to the signup page and all the information the system will need to produce an account to engage in sessions with the user.

CREATE AN ACCOUNT

Please enter the following details.

First Name

Last Name

Email Address

Username

Feel free to use special characters and numbers in your username.

Password

Retype Password

SUBMIT

Legal disclaimer for individuals under the age of 18.
The content of this website may be inappropriate for users under the age of 18. User discretion is advised.

Input:

- **First name:** String representing the first name of the user. Minimum of five characters and a maximum of 30 characters.
- **Last name:** String representing the last name of the user. Minimum of five characters and a maximum of 30 characters.
- **Email:** User's email address used for communication and verification. Should match the standard email pattern "[name@example.com](#)". Minimum 10 characters, with a maximum of 125 characters. Standard regex will be used to verify valid email pattern.
- **Username:** String representation of the username. Can be a minimum of 6 characters and a maximum length of 20 characters. The following special characters are allowed: !@#\$%&(). All numbers are allowed. Alphabetic characters will not be case sensitive.
- **Password:** String representation of the user's password. The minimum length of the password will be 8 characters with a maximum of 25 characters. A minimum of one capital letter and one special character will be required. The following special characters are allowed: !@#\$%&(). All numbers are allowed. Alphabetic characters will be case sensitive for all instances of user passwords.
- **Re-enter password:** This string must match the string in the password field for the purposes of verification. This is the only use.

Output:

Upon successful creation of the user, the system will display a welcome message. If the user creation is not successful, the system will display a warning message. In the case that data is not valid, the system will show a list of the fields with invalid data in a red font.

Possible messages:

Success: "Welcome [Firstname] [Last Name]. Thank you for registering an account!"

- **Failure:** "The following fields are incorrect: Username - there is already a user with that account name."
- **Failure:** "The following fields are incorrect: Password - the passwords do not match. Please try again."
- **Failure:** "The following fields are incorrect: Password - the passwords do not match. The password does not match the requirements. Please see requirements. Please try again."

3.1.1.2 Actions

Upon submitting the form, using a submit button, the system will validate that all required fields have input, and that each field matches the required data in the required data format.

If the validation is successful, the system will run a query to store the user's information in the database.

If the validation failed, the system will abort the transaction and return an error message for each invalid field.

3.1.1.3 Pre and Post Conditions

Pre-condition: User has not registered with the system already. The system should validate that the email and username are unique by referencing them against the stored user values in the database.

Post-conditions: User record is stored successfully. A user session is created when the user logs on. A user will receive an email asking them to verify their email address.

3.1.1.4 Validation

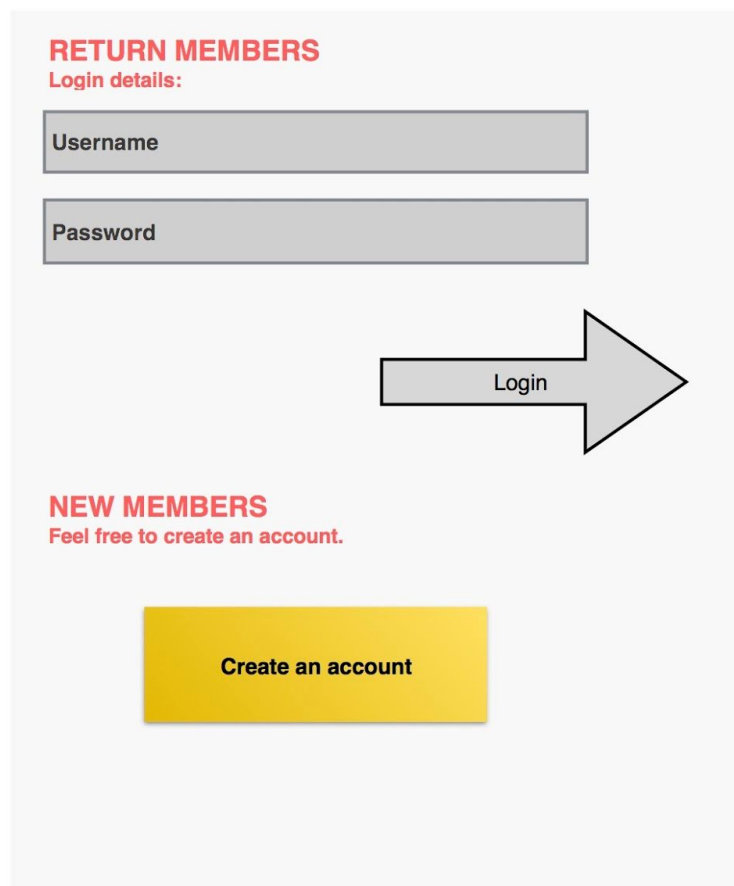
The following validation will be performed on the submitted data:

- **First name:** This field must be at least the minimum specified length and no more than the maximum length. Does not have to be unique.
- **Last name:** This field must be at least the minimum specified length and no more than the maximum length. Does not have to be unique.
- **Username:** This field must be unique and follow the requirement guidelines to only allow characters that are specified.
- **Email:** This field must be unique and must meet the requirements as implemented by the standard email regex. The general format will be "[name@example.com](#)" and will be validated by the regex.
- **Password:** This field must meet the password policy. The password must be stored using a mechanism that will prevent unauthorized access. In addition, all passwords that are stored in the database will be hashed.

3.1.2 Login

3.1.2.1 Input & Output

The login page will allow users to login and begin a session with the system. The following image describes a basic approach to the login page and all the information the system will need to begin a session. In addition, the page will feature a link that will take users to the account creation page if they do not have login credentials. Otherwise, the user can simply proceed to login.



The diagram illustrates a login page layout. At the top, the heading "RETURN MEMBERS" is displayed in red, followed by the subtext "Login details:" in red. Below this, there are two input fields: "Username" and "Password", both represented by gray rectangular boxes. To the right of these fields is a large gray arrow pointing to the right, with the word "Login" written inside its tail. Below the arrow, the heading "NEW MEMBERS" is displayed in red, followed by the subtext "Feel free to create an account." in red. At the bottom, there is a yellow rectangular button with the text "Create an account" in black.

Input:

- **Username:** This field will accept a string representing the username. The string can be a minimum of 6 characters and a maximum of 20 characters.
- **Password:** This will will accept a string representing the user's password. This must be a minimum of 8 characters. The user password will always contain, at minimum, one capital letter and one one number.

Output:

Upon a successful login a session will be created the for user. The user will be granted access to the information and services as specified to their role. A message will be displayed as well to notify the user of a successful login and a welcome message. If the user is unable to login a warning or error message will be displayed.

3.1.2.2 Actions

Once the user has entered their username and password, the "Login" button will prompt the site to authenticate the user. Their role will be determined and they will be granted access to the modules available to their appropriate role level. Based on the user's assigned role, the user might be rerouted to the administrator's page or the base user's page.

3.1.2.3 Pre and Post Conditions

Preconditions: Users and administrators both require a valid username and password, the username will be assigned when the user signs up. Administrators will be assigned usernames by the system but will choose their own passwords.

Postconditions: Successful entry of credentials will result in a portal page. Unsuccessful entry of credentials will result in an error page, where the user or administrator is required to re-submit credentials.

3.1.2.4 Validation

Both the username and password will be validated by comparing them to specific user credentials stored in the database.

Password security: The password must be stored using a secure mechanism that will prevent unauthorized access. All passwords will be hashed and stored as such in the database.

3.1.3 Change Password

3.1.3.1 Input & Output

The change password page will allow users to change and update their passwords. The following image describes a basic approach to change password page and all the information the system will need to make the necessary changes.

CHANGE PASSWORD
Please enter the following details.

Username

Password

Remember password guidelines.
Your password must contain, at minimum, one capital letter and one number. Strong passwords have combinations of numbers, letters, and special characters.

New Password

Retype Password

SUBMIT

Input:

- **Username:** This field will be a String. User must already exist in the database.
- **Current password:** This field will be a String. Must be a minimum of 8 characters and a maximum of 25. A minimum of one capital letter and one special character will be required. The following special characters are allowed: !@#\$%&(). All numbers are allowed. Alphabetic characters will be case sensitive for all instances of user passwords.
- **New password:** This field will be a String. Must be a minimum of 8 characters with a maximum of 25. A minimum of one capital letter and one special character will be required. The following special characters are allowed: !@#\$%&(). All numbers are allowed. Alphabetic characters will be case sensitive for all instances of user passwords.
- **Retype new password:** This field will also be a String. This field must be the exact same as the entry in the **New password** field.

Output:

If the user enters their current password, without errors, the system will store the new password and return a message to notify the user of a success.

If the user enters one or more of the fields improperly, the system will not store the new password and notify them of the following:

- If the current password was not entered correctly the user will receive a notification.
- If the new password does not meet the appropriate criteria the user will receive a notification.
- If the new password does not match in both the new password and retype new password fields, the user will receive a notification.

3.1.3.2 Actions

Upon submission of the form, the system will validate that the user has entered the correct password. To do so, the system will use the current session username and retrieve the current password from the database. The current password entered will be verified against the stored password and if successful, it will also validate both the "New password" and "Retype password" value are identical.

3.1.3.3 Pre and Post Conditions

Preconditions: The user must already be registered and the account must be active.

Postconditions: Successful transactions will update the user's password the the specified new values.

3.1.3.4 Validation

Current passwords will be validated against the stored password. New passwords will be validated against the "Retype password" value, which should match. Also, the new password must meet the password policy.

Password security: The password must be stored using a secure mechanism that will prevent unauthorized access. All passwords must also be hashed and stored as such in the database.

3.2.1 Walkthroughs

3.2.1.1 Input & Output

The walkthroughs page will allow to create walkthroughs. The following image describes a basic approach to walkthrough page and all the necessary or optional information.

WALKTHROUGH
Enter the following details:

Game Title

Game Genre

Publisher

Platform

Credits

If you have a .txt document you can copy and paste the contents of it directly into the box below.

Please copy and paste your walkthrough here. Or type it out.

SUBMIT

Input:

- **Game title:** String representing the title of the game. Minimum of a single character with a maximum of 125 characters. Case sensitive and all special characters and numbers allowed. This field is required.
- **Game genre:** String representing the genre of the game. Minimum of five letters with a maximum of 35. Only alphabetic letters allowed.
- **Publisher:** String representing the publisher of the game. Minimum of one character allowed with a maximum of 125. This field is required.
- **Platform:** String representing the platform of the game. Minimum 3 characters with a maximum of 25.
- **Guide:** String representing the guide for the game. Minimum 5000 characters with a maximum of 200000 characters. Special characters allowed. Numbers allowed as well. This field is required.
- **Credits:** String representing the individuals or organizations the author/user wants to credit. Limited to six characters with a maximum of 75 characters.

Output:

Upon successful creation of the walkthrough, the system will display a success message. The walkthrough will then be posted to the website and be available for viewing from the walkthrough portal.

If the guide is not posted the user will be notified of the failure and the cause for the failure. Causes for failure would entail the following:

- Failure to enter a String into the game title field.
- Failure to enter a String into the publisher field.
- Failure to enter a String into the Guide field.
- Timeout connection from the server.
- Any combination of the above.

3.2.1.2 Actions

Upon submitting the form, the system will validate that all the required fields have inputs in them and that each field matches the required data format. If the validation is successful, the system will run a query to store the user's information into the database.

If the validation fails, the system will abort the transaction and return an error message for each invalid field and cause of failure.

3.2.1.3 Pre and Post Conditions

Preconditions: The user must already be registered and the account must be active.

Postconditions: Successful transactions will be stored successfully. Successful transactions will also result in changes to the site and the guide will be available for viewing by all parties; those that do and don't have accounts with the site.

3.2.1.4 Validation

The following validations will be performed on the submitted data:

- **Game title:** String representing the title of the game. Minimum of a single character with a maximum of 125 characters. Case sensitive and all special characters and numbers allowed.
- **Publisher:** String representing the publisher of the game. Minimum of one character allowed with a maximum of 125. This field is required.
- **Guide:** String representing the guide for the game. Minimum 5000 characters with a maximum of 200000 characters. Special characters allowed. Numbers allowed as well.

3.2.2 Walkthrough Update

3.2.2.1 Input & Output

Many of the fields for the walkthrough update are the same as the original walkthrough page. All fields are allowed to be editable so that any mistakes may be changed if necessary and so that guides can be produced in stages, should they require continuous editing and updating. However, if a field is not modified then it will not be updated, allowing for fields to remain empty.

Input:

- **Game title:** String representing the title of the game. Minimum of a single character with a maximum of 125 characters. Case sensitive and all special characters and numbers allowed. This field is not required and may be empty.
- **Game genre:** String representing the genre of the game. Minimum of five letters with a maximum of 35. Only alphabetic letters allowed. This field is not required and may be empty.
- **Publisher:** String representing the publisher of the game. Minimum of one character allowed with a maximum of 125. This field is required.
- **Platform:** String representing the platform of the game. Minimum 3 characters with a maximum of 25. This field is not required and may be empty.
- **Guide:** String representing the guide for the game. Minimum 5000 characters with a maximum of 200000 characters. Special characters allowed. Numbers allowed as well. This field is not required and may be empty.
- **Credits:** String representing the individuals or organizations the author/user wants to credit. Limited to six characters with a maximum of 75 characters. This field is not required and may be empty.

Output:

Upon the successful update of the guide, the system will display a success message and redirect the user to the updated walkthrough's page. The system will update the entry in the database so that all other sessions on the site reflect the change.

If the guide is not updated the user will be notified of the failure and the cause for the failure. Causes for failure would entail the following:

- Timeout connection from the server.

3.2.2.2 Actions

Upon submitting the form, the system will validate that all the required fields have inputs in them and that each field matches the required data format. If the validation is successful, the system will run a query to store the user's information into the database.

3.2.2.3 Pre and Post Conditions

Preconditions: The user must already be registered and the account must be active.

Postconditions: Successful transactions will be stored successfully. Successful transactions will also result in changes to the site and the guide will be available for viewing by all parties; those that do and don't have accounts with the site.

3.2.2.4 Validation

When the system updates the walkthrough the system will check to see if the following values have changed by comparing them to the current values in the database:

- **Game title:** This field is not required and is allowed to be empty. If the field is empty or has the same value as the database, no action will be taken. Otherwise the system will compare the current value to the value in the field and proceed to update the value in the database with the value presented in the field.
- **Game genre:** This field is not required and is allowed to be empty. If the field is empty or has the same value as the database, no action will be taken. Otherwise the system will compare the current value to the value in the field and proceed to update the value in the database with the value presented in the field.
- **Publisher:** This field is not required and is allowed to be empty. If the field is empty or has the same value as the database, no action will be taken. Otherwise the system will compare the current value to the value in the field and proceed to update the value in the database with the value presented in the field.
- **Platform:** This field is not required and is allowed to be empty. If the field is empty or has the same value as the database, no action will be

taken. Otherwise the system will compare the current value to the value in the field and proceed to update the value in the database with the value presented in the field.

- **Guide:** This field is not required and is allowed to be empty. If the field is empty or has the same value as the database, no action will be taken. Otherwise the system will compare the current value to the value in the field and proceed to update the value in the database with the value presented in the field.
- **Credits:** This field is not required and is allowed to be empty. If the field is empty or has the same value as the database, no action will be taken. Otherwise the system will compare the current value to the value in the field and proceed to update the value in the database with the value presented in the field.

3.3 Reviews

3.1.1 Input & Output

The reviews page will allow users to create reviews. The following image describes a basic approach to reviews page and all the necessary information.

MEMBER REVIEWS
Enter the following details:

Game Title

Game Genre

Publisher

Platform

Remember the following:
Do not use profanity in your review. Using profanity will result in your account being suspended indefinitely.

Review text goes here.

SUBMIT

Input:

- **Game title:** String representing the title of the game. Minimum of a single character with a maximum of 125 characters. Case sensitive and all special characters and numbers allowed. This field is required.

- **Game genre:** String representing the genre of the game. Minimum of five letters with a maximum of 35. Only alphabetic letters allowed.
- **Publisher:** String representing the publisher of the game. Minimum of one character allowed with a maximum of 125. This field is required.
- **Platform:** String representing the platform of the game. Minimum 3 characters with a maximum of 25.
- **Review:** String representing the guide for the game. Minimum 5000 characters with a maximum of 200000 characters. Special characters allowed. Numbers allowed as well. This field is required.

Output:

Upon successful creation of the review, the system will display a success message. The review will then be posted to the website and be available for viewing from the reviews portal.

If the review is not posted the user will be notified of the failure and the cause for the failure. Causes for failure would entail the following:

- Failure to enter a String into the game title field.
- Failure to enter a String into the publisher field.
- Failure to enter a String into the review field.
- Timeout connection from the server.
- Any combination of the above.

3.1.2 Actions

Upon submitting the form, the system will validate that all the required fields have inputs in them and that each field matches the required data format. If the validation is successful, the system will run a query to store the user's information into the database.

If the validation fails, the system will abort the transaction and return an error message for each invalid field and cause of failure.

3.1.3 Pre and Post Conditions

Preconditions: The user must already be registered, the account must be active, and the walkthrough must already exist.

Postconditions: Successful transactions will be store. Successful transactions will also result in changes to the site and the walkthrough will be available for viewing in the reviews section of the site. Both user and

anonymous visitors to the site will be allowed to access the walkthroughs. Only the user that created the review is allowed to update or modify it.

3.1.4 Validation

The following validations will be performed on the submitted data:

- **Game title:** String representing the title of the game. This field will be checked to ensure that it conforms to the specified requirements and will result in an error message if it does not.
- **Game genre:** String representing the genre of the game. This field will be checked to ensure that it conforms to the specified requirements and will result in an error message if it does not.
- **Publisher:** String representing the publisher of the game. This field will be checked to ensure that it conforms to the specified requirements and will result in an error message if it does not.
- **Platform:** String representing the platform of the game. This field will be checked to ensure that it conforms to the specified requirements and will result in an error message if it does not. This field is not required and will not result in an error message if the field is empty.
- **Review:** String representing the guide for the game. This field will be checked to ensure that it conforms to the specified requirements and will result in an error message if it does not.

4 Component Design

ENTITIES

User: The User class represents both customer and staff entities. It is the base class for all user information. This class is responsible for managing the user personal information as well as defining the formatting of names.

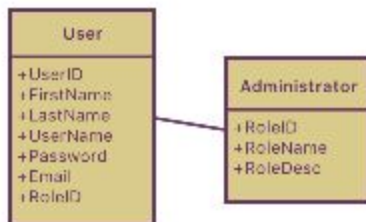
User Attributes/Fields:

- **User ID:** unique identifier for a single User instance
- **FirstName:** The first name of the user
- **LastName:** The last name of the user
- **UserName:** A unique string chosen by the user to gain access to the system
- **Password:** a string representing the password selected by the user to gain access to the Web site. It should be no less than 8 characters
- **Email:** a string representing a valid email address in the form of [account]@[domain].[suffix]
- **RoleID:** An association to the Administrator entity. It defines the type of the current user instance

Relationships:

- **Admin:** Each user has a one to one relationship with a type entity.
- **CatalogueObject:** Each item in the CatalogueObject table has a one to one relationship with with a single user.

Admin: Entity that identifies one of the possible types of users. Possible options might be "customer" or "admin." Users who are not signed in receive the type "guest".



GameInfo: The GameInfo class represents all games in the database. It is the base class for all game information. This class is responsible for managing the game production information as well as defining the formatting of information.

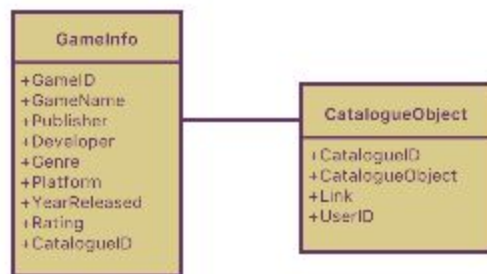
GameInfo Attributes/Fields:

- **GameID:** Unique identifier for a single CatalogueObject instance.
- **GameName:** The name of the game.
- **Publisher:** The name of the game's publisher.
- **Developer:** The name of the game's developer.
- **Genre:** The type of game.
- **Platform:** The platform this game is available on.
- **YearReleased:** The game's release year.
- **Rating:** The game's ESRB/PEGI rating.
- **CatalogueObject:** An association to the CatalogueObject table, which defines the available walkthroughs or other items available for the game.

Relationships:

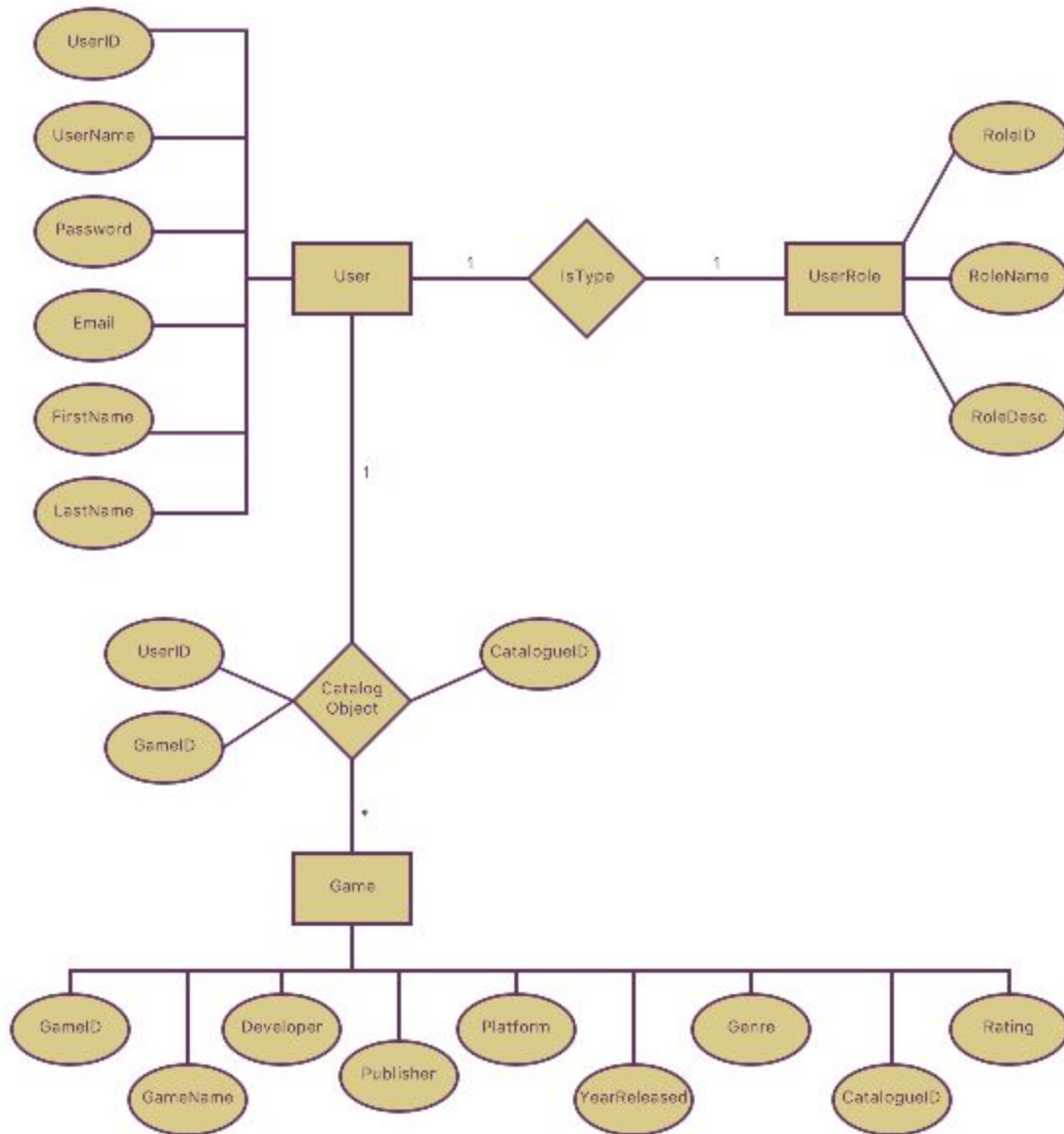
- **CatalogueObject:** Each item in the CatalogueObject table has a from one to many relationships with a GameID, depending on number of game platforms

CatalogueObject: Entity that identifies the items available for each individual games. Each game may have from zero to multiple items associated with it. Some items may be associated with games on multiple platforms, if applicable. These items are user-submitted walkthroughs and reviews. A walkthrough or review has a unique CatalogueObject entry, and a CatalogueID which may apply to different editions or platforms of the same game.



5 Data Design

The following diagram describes the relationships between the user objects and the array of information for the catalog objects.



6 Database Directory

| GameCatalogue Table | | | |
|---------------------|-------------|----------|--|
| Entity | Type | Value | Description |
| GameID | INT | Not null | PK. Unique identifier of game. |
| GameName | varchar(50) | Not null | Game's name. |
| Publisher | varchar(50) | Not null | Game's publisher. |
| Developer | varchar(50) | Not null | Game's developer. |
| Genre | varchar(20) | Not null | Game's genre. |
| Platform | varchar(20) | Not null | Game's platform. |
| YearReleased | varchar(4) | Not Null | Game's year of release. |
| Rating | varchar(5) | Null | Game's age rating. |
| CatalogueID | INT | Not null | FK. Relationship key to Catalogue Items. |

| CatalogueObject Table | | | |
|-----------------------|-------------|----------|---|
| Entity | Type | Value | Description |
| CatalogueObject | INT | Not null | PK. Unique identifier for guide/review. |
| CatalogueID | INT | Not null | FK. Unique identifier for game's items. |
| Data | varchar(50) | Not null | Guide/review data. |
| UserID | INT | Not null | FK. Catalogue item's creator. |
| OK | Boolean | Not null | Boolean signifier of administrative |

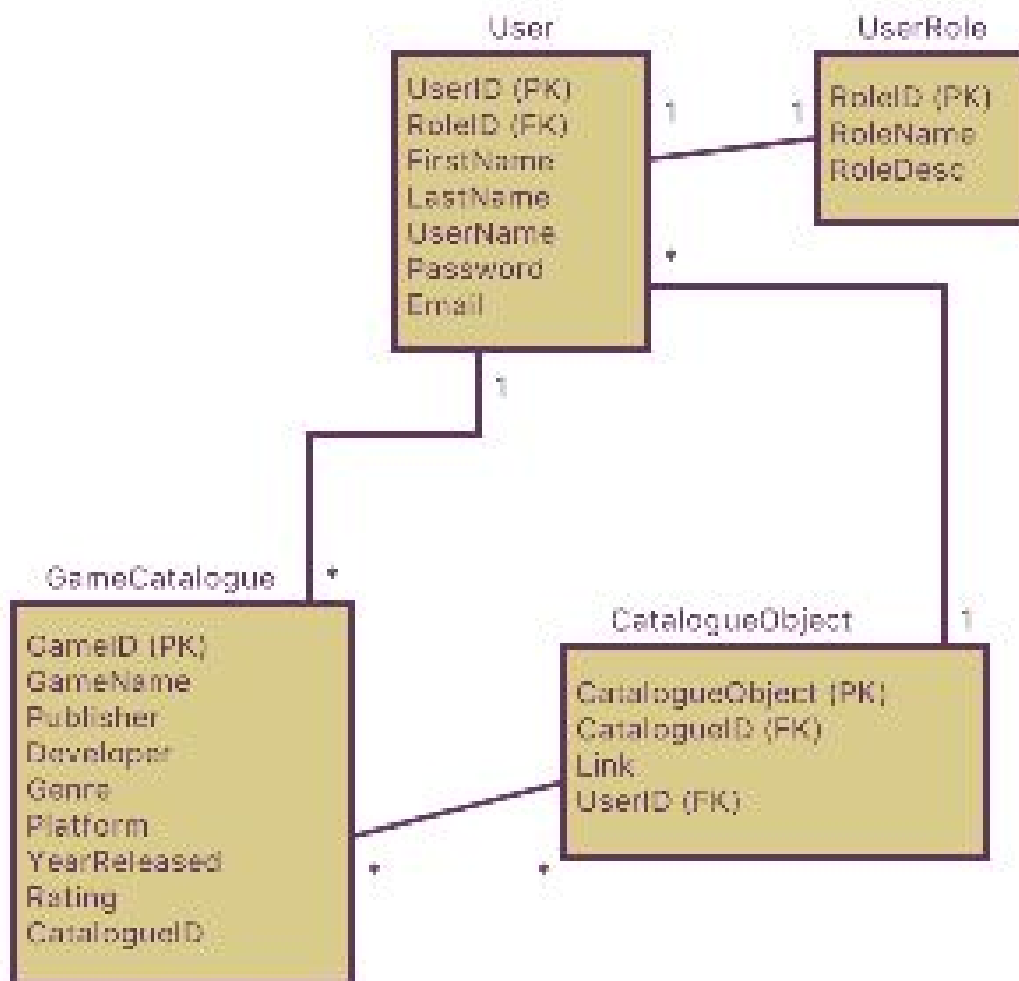
| | | | |
|--|--|--|-----------|
| | | | approval. |
|--|--|--|-----------|

| User Table | | | |
|------------------|-------------|----------|--|
| Entity | Type | Value | Description |
| UserID | INT | Not null | PK. Unique identifier of a user. |
| RoleID | INT | Not null | FK. Relationship key to User_Role table. |
| FirstName | varchar(20) | Not null | User's preferred first name. |
| LastName | varchar(20) | Not null | User's preferred last name. |
| UserName | varchar(50) | Not null | User's... user name. |
| Password | varchar(32) | Not null | User's password. |
| Email | varchar(50) | Not null | User's contact email. |
| DateReg | date | Not null | Date Registered |

| UserRole Table | | | |
|-----------------|-------------|----------|-----------------------------|
| Entity | Type | Value | Description |
| RoleID | INT | Not null | PK. Used to flag user role. |
| RoleName | varchar(15) | Not null | Name of user role. |
| RoleDesc | varchar(50) | Not null | Description of user role. |

7 Full Database Model

The following describes the database model and relationships between tables. Associations are related between Primary Keys(PK) and Foreign Key(FK).



8 Requirements Matrix

| Module | Use Case | Design Component |
|--------------------|-----------------|---------------------------------------|
| Login/Registration | Registration | SRS 3.1.3, Scenario (1) User Table |
| | Login | SRS 3.1.3, Scenario (2) User Table |
| | Change Password | SRS 3.1.3, Scenario (3) User Table |

| Module | Use Case | Design Component |
|---------|-------------|--|
| Reviews | Post Review | SRS 3.5.3, Scenario (2) User Table GameCatalogue Table |

| | | |
|--|-------------|--|
| | View Review | SRS 3.5.3, Scenario (1) User Table GameCatalogue Table |
|--|-------------|--|

| Module | Use Case | Design Component |
|--------------|------------------|--|
| Walkthroughs | Post Walkthrough | SRS 3.4.2, Scenario (2) User Table GameCatalogue Table |
| | Edit Walkthrough | SRS 3.5.3, Scenario (2) User Table GameCatalogue Table |
| | View Walkthrough | SRS 3.4.2, Scenario (1) User Table GameCatalogue Table |