SENG 401 Final Project Report

Github Repository Link: https://github.com/Apostolos-sc/HouseOfCards

Live Website Link: https://goldenagesolutions.ca/HouseOfCards/index.php

Group Number: 31

Group Members:

- Ethan Winters
- Sayma Haque
- Nick Lam
- Carter Marcelo
- Jamie Stade
- Alexander Sembrat
- Apostolos Scondrianis

Project Description:

The software project our group has completed, titled House of Cards, is an online website where users can create accounts to then rate and discuss their favorite card games. This involves a comment system and a rating system much like you would see on a social media site, where each game can be rated and saved as a favorite by users and comments about the game can be posted on its page.

Purpose of the Project:

This project will provide an online environment for card game players to browse rules for many popular playing card games. Unlike similar card game websites, this platform will feature much more user interactivity. A website will provide users the ability to favorite card games they like, rate card games, and leave comments on a card games' page.

Project Architecture:

The selection of the MVC architecture model for this project was due to the large amount of organization and modularity this architecture allows for. The division of the data system (model) from the various interfaces (views) allows for additional functions and features to be added quite easily. These modules allow for the different functions to be accessed in various different manners from various places in the system. There are a few main modules to the system:

Session Controls: Files such as index.php, logout.php and userpage.php are located outside of the view modules, yet are key to the actual layout of the webpage. They are utilized by the view functions to construct web pages as they are required. They allow for the login sessions to be displayed correctly on the page and for functions such as search to work correctly in user sessions.

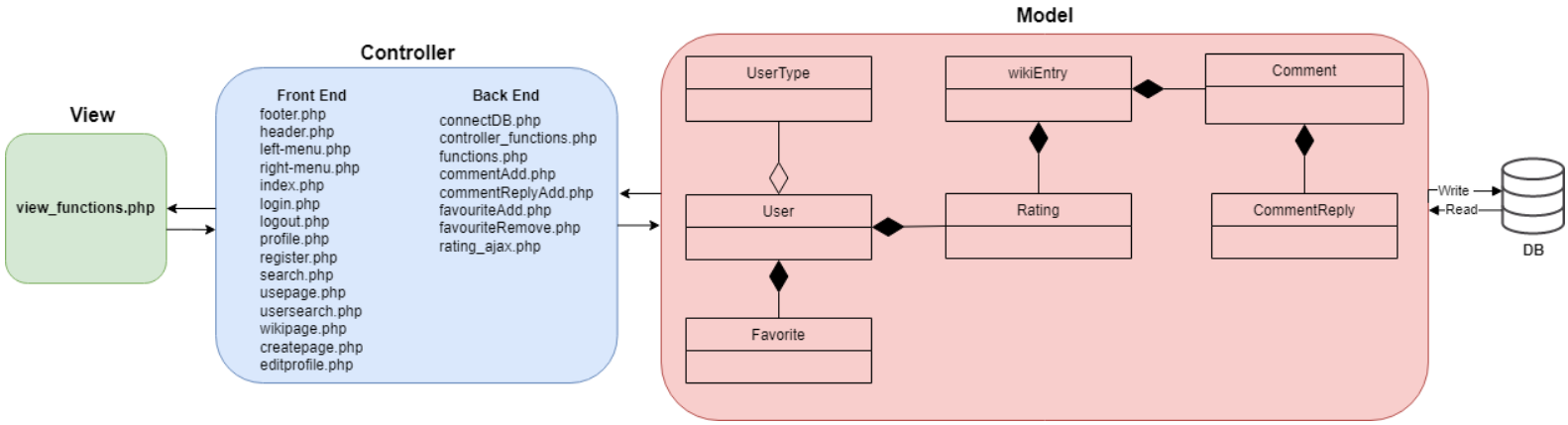
Data/Model Module: These files are located inside the model module, and are used to organize and maintain the structure of the various objects seen in this project. These include favourite.php, rating.php, user.php and they all focus on a different object type in the project. All of them reference the database to store and retrieve object data and the connections between said objects.

Controller Module: These files are located inside the controller module and are used to establish a connection between the view system and various aspects of the model. This is also where the database connection class is located, which allows the model to connect to the database to access the stored objects.

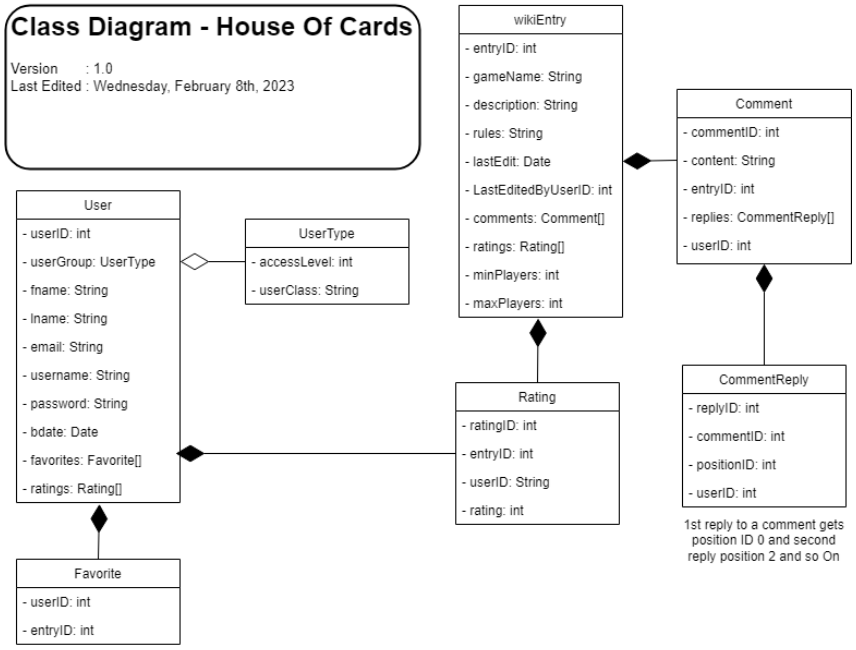
View Module: This module is where the session control functions are utilized to build the correct view of the webpage. This is crucial for various systems to ensure the various functions and elements are displayed correctly. This module also focuses on setting up the correct elements based on login conditions.

Design Documents:

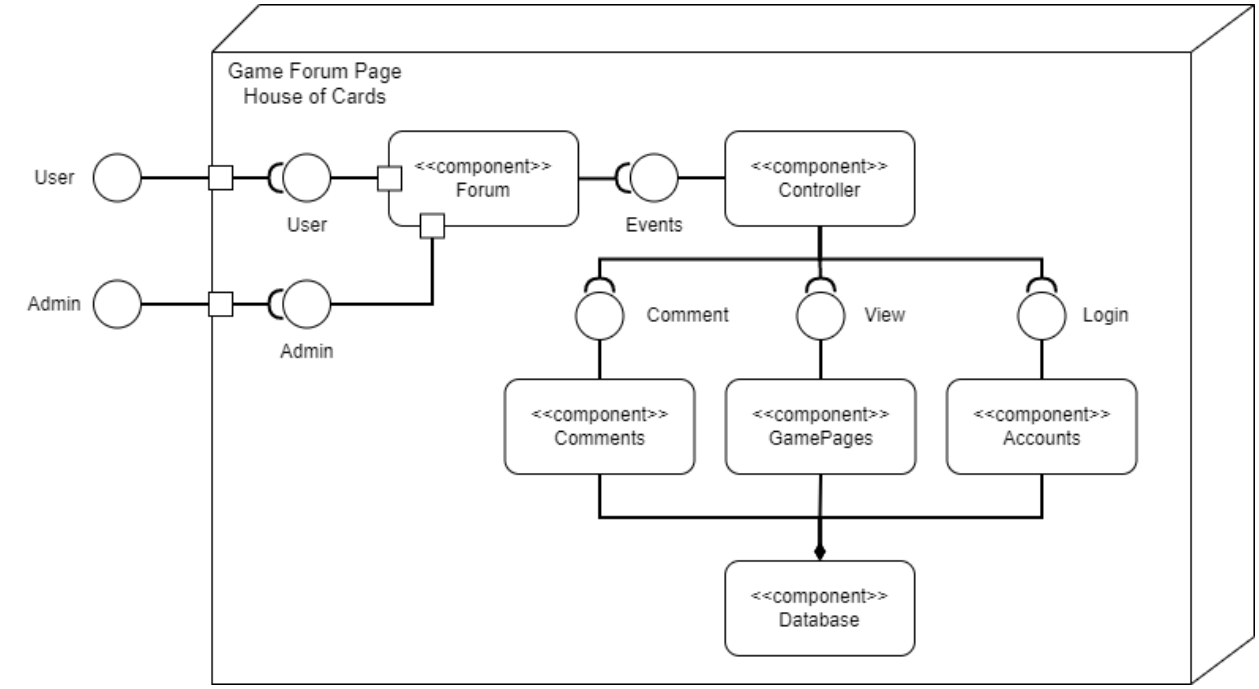
Architecture Diagram



Class Diagram

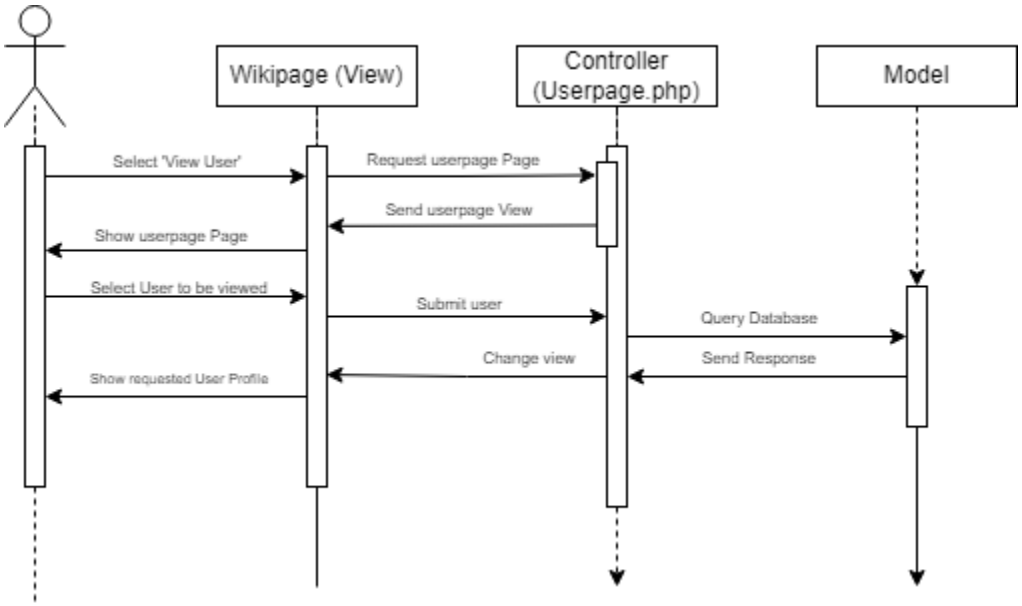


Component Diagram

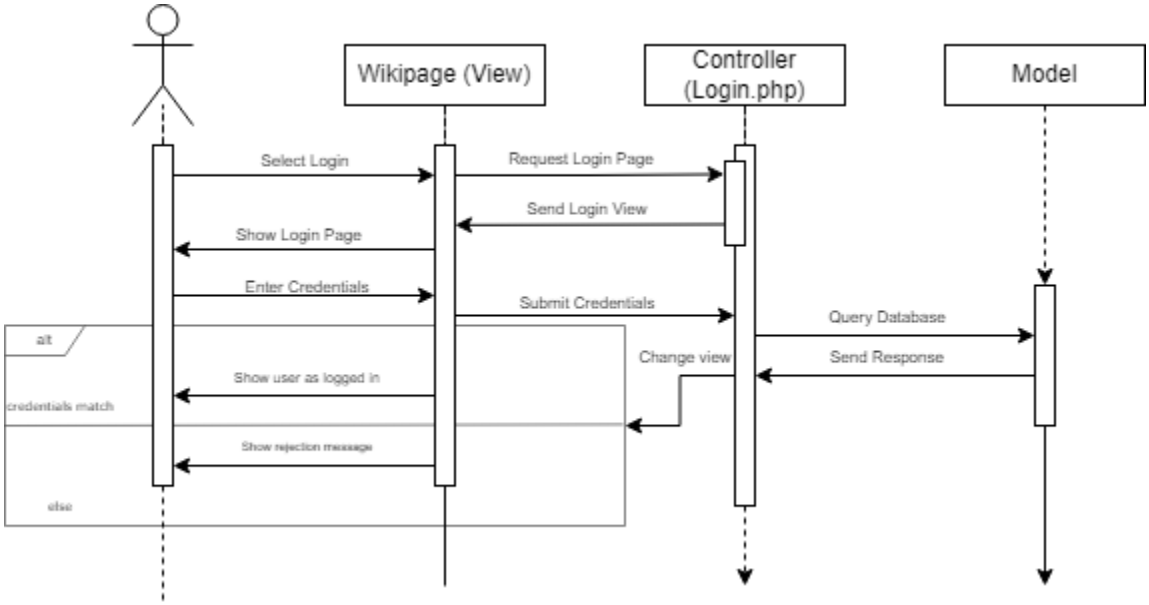


Sequence Diagrams

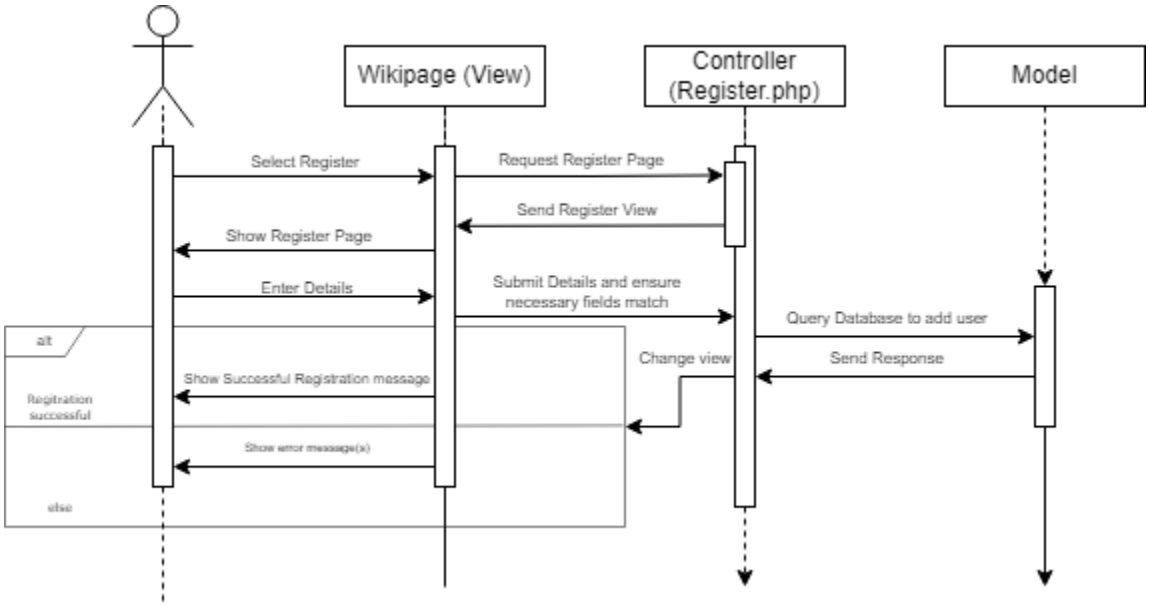
View Users



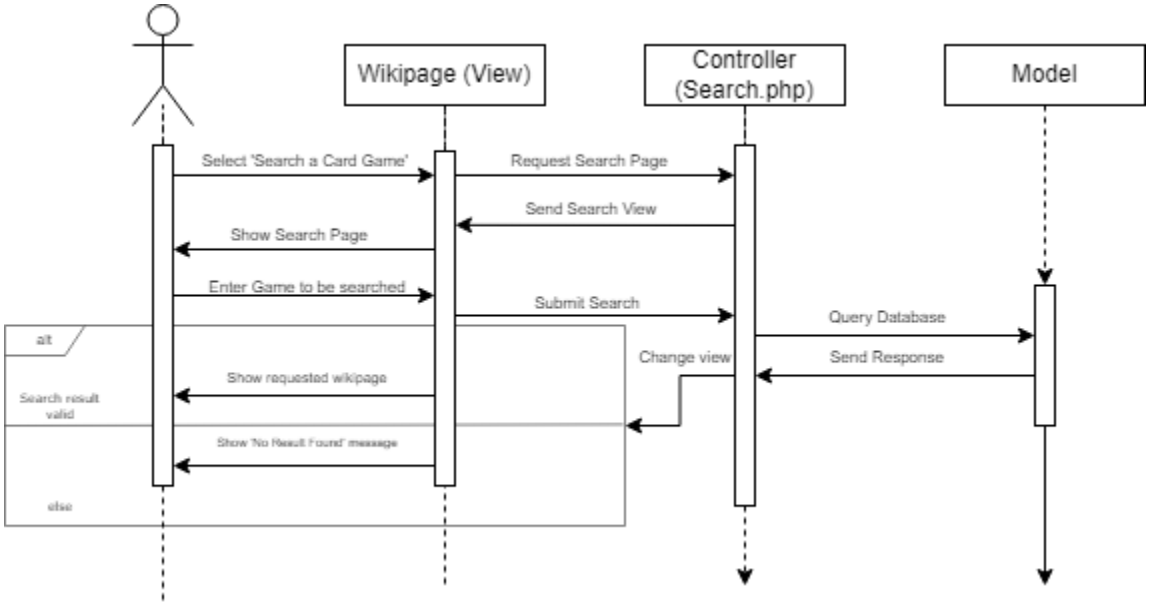
Login



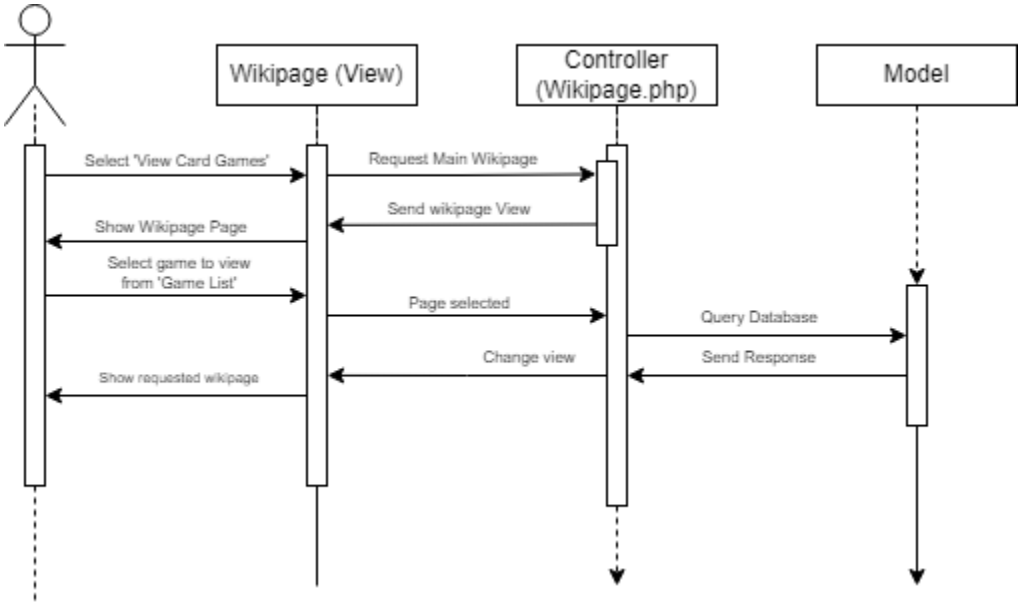
Register



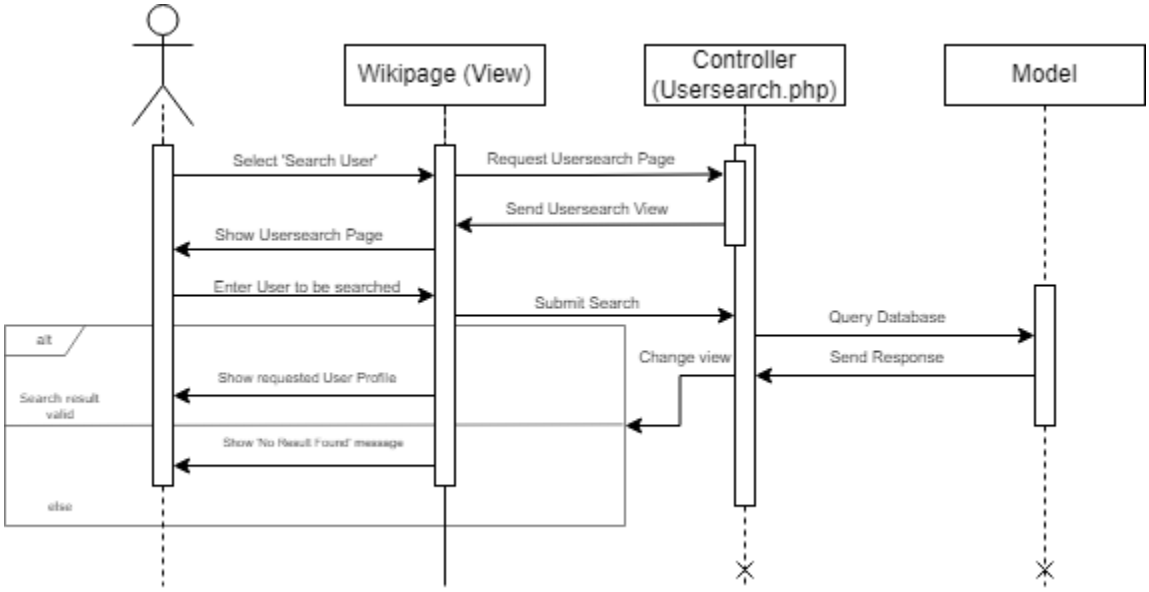
Search Game



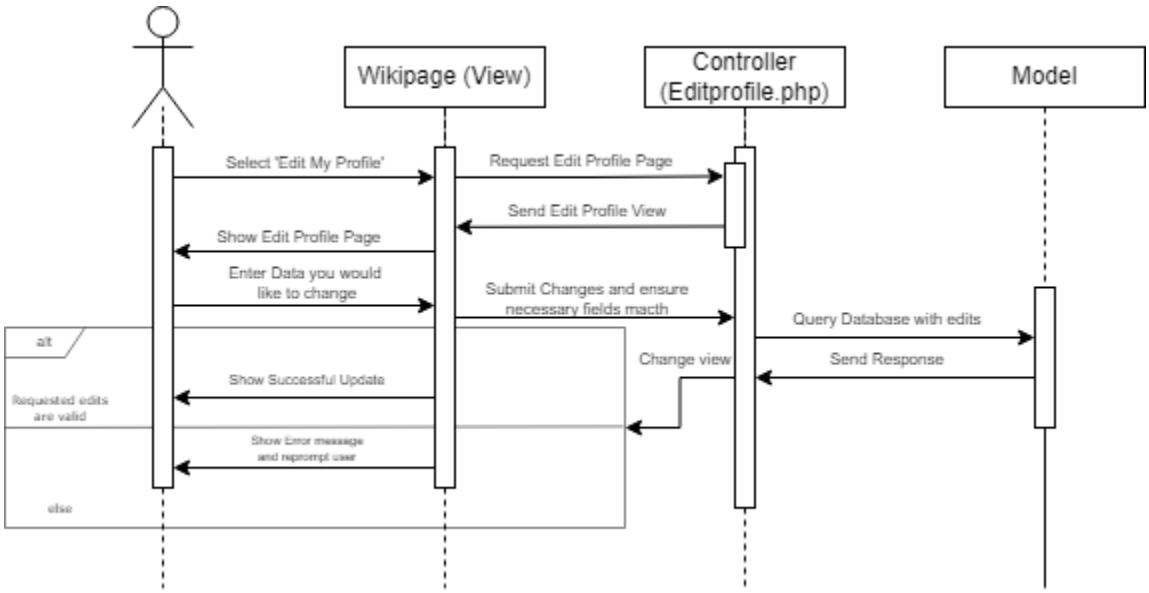
View Game



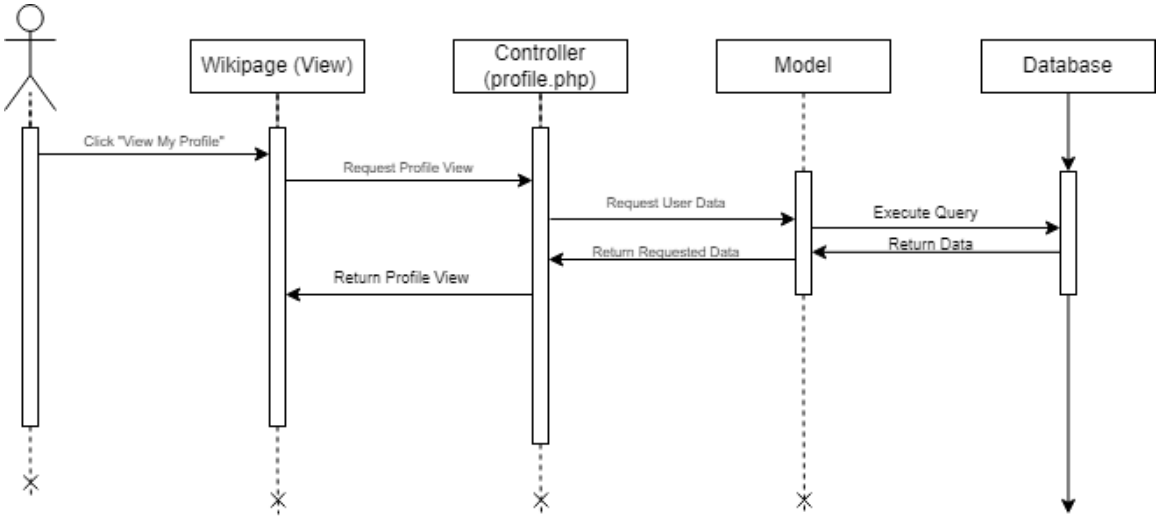
Search User



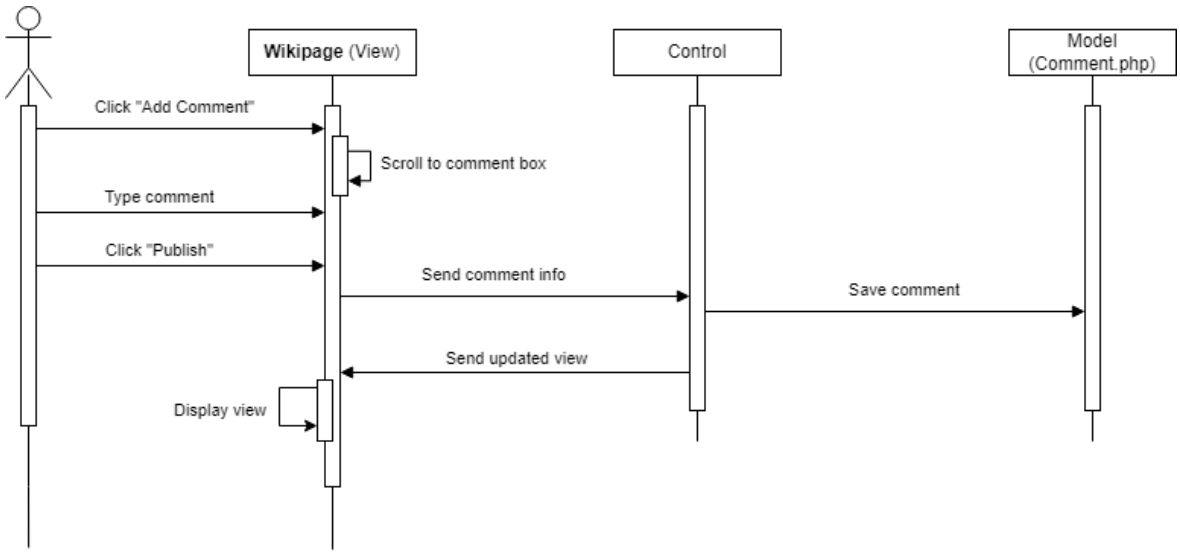
Edit Profile



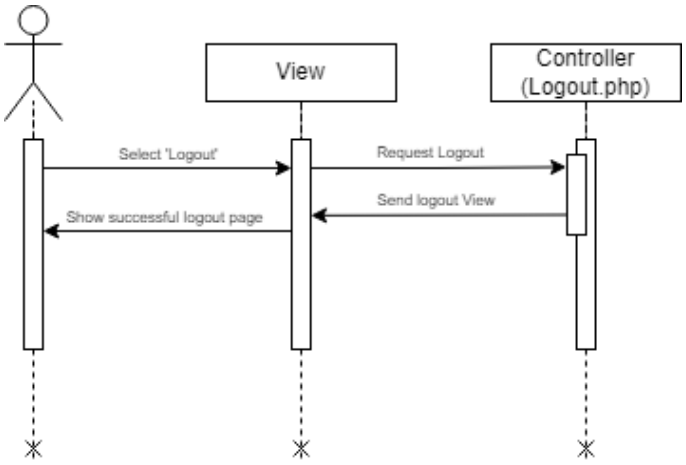
View My Profile



Comment On Card Game



Logout



RTM

SENG 401 Project				
House of Cards - Online Wiki for Card Games				
Business Requirements		Functional Requirements		Priority
Business Requirement ID#	Business Requirement Description	Functional Requirement ID#	Functional Requirement Description	Priority Rating (1-5)
1	Online Access	1	Users can access page online	1
		2	Users can access page on different device platforms	1
2	Account System	3	Users can register an account	1
		4	Users can login	1
		5	Users can rate games	3
		6	Users can favourite game pages	2
		7	Users can view their profile	2
		8	Users can edit their profile	3
		9	Admins have separate users type for admin privilages	1
		10	Registered users have separate user type	1
3	Game Pages	11	Game pages contain information, rating, comments, and edit history	1
		12	Users can select game pages	1
		13	Users can view information for each game page	1
		14	Users can search for a game page	3
		15	Users can view other users	5

		16	Users can search for a user	5
		17	Users can read comments for each game page	4
		18	Users can post a comment on a game page	4
		19	Users can reply to a comment on a game page	5
		20	Admin can add game pages	1
4	Activity History	21	Game pages have last edit date/time	3
		22	Comments and replies have associated users and date/time	3

Business Requirements		Non-Functional Requirements	
Business Requirement ID#	Business Requirement Description	Non-Functional Requirement ID#	Non-Functional Requirement Description
1	Performance	1	The page must be responsive
2	Usability	2	The page must be easy to navigate
		3	The page can be accessed from multiple device platforms
3	Availability	4	The page must be available at all times
		5	The page must be easy to update
4	Security	6	User information cannot be accessible to other users

Testing

To test the system selenium was used, a number of test cases were created to test the functional requirements in the RTM above. The tests are detailed below, all of the tests pass, and the input data shown is the basic data, some tests require a large amount of data or data that is just text assertions not credentials or other important info, if you wish to see the additional data it is contained in the testing suite. The testing suite is part of our submission.

Testing				
Test Case ID#	Test Case Name	Test Case Description	Related Functional Req. ID#	Data Used
1.1	CheckMainElements	Ensure web page is accessed correctly by checking all elements of the main page	1	N/A
3.1	RegisterValid	tests creating a new account with all valid parameters	3	Username: testuser1 Password: abcdABCD1234!@#$ Email: testuser4952431@gmail.com
3.2	RegisterUsernameTaken	tests creating a new account with a taken username	3	Username: adminjamie Password: abcdABCD1234!@#$ Email: testuser4952431@gmail.com
3.3	RegisterUsernameShort	tests creating a new account with a username less than 10 characters	3	Username: testuser Password: abcdABCD1234!@#$ Email: testuser4952431@gmail.com
3.4	RegisterPasswordShort	tests creating a new account with a password less than 8 characters	3	Username: testuser4321 Password: abcd123 Email: testuser4952431@gmail.com
3.5	RegisterEmailNotMatch	tests creating a new account when repeat email does not match the original email	3	
3.6	RegisterPasswordNotMatch	tests creating a new account when repeat password does not	3	

		match the original password		
4.1	ValidLogin	tests a user logging in with valid login credentials	4	Username: normaluser Password: password
4.2	InvalidUsernameValidPassword	tests a user logging in with an invalid username and valid password	4	Username: notrealuser Password: password
4.3	ValidUsernameInvalidPassword	tests a user logging in with an valid username and invalid password	4	Username: normaluser Password: notpassword
4.4	InvalidUsernameInvalidPassword	tests a user logging in with an invalid username and invalid password	4	Username: notrealuser Password: notpassword
4.5	LoginLogout	tests a user logging out	4	Username: normaluser Password: password
5.1	RateGame	Tests rating a game	5	Username: adminnick Password: adminpass
5.2	RateGameInvalid	Tests rating a game with a invalid rating	5	Username: adminnick Password: adminpass
6.1	AddFavourite	tests a user adding a card game to their favourites	4, 6	Username: normaluser Password: password
6.2	RemoveFavourite	tests a user removing a card game to their favourites	4, 6	Username: normaluser Password: password
7.1	ViewProfile	tests a user viewing their own profile	4, 7	Username: adminnick Password: adminpass
8.1	EditProfilePassword	tests a user changing the password for their account	4, 8	Username: normaluser Password: password NewPassword: password2
8.2	EditProfileEmail	tests a user changing the email for their account	4, 8	Username: normaluser Password: password Email: msnormal@gmail.com

8.3	EditProfileName	tests a user changing the first and last name for their account	4, 8	Username: normaluser Password: password FirstName: Girl LastName: Abnormal
8.4	EditProfileDOB	tests a user changing the date of birth on their account	4, 8	Username: normaluser Password: password
8.5	EditProfileMismatch	test a user changing their email when repeat email does not match	4, 8	Username: normaluser Password: password Email: msnormal@gmail.com Email2: missnormal@gmail.com
9.1	AdminPanel	tests admin logging in and accessing the admin panel	9	Username: adminnick Password: adminpass
10.1	RegisteredUserMenus	Tests if registered users have additional menus; search for user and profile menus	10	Username: normaluser Password: password
10.2	RegisteredUserGameFunctions	Tests if registered users can favourite, rate, comment and reply	10	Username: normaluser Password: password
12.1	SelectGamePage	tests a user selecting a page	11, 12	N/A
13.1	ViewGamePage	verifies elements on game page	11, 12, 13	Game Name: Go Fish
14.1	SearchLowercase	test user searching for a valid game in lowercase	14	Game Name: blackjack
14.2	SearchUppercase	test user searching for a valid game in uppercase	14	Game Name: BLACKJACK
14.3	SearchMixedcase	test user searching for a valid game with mixed case	14	Game Name: Blackjack
14.4	SearchInvalid	test user searching for an invalid game	14	Game Name: jack black

15.1	ViewOtherUsersPage	test user viewing another user's page from the user menu	4, 15	Username: normaluser Password: password
15.2	ViewOtherUsersComment	test user viewing another user's page from a comment	4, 15	Username: adminnick Password: adminpass
16.1	SearchUser	test user searching for an existing user	4, 9, 10, 16	Username: adminnick Password: adminpass User Search: adminsayma
16.2	SearchUserInvalid	test user searching for an invalid user	9, 10, 16	Username: adminnick Password: adminpass User Search: admincarterr
17.1	VerifyComments	verify that comments appear under game pages	17	N/A
18.1	PostComment	test user posting a comment	9, 10, 18	Username: normaluser Password: password Comment: I like jack black.
19.1	PostCommentReply	test user replying to an existing comment	9, 10, 17, 19	Username: normaluser Password: password Comment: even cooler the jack black
20.1	AddGamePage	try to add a new game with valid fields	9, 20	Username: adminnick Password: adminpass Game Name: Snap
20.2	AddGameZeroPlayers	try to add a new game with minimum players set to 0	9, 20	Username: adminnick Password: adminpass
20.3	AddGameNegativePlayers	try to add a new game with minimum players set below 0	9, 20	Username: adminnick Password: adminpass
20.4	AddGameNonNumericPlayers	try to add a new game with a non-numeric value for players	9, 20	Username: adminnick Password: adminpass
20.5	AddGameMinGreater	try to add a new game with minimum players	9, 20	Username: adminnick Password: adminpass

		greater than maximum players		
20.6	AddGameEmptyFields	try to add a new game with one or more empty fields	9, 20	Username: adminnick Password: adminpass
21.1	VerifyDateTimeGame	verify that appropriate date and time are added to edited game pages	12, 13, 21	N/A
22.1	VerifyDateTimeComment	verify that appropriate date and time are added to new comments	17, 22	N/A

Other System Information

Users and Permissions:

Guests

Guest users are a subtype of users that don't have accounts on the website. They do not have to log in, but cannot post comments or replies; they can only view the comments and pages. Guests may only view and search for card games.

Regular Users

Regular users must log in. They are able to access the various pages on the wiki containing the different card games. These pages include the information regarding the game and an option to leave comments and rate the various games. Regular Users can post and reply to comments here as well as favourite the games. They may also search for and view other users profiles.

Admins

Administrators are the final type here, and they are required to log in, and have access to everything and all options that users do, yet they also can create new pages, edit the content on pages, such as game rules and delete comments that violate the guidelines.

Web Pages:

- Wikipage: The page where the games and comments for them are displayed.
- Search: Used to search for a game.
- Userpage: Used to view users (User only access).
- Userearch: Used to search for a user (User only access).
- Login: Used for users to login.
- Logout: Used for users to logout.
- ViewMyProfile: Used for a user to view their profile information.
- EditProfile: Used for a user to edit their profile information.
- CreatePage: Used to create a new game entry (Admin only access).

Files, Classes and their Purpose:

There are a wide variety of classes seen in this project, each of which has a unique purpose.

Session Controls / Other:

- commentAdd.php: This file is used to add a comment to a wiki entry, a Comment object is attached to a game.
- commentReplyAdd.php: This file is used to add a comment reply to a comment, a CommentReply object is attached to a comment.
- Createpage.php: This file is used by administrators to add a new game to the wiki.
- editprofile.php: This file is used by users to edit their profile. They may change things such as their password, name, email, etc.
- favouriteAdd.php: This file is used to add a game to favourites for a user, when a user favourites a game it is recorded.
- favouriteRemove.php: This file is used to remove a game from a users favourites.
- index.php: This file is the default start page for the website.
- Login.php: This file is the login page for the website, a user inputs their credentials and if they are valid a login session will be created.
- Logout.php: This file is the used when a user wants to logout, the users session is destroyed.
- Profile.php: The webpage where users can view their profile information.
- Rating_ajax.php: This file is used to help display the rating of a game.
- Register.php: The webpage where guests can register to become users.
- Search.php: The webpage where guests or users can search for a game.
- userpage.php: The webpage that displays a list of the websites users
- Usersearch.php: The webpage where users can search for other users.
- Wikipage.php: The webpage where the game pages and their comments are displayed.

Model:

- CommentReply.php: This class focuses on replies to comments posted on pages in the wiki. It has getters and setters to get the information and allows comment replies to be organized and searched for by comment ID and user ID.
- Comment.php: This class focuses on the comments posted on various pages. It includes attributes such as positionID and postedOn to keep track of the comment's location and date. This allows for easier organization of comments.
- Date.php: This class allows for the creation of timestamps to go with comments, pages or anything else that may need a timestamp. It has day, month, year, hour, minute and second attributes and can assemble all of these into a timestamp string for easier use.
- Favourite.php: This class allows for the function that users can use to favorite a card game. It keep track of user IDs and entries to list all the favorites a user has.
- Rating.php: This is what stores the ratings on the games on the wiki. It has getters and setters as well as functions to get the various ratings based on user ID and entry ID.
- User.php: This is the main user classes. It builds and works with objects of the user type. This involves storing information such as user ID, password and username. This class also has functions to store users in the database and confirm a user's username and password when they sign in.

- userType.php: This is the class that organizes users into the different types in the system. These types include user and admin, and each is associated with different access levels. This class ensures that users have access to all they need and that regular users cannot access admin features.
- Wikientry.php: This is the class that keeps track of and structures the entries on the wiki. These include the game pages, and the attributes of this class include the name, ratings and date last edited. This class is crucial to manage all the information regarding a page on the website.

View:

- View_function.php: This file contains the functions that are used by the pages to generate the content that the user sees, and can interact with.

Controller:

- connetctDB.php: Used to initialize a connect to the database
- Controller_functions.php: This file has many helper functions that query the database.
- Footer.php: Used to create the footer for the bottom of the page.
- Functions.php: This is where the database class is found. It's located in the controller module and allows the model to connect to the database to retrieve different items. This class initializes and closes database connections.
- Header.php: Used to display the header on the page.
- Left-menu.php: Used to display the left menu on the page.
- Right-menu.php: Used to display the right menu on the page.
- JavaScript Libraries:
    - Jquery-3.2.1.min.js
- CSS: Used for the themes of the website
    - Theme.css
    - https://use.fontawesome.com/releases/v5.0.13/css/all.css : External CSS Library for Awesome Fonts used in the star rating system.