

GAMEON GAME HUB

SOFTWARE REQUIREMENTS & SPECIFICATIONS

Group 4

Zainab Akubat

Richard Hall

Muhammad Mozzam Khan

CONTENTS

Contents	2
Introduction	5
Overall description	7
Product perspective	7
Product functions	7
User characteristics	8
Constraints	8
Assumptions and dependencies	9
FUNCTIONAL Requirements	10
GAMES SEARCH/ RESULTS: USER REQUIREMENTS	10
GAME SEARCH/ RESULTS	10
Game Search: SEARCH GAMES	10
Game Search: CONFIGURE TAGS	11
Game Search: SET RESULTS RETURNED	11
Games Catalogue: User Requirements	12
1. View Catalog	12
2. View Game Details	13
3. Configure Catalog	13
News Articles: USER REQUIREMENTS	14
News Articles	14
Browse ARTicles	14
Post Articles	15
Edit Articles	15
Define Articles	15
Reviews: User Requirements	16

View Game Review	16
Post Game Review	16
Organize Reviews.....	17
Newsletter: USER REQUIREMENTS.....	19
Newsletter: register	19
Newsletter: Create newsletter	20
Newsletter: edit newsletter	20
Newsletter: PREVIEW NEWSLETTER	20
Support: USER REQUIREMENTS.....	22
Support: Post Questions	22
Support: Post Comments on Others questions	23
Support: Be Notified of Updates	23
Support: Respond to Questions.....	23
Support: Update Questions	23
Support: View Question Details.....	24
Support: Respond to Comments	24
System Requirements	25
Games Search: SEARCH GAMES/RESULTS	25
Games Search: CONFIGURE TABS	26
Games Search: SET NUMBER OF RESULTS RETURNED	27
Module – Games Catalog.....	28
Games Catalog: Configure / Edit Details.....	28
Games Catalog: View Details	28
Games Catalog: View Catalog	29
Module – News Articles.....	29
News Artilces: Browse Articles	30
News Artilces: Post Articles	30

News Artilces: Edit Articles	31
News Artilces: Define Articles.....	31
Module – Reviews	32
Reviews: View Review	32
Reviews: Post Game Review	32
Reviews: Organize Reviews	33
Module – Create Newsletter	34
register	34
Create newsletter	35
EDIT NEWSLETTER	35
PREVIEW NEWSLETTER	36
Module - Support.....	37
Support: Post Questions	37
Support: Post Comments On Other Questions.....	37
Support: Be Notified of Updates	38
Support: Respond To Questions	38
Support: Update Questions	39
Support: View Question Details.....	40
Support: Respond To Comments.....	41
SYSTEM MODELS	42
Module - Game Search:	42
Game Search: SEARCH GAMES	42
Game Search: CONFIGURE TAGS	44
Game Search: SET RESULTS RETURNED.....	46
Module - Game Catalog:.....	48
Game Catalog: View Details.....	48
Game catalog: View Catalog.....	50

Game Catalog: Configure Catalog.....	52
Module – News Articles.....	54
News Articles: Browse Articles	54
News Articles: Post Articles	56
News Articles: Edit Articles	58
Module – Reviews	60
Reviews: Post Reviews.....	60
Reviews: View Reviews.....	62
Reviews: Configure Reviews	64
Module - Newsletter.....	66
Newsletter: rEGISTER.....	66
Newsletter: create newsletter.....	68
Newsletter: edit newsletter.....	70
Newsletter: PREVIEW NEWSLETTER	72
Module - Support.....	74
Support: Post Questions	74
Support:Post Comments on Other’s Questions.....	76
Support: Be Notified of Updates	78
Support: Respond to Questions.....	80
Support: Update Questions	82
Support: View Question Details.....	84
Respond to Comments	86
Domain Analysis	88
NON-functional requirements	89

In today's fast paced world of game development, users find that they need a trusted, one stop shop for all of their gaming needs. Through a series of focus groups, we have determined that those needs include game catalogs, reviews, support, news, a searchable games database, and a newsletter that provides timed games news, formatted for the "gamer on the go" with pertinent game highlights. To provide a source for these needs, we have developed "GameOn."

Guidelines:

- **Purpose:**

- *Currently, a gamer has no centralized location to get all of their gaming needs.*
- *Our software will provide a "one stop shop." Gamers will no longer need to roam the internet hoping to find the relevant information they require for their game of choice.*
- *Our objective is to offer competition to sites such as Steam and Origin. We plan on optimizing our loading times to less than one second per page load. We plan on accomplishing this goal by providing simple, yet flamboyant layouts, based on PHP, that quickly inform our customers of exactly what they need to know, whenever they need to know it.*

- **Scope:**

- *Our site will generate a custom web experience based on the database management software PHP. A user will request information that our database will return, tailored to the user.*
- *Our software will allow users to search for games, read reviews, sort games by popular headings such as ratings, genre, platform, top selling, and top fan favorites.*
- *Our software will provide a closed community of gamers without the fear of flaming or spam. This will be moderated by our community log in membership requirement.*
- *A benefit of our software is that we strive to build a community of gamers who share in the gaming experience, building a network of trusted peers within the gaming community.*

- **Definitions, acronyms and abbreviations:**

Admin: *An elevated system user who has extended privileges including viewing question metadata, creating other admins, answering forum questions, adding definitions to games in the database, and editing customer questions in the forum.*

Customer: *any individual who registers on "GameOn" is considered a customer.*

Database: *our continuously updated reservoir of knowledge, in digital form, that a user may access.*

Gamer: *an all-inclusive term that simply means any individual interested in learning more about popular video games.*

PHP: *is a server-side scripting language designed for web development but also used as a general-purpose programming language.*

References

IEEE. (2015 , October 12). <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=278253>. Retrieved from IEEE.org: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=278253>

IEEE. (2015, October 12). ieeexplore.ieee.org. Retrieved from <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=392555>:

- **Overview:**

The following information regarding the software proposal serves only as an overview. For more detailed information, please contact us directly.

OVERALL DESCRIPTION

- a) *Product perspective;*
- b) *Product functions;*
- c) *User characteristics;*
- d) *Constraints;*
- e) *Assumptions and dependencies;*

PRODUCT PERSPECTIVE

User interfaces: A user of GameOn will access the site via the web. Upon reaching the site, a simple log in with a name and password will provide access to a wealth of gaming knowledge. A user will not be excluded on race or sexuality. All users will feel welcomed. A dynamic web experience will await the user as each web page is individually constructed based on PHP per the user's wishes.

PRODUCT FUNCTIONS

Our product will be a web based community of like-minded gamers who converge to discuss and review current games. An internet connection and device capable of web browsing are the only two software hardware requirements.

USER CHARACTERISTICS

A typical user of GameOn could be just about anyone. Our wide product base will be a key pillar of success upon which our business plan will excel. No user will ever be made to feel excluded on GameOn. The only prerequisite is an internet connection and a desire to learn more about gaming.

CONSTRAINTS

This subsection of the SRS should provide a general description of any other items that will limit the developer's options.

These include:

- a) Regulatory policies: Our site will be governed by the current laws of the United States. All profanity and decency laws will be observed.*
- b) Hardware limitations: Our site will be governed by the speed of the internet connection of the user.*
- c) Interfaces to other applications: Our interface will be any web browser.*
- d) Parallel operation: a user is allowed a one to one connection with the server.*
- e) Audit functions: our system will be auditable through a review of our system logs.*
- f) Control functions: our site will govern input and output through the use of a credential verification.*
- g) Higher-order language requirements: Our site will be built on a combination of HTTP and PHP.*
- h) Signal handshake protocols: Our site will utilize the TCP protocol.*
- i) Reliability requirements: GameOn will strive to attain a 99.9% reliability rate.*
- j) Criticality of the application: Our application will be hosted in the cloud. Our cloud servers contain redundant power supplies as well as redundant data, therefore our application will reside at level 3.*

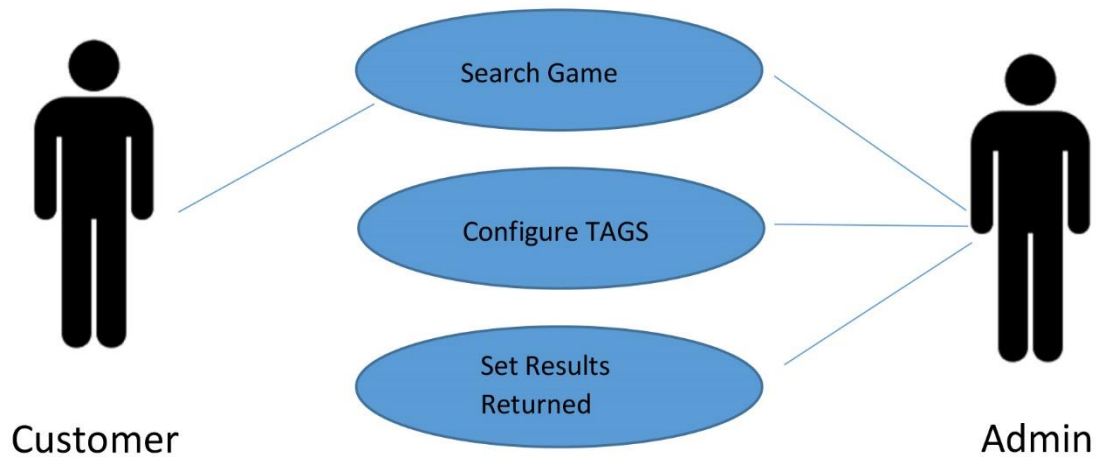
- k) *Safety and security considerations: All users must log in and create an account. An admin will monitor user activity. In the event of a customer creating an unsafe environment, that user will be blocked from the site.*

ASSUMPTIONS AND DEPENDENCIES

At GameOn, we assume that the user will have an internet connection paired with a suitable interface for accessing the online database. We are also assuming that a user will have a general knowledge of the gaming environment and be familiar with one or more gaming systems. Our site is dependent on an internet connection.

FUNCTIONAL REQUIREMENTS

GAMES SEARCH/ RESULTS: USER REQUIREMENTS



GAME SEARCH/ RESULTS

Any user or customer will have the opportunity to search a game on the “Game On” website. The user or customer will have the option of searching for the game based on the name of the game or the genre of the game. User/customer will type in words or partial words and the search database will show the listing of the games that are matching to the letters the customer has put in the search bar.

The “Game Search” use case contains the following basic steps:

1. Customer will search for a game based on the name or genre
2. GameOn will search it’s database and return the results
3. User may search with words or partial words.

GAME SEARCH: SEARCH GAMES

Any user should be able to search any game based on the name of the game or based on the genre of the game. The user will be able to type in words or partial words of the game or the genre of the game and the system should display results based on the input the user has typed. If the user types in a game or a genre of a game that is not in the database, the system will return “not found”.

The “search game” use case contains the following basic steps:

1. Game On user will enter name or genre of game in the search bar

2. Game On system will search the database
3. If game or genre is found in the database the system will display the information
4. If game or genre is not found in the database the system will display “not found”.

GAME SEARCH: CONFIGURE TAGS

The admin will configure tags in two ways. One will be based on the name of the game, and the other will be based on the genre of the game. If the user enters the name of the genre it will only search games within that genre. If the user enters partial words that are a name of a game it will display names of the games only. The admin will configure tags and save them in the “Game On” database.

The “configure tags” use case contains the following basic steps:

1. Admin will create tags for the games or game genres
2. Admin will submit tags
3. If the admin correctly submits tags the system will create the tag and save it inside the database of the website
4. If the admin fails to correctly submit tags, then the system will notify the admin and the admin will have to make correction and begin from the top

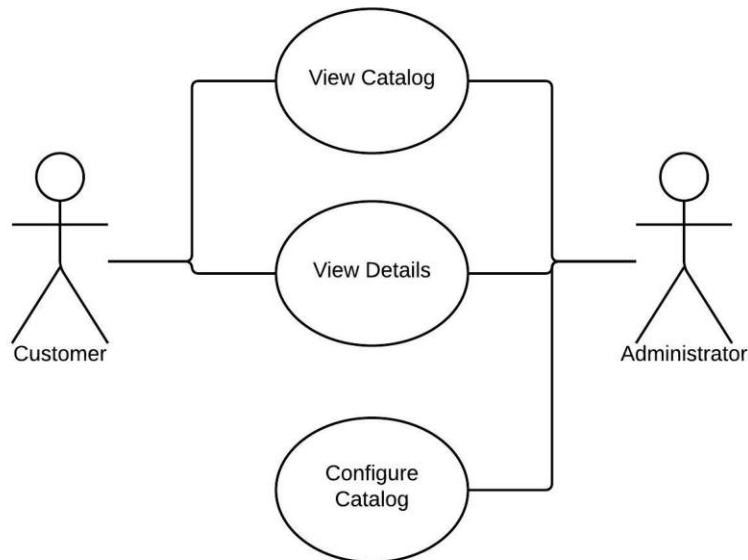
GAME SEARCH: SET RESULTS RETURNED

The customer will search for a game or a genre of a game in the search bar and the number of matching results will be displayed. The customer will be typing words or partial words to the name of the game or the genre of the game. Based on the matching letters the amount of results should be displayed. If the user types a name of the game or a genre of the game and the system doesn’t recognize it, the system will return, “0 results found”.

The “set results returned” use case contains the following basic steps:

1. The user will enter the name or genre of a game
2. The system will search for it in the database
3. If the game or genre of game is found in the database, the system will return the amount of results found
4. If the game or genre of game is not found in the database, the system will return, “0 results found”.

GAMES CATALOGUE: USER REQUIREMENTS



1. VIEW CATALOG

A registered user will be able to head to the catalog section and view the games list provided by GameOn. The user will be able to type in the name of the game they wish to see in the search bar. If they are simply just exploring the catalog, they will have the option to sort the games by different attributes such as price, platform, genre, name and ratings.

1. Click on the 'Games Catalog' section on the website.
2. View the 'featured' games list provided by the website.
3. If the user wishes to search for a game or explore the games, they can use the search bar and use filters to sort the games by attributes.
4. Once the search query is made, the list of games will show up.

2. VIEW GAME DETAILS

A registered user will be able to search for any game and they will have the option to click on the game and view the details of the game such as a description, price, reviews, ratings, platform and other details about the game. This provides the user with a detailed scope about the game they are viewing and helps them decide if they should invest in buying the game or not.

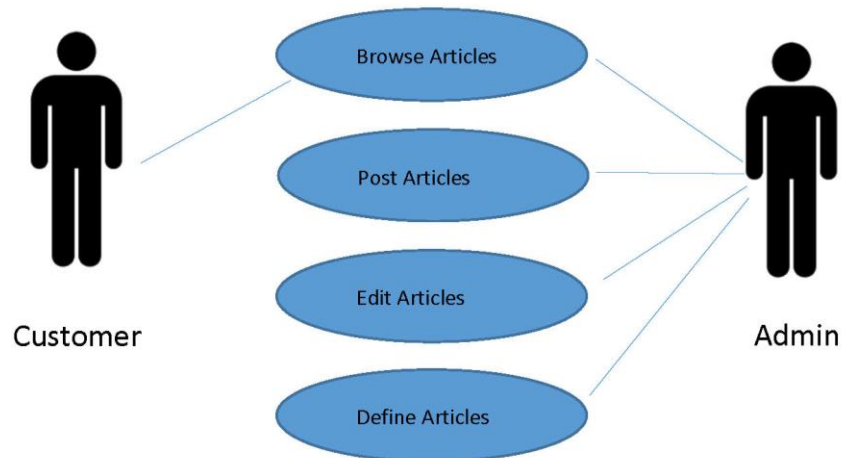
1. The registered user can search for a game and the list shows up.
2. Once the list appears, the user can select a game.
3. The game details open up and the user is able to view all the information regarding the game.

3. CONFIGURE CATALOG

The catalog has a collection of all games and the Administrator has the power to add or remove games from the catalog. The admin can also edit current games and change anything in the game details such as the genre, platform or description of the game. This allows the users to have access to the latest games that the administrator adds.

1. View the catalog
2. Administrator can select the option to 'Configure Catalog'
3. Once selected, the admin now has options to add or remove or edit current games

NEWS ARTICLES: USER REQUIREMENTS



NEWS ARTICLES

Any customer should be able to browse the news articles. News articles are listed in order of recent articles. An admin may create and post news articles. An admin may edit existing news articles. An admin has the ability to add tags to an article in order to define the keywords of an article. Customers cannot register another user as an admin. Customers may not post, define, or edit articles.

The “news articles” use case contains the following basic steps:

1. Customer requests access to the “News” section of the site.
2. GameOn site hosts responds with a dynamic listing of current news articles.
3. User may sort or search articles by date or defined key words.

BROWSE ARTICLES

Any customer or admin should be able to browse news articles. After successfully navigating to the “News” area of GameOn, the user will be presented with a list of current articles. Both customers and administrators should use the same News browse screen.

The “Browse” use case contains the following basic steps:

1. User navigates to news section
2. Database is queried and pertinent news articles are returned
3. If user has added a search term, the database will be queried, returning news articles tagged with matching definitions

POST ARTICLES

Only an admin may post an article into the news section. Upon completion of authorship, an admin may upload a new article into the news database. This document is then added to the available documents that a customer may access.

The “post articles” use case contains the following basic steps:

1. An admin level user writes a topical news article
2. The admin uploads the document to the news database

EDIT ARTICLES

Only an admin may edit an article in the news section. Edits may include grammar, spelling, tag definition, and deletion of news article files.

The “edit articles” use case contains the following basic steps:

1. An admin level user selects an article
2. The desired changes are completed
3. The amended document is released back into the document database

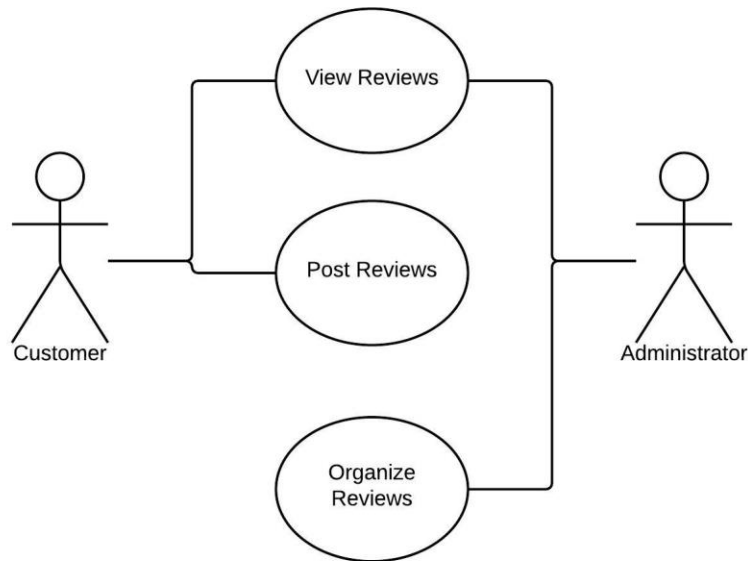
DEFINE ARTICLES

Only an admin may define an article in the news section. An admin may tag an article with relevant key words such to help customers improve their news feed results. Tag definitions such as “OSi,” “First person shooter,” “FPS,” and “Xbox” would be a few examples. The amount of definitions will be a constantly evolving and increasing amount as future needs dictate new tags.

The “tag articles” use case contains the following basic steps:

1. An admin level user selects a news article
2. The admin adds the relative tag definitions to the article and uploads the document to the news database
3. The article is now searchable with the included tags as search parameters.

REVIEWS: USER REQUIREMENTS



VIEW GAME REVIEW

The registered user whether it's a customer or administrator, can view the reviews posted by other users. The user will be able to select from a list of different games and will be able to see the games reviews which will also have a rating system to see how many stars it has received (ratings will be from 1-5 stars).

1. The user has to search for a game.
2. Once search results show up, click on a game.
3. The user will be presented with all the game descriptions and information about the game including the reviews.
4. The user will now be able to read the review of the game posted by other users.

POST GAME REVIEW

The registered user (customer) will be able to view the reviews posted by other users and will also have the power to post their own review. They can choose whichever game they would like to review and can select it and proceed with making a post about it. While reviewing, they can also select from the range of 1-5 stars to rate the game. The post review option will present the user with a title bar and a description box to fill in for the review. Once the user fills in the review and hits submit, the review will then be held in a 'pending' state until the administrator approves or rejects the review.

1. User will have to search for a game.
2. Once the results show up, click on a game.
3. The user can now view all the information about the game including the views.
4. Select 'Post a review' in the reviews section.
5. Fill in the title and description boxes and select the ratings for the game from 1-5 stars.
6. Click 'Submit' and the review will be posted upon administrator's decision of approval.

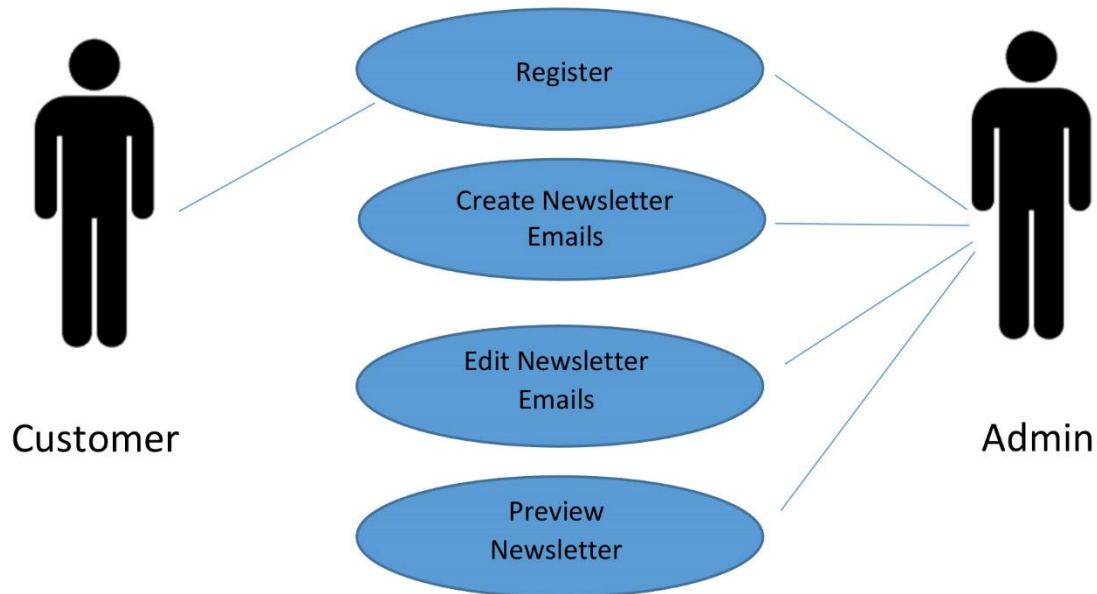
ORGANIZE REVIEWS

The administrator will have the power to view the reviews posted by the users and before they are posted, they must go through the administrator. The Admin will have the right to accept or reject any 'pending' reviews that a user posted. Once the admin selects approve, the review will be posted on the website viewable by the customers. If the admin rejects the pending review, the review will be deleted.

1. Under the reviews section, the admin will be able to view all the reviews.
2. There will be a section for 'pending reviews' which the admin can select.

3. Once in the 'pending reviews' page, the admin will see reviews posted by users.
4. Admin can accept or reject the reviews.
5. If accepted, the reviews get posted.
6. If rejected, the reviews get deleted.

NEWSLETTER: USER REQUIREMENTS



NEWSLETTER: REGISTER

The customer will be asked to register to receive newsletter emails from the website. The customer will be asked to give their first name, last name, and email address. Once the customer puts all the required information for receiving emails the system will validate it. If the information is correct and it is successful, the system will return success and display a message saying, "Thank You". If the system does not validate the given information the system will be failed and will notify the user. The user will correct the errors and start from the beginning.

The "register" use case contains the following basic steps:

1. The user will submit the required data
2. The system validates it
3. If the validation is successful, the system will return success and display a message saying, "Thank You"
4. If the validation has failed, the system will notify the user and the user will fix the errors and resubmit the required data

NEWSLETTER: CREATE NEWSLETTER

The admin will create a newsletter and submit it. If the submission was successful the admin will be able to send out emails. Once the admin sends out emails, both the admin and the user will be notified. If the submission has failed, the system will notify the admin, and the admin will fix the errors and resubmit it.

The “create newsletter emails” use case contains the following basic steps:

1. Admin will create Newsletter
2. The admin will submit the newsletter
3. If the submission was successful, the admin will send out emails
4. If the submission has failed, the system will notify the admin, and the admin will fix the errors and resubmit the newsletter.
5. The admin and user will be notified of the submission

NEWSLETTER: EDIT NEWSLETTER

The admin will be able to go into the Newsletter that they want to edit and make the changes needed. The admin will submit the Newsletter once the changes are made. If the submission was successful, the system will place the original Newsletter with the new Newsletter. The admin will then be able to send out emails to registered users. If the submission failed, the system will notify the admin, the admin will make necessary changes and resubmit the newsletter.

The “edit newsletter” use case contains the following basic steps:

1. Admin will go to the Newsletter that they want to edit, and will make changes to it
2. Admin will submit the Newsletter
3. If the submission was successful, the system will place the original Newsletter with the new Newsletter
4. If the submission has failed, the system will notify the admin, the admin will make appropriate changes and resubmit the Newsletter
5. Once the submission is complete, the admin will send emails to registered users

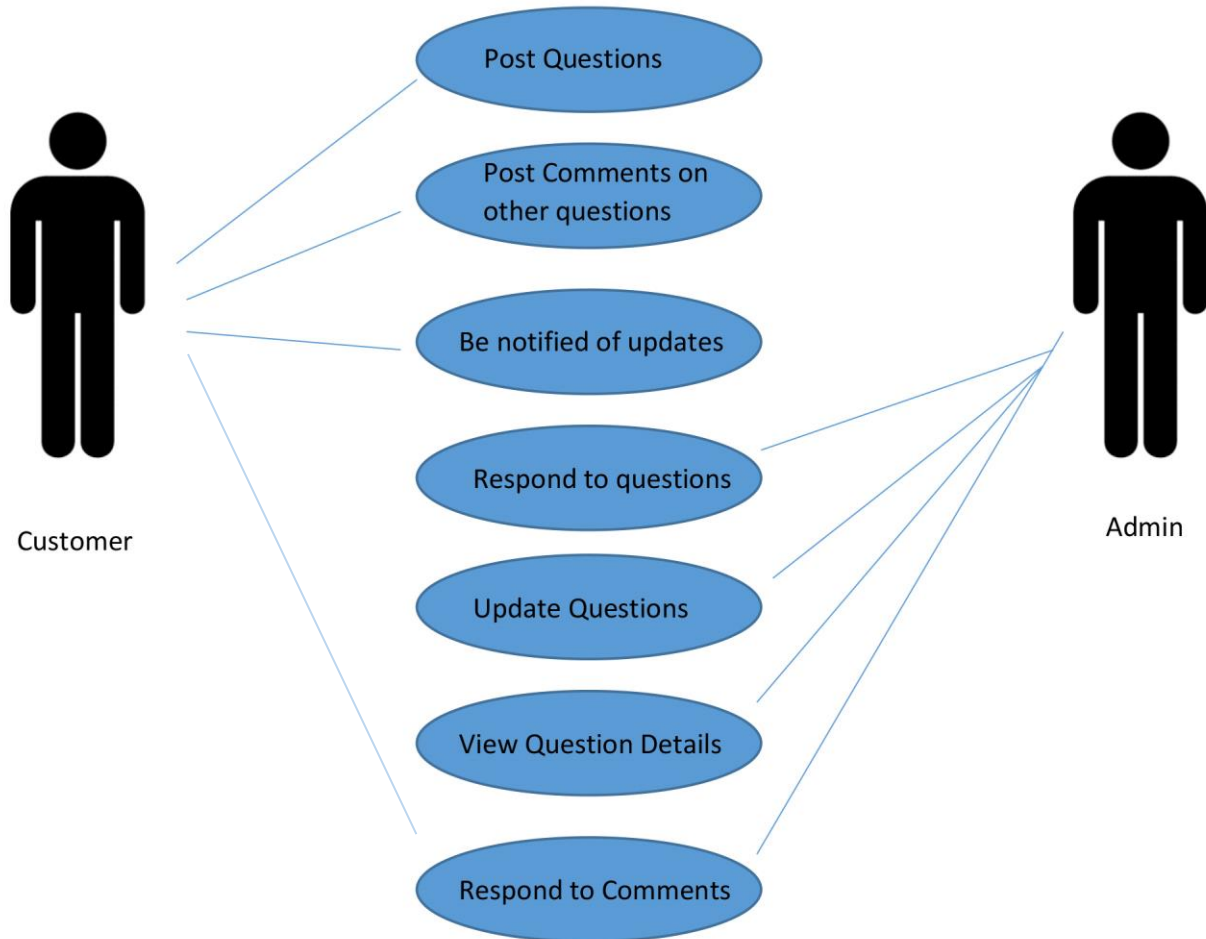
NEWSLETTER: PREVIEW NEWSLETTER

The admin will create a Newsletter and submit it. If the submission was successful, the admin will be able to preview the newsletter as a final product. If the submission failed, the system will notify the admin, the admin will make appropriate changes and resubmit the Newsletter. Once the submission is complete, the admin will sent out emails and both the user and admin will be notified.

The “preview newsletter” use case contains the following basic steps:

1. The admin will create the Newsletter
2. Admin will submit the Newsletter
3. If the submission was successful, the admin will be able to preview the final product of Newsletter
4. If the submission failed, the system will notify the admin, admin will make corrections and resubmit the Newsletter
5. Once submission is complete, admin will sent out emails and both user and admin will be notified

SUPPORT: USER REQUIREMENTS



SUPPORT: POST QUESTIONS

Any customer should be able to browse the news articles. Any customer may post a new question. Any customer may also post comments onto other customer's questions. All customers who post a question or comment should be notified when the question has been updated. An admin may respond to questions. An admin may also update any posted questions. An admin may view the question details or metadata. An admin may also respond to comments.

The "support" use case contains the following basic steps:

1. Customer requests access to the "Support" section of the site.
2. GameOn site hosts responds with a dynamic listing of current support topics.
3. User may sort or search questions by date or defined key words.

SUPPORT: POST COMMENTS ON OTHERS QUESTIONS

Any customer or admin should be able to post a comment to another person's question.. After successfully navigating to the "News" area of GameOn, the user will be presented with a list of current articles. Both customers and administrators should use the same News browse screen.

The "Post Comments on Others Questions" use case contains the following basic steps:

1. Customer navigates to support
2. Database is queried and pertinent support articles are returned
3. If user has added a search query, the database will be searched, returning relevant support questions.

SUPPORT: BE NOTIFIED OF UPDATES

When a question has received an update in the form of an edit or comment response, a customer will be notified.

The "Be Notified of Updates" use case contains the following basic steps:

1. An admin or customer composes a comment linked to a customer question.
2. A message is generated by the system to notify the customer of the new activity.

SUPPORT: RESPOND TO QUESTIONS

In GameOn, a customer may ask a question in support directly to an admin. Only an admin may therefore answer a question.

The "Respond to Questions" use case contains the following basic steps:

1. A customer asks a question directed towards the admin staff.
2. The admin staff will attempt to answer the customer's question to the best of their ability.
3. A message is sent by the system to notify the customer.

SUPPORT: UPDATE QUESTIONS

Only an admin may update questions in the support section. Edits may include grammar, spelling, tag definition, and deletion of support questions.

The “Update Questions” use case contains the following basic steps:

1. An admin level user selects a question.
2. The desired changes are completed.
3. The amended question is released back into the database.

SUPPORT: VIEW QUESTION DETAILS

Only an admin may see the metadata regarding questions. Metadata would include the customer’s real name and contact information, the customer’s user id, and other defining information.

The “View Question Details” use case contains the following basic steps:

1. An admin level user selects a support question
2. The admin may now read through all attached question metadata.

SUPPORT: RESPOND TO COMMENTS

An admin or a customer may respond to comments. Responding to comments creates a fluid dialogue between interested parties, enriching the experience of all users.

The “Respond to Comments” use case contains the following basic steps:

1. A customer or admin reads a comment that elicits a response.
2. The other part then posts a reply in order to continue the discussion.
3. The other parties in the comment thread are notified via message of new comment activity.

SYSTEM REQUIREMENTS

GAMES SEARCH: SEARCH GAMES/RESULTS

The following table outlines the details of use case # 1: Search Games

Use Case ID:1	Search Games
Description	<p>Customers will search for a game by, the game name or the game genre and the search should understand that the customer may type partial words, it may not be an exact match.</p> <p>The search will then display the game information that are in match with the customer search.</p>
Input	<ul style="list-style-type: none">• Name of Game• Genre of Game
Output	<p>If transaction was successful, then the system will display the information of games. If the transaction failed, the system will return the message "No match".</p>
Preconditions	<p>The database of games should be connected to the search bar in order to correctly display the information the customer would be asking for.</p>
Post Conditions	<p>If succeeded, the results of the games searched will be displayed.</p>
Frequency of Use	<p>Many time a day</p>

GAMES SEARCH: CONFIGURE TABS

The following table outlines the details of use case # 2: Configure Tabs

Use Case ID: 2	Configure "TAGS"
Description	<p>The administrator will configure tags in two parts. The first will have tags for genres and inside genres will be names of games.</p> <p>When the customer searches based on the genre it will search that particular genre only, but if the customer searches by the game name then it will search the entire database of games.</p>
Input	<ul style="list-style-type: none">• Games based on Genre• Name of Games
Output	If the search matches the tags, the results will be displayed. If the search does not match or fails, the admin will display, "not found".
Preconditions	The Email/Username does not exists within the system at the time the user submits the form.
Post Conditions	The admin will configure tags before any customer is able to search for games.
Frequency of Use	Once, when making the database of genres and games.

GAMES SEARCH: SET NUMBER OF RESULTS RETURNED

The following table outlines the “set number of results returned” use case # 3:

Use Case ID: 3	Set the number of Results Returned
Description	<p>The customer will search for a game or genre of a game in the search bar and the number of matching results will appear on the screen.</p> <p>The customer will be typing in partial words or words that are related to their search.</p>
Input	<ul style="list-style-type: none">• Genre of Game• Name of Game
Output	<p>If the search was successful and there was a match to the genre or games the number of matches or results will be returned.</p> <p>If the search failed or did not have a match to the database, the admin will display, “0 results found”.</p>
Preconditions	The admin should be able to recognize words or partial words of the game or genre.
Post Conditions	<p>If there is a match, the system will display the number of results found.</p> <p>If the match is not found, the system will display, “0 results found”.</p>
Frequency of Use	<ul style="list-style-type: none">• Many times in day

MODULE – GAMES CATALOG

GAMES CATALOG: CONFIGURE / EDIT DETAILS

Use Case ID: 1	Configure/Edit Catalog
Description	As the customers get to view and browse the game catalogue, the administrator is the one who sets up the catalog and the games in the catalog. The administrator must be logged in at first and can add or remove games from the catalog. Admin must also give each game the correct label for the attributes such as genre, price, platform, etc.
Input	Open catalogue and select 'Edit Catalog'
Username/Password	Add or remove games from the catalog
Output	Updated 'Games Catalog' page with the latest added or removed games.
Preconditions	Must be logged into the system as an admin to edit the catalog.
Post Conditions	The admin must save changes for the changes to appear. The updated catalog will be visible to all users.
Frequency of use	Whenever there is need for a game to be added or removed. Usually every 1 to 2 weeks.

GAMES CATALOG: VIEW DETAILS

Use Case ID: 1	View Details
Description	The user will be able to view any details such as descriptions and ratings and reviews and many other things about the game they select to view. The details of the game help the user to determine what the game actually is like, they can read the storyline in description. They can also know what platform it is on and see how the other users have rated and reviewed the game.

Input	<ul style="list-style-type: none"> • Username/Password • Search query for a game they wish to see details about
Output	Details page of the selected game appears to the user.
Preconditions	The user must be logged in and must search for a game in order to see the details.
Post Conditions	The users will be able to see the details about the game.
Action	The system checks if there is a registered user logged in or not, if yes then the user will be allowed to search for a game and read the details of the game.

GAMES CATALOG: VIEW CATALOG

Use Case ID: 1	View Catalog
Description	The registered user will be able to view any game that is listed in the catalog. They are allowed to browse the game lists and they can sort the games by their attributes. The attributes such as platform, genre, ratings, price and reviews. This helps the user categorize different games into something they prefer to see.
Input	<ul style="list-style-type: none"> • Username/Password • Search tag for a game • Sort by attributes
Output	List of games that are in the catalog. May be sorted according to filters applied by user.
Preconditions	The user must be logged in and must search for the games in order to see the results list. Otherwise, a featured default list of games will appear.
Post Conditions	The users will be able to see the list of all the games that are under the filter applied during search.
Action	The system checks if there is a registered user logged in or not, if yes then the user will be allowed to search for a game and view the list of all games.

MODULE – NEWS ARTICLES

NEWS ARTILCES: BROWSE ARTICLES

Use Case ID:1	Browse Articles
Description	There are two types of users who can be logged into the GameOn site, customers and admins. Customers and admins may browse the online news article database.
Input	Any desired search definitions or blank for newest news items
Output	If transaction was successful, the system will return a dynamically populated web page based upon the user input. If the transaction failed, the system will return the appropriate error message
Preconditions	A user must be logged in as either a customer or an admin to use the news article system.
Post Conditions	The article will be supplied to the customer.
Action	<p>The system will perform the following checks before allowing the news article list to be dynamically generated</p> <ul style="list-style-type: none"> • User is logged in to the system as either an admin or a customer • Check if any requested tags have been amended to the search request

NEWS ARTILCES: POST ARTICLES

Use Case ID:2	Post Articles
Description	<p>An administrator will successfully log into the GameOn site.</p> <p>The admin will upload a new news article to the searchable database.</p>
Input	Admin uploads a new news article to the GameOn site.
Output	Will receive a confirmation page that the news article has been posted.
Pre-conditions	User must be elevated to admin status.
Post Conditions	News article is updated with the new news article.

Action	An admin uploads a new news article to the news article database. The system will return a message stating the news article upload has been successful.
--------	---

NEWS ARTILCES: EDIT ARTICLES

Use Case ID:3	Edit Articles
Description	An admin determines an edit is required on a published news article. The admin is given the option of updating the article with revised information.
Input	Revised information designed to replace incorrect or missing previously posted information
Output	If the transaction succeeds, the system will post the updated article on the site, and the system will generate an “update successful” message. If the edit fails, the system will send back an “edit unsuccessful” message.
Pre-conditions	The user is registered as an admin level user.
Post Conditions	The submitted edited news article is stored in the database. An edit successful message is displayed.
Actions	The system replaces the previous news article entry with the revised news article entry.

NEWS ARTILCES: DEFINE ARTICLES

Use Case ID:3	Define Articles
Description	An admin determines an article fits a general tag and a definition is added to the article metadata, searchable in subsequent searches.
Input	Relevant keywords that help define an article.
Output	If the transaction succeeds, the system will update the database with the additional article metadata..

Pre-conditions	The user is registered as an admin level user.
Post Conditions	The submitted edited news article is stored in the database with the additional articles included. An edit successful message is displayed.
Actions	The system replaces the previous news article entry with the revised news article entry containing the additional metadata.

MODULE – REVIEWS

REVIEWS: VIEW REVIEW

Use Case ID: 1	View Review
Description	The user will be able to view any feedback or review left by other customers reading the performance and how they feel about the game. The games are evaluated using the reviews and an additional feature which is a rating out of 5 stars. The reviews will be crucial to the customer because they want to know how the game is before getting it.
Input	<ul style="list-style-type: none"> • Username • Password • Search query for the game
Output	The user once they search for a game and click on it. They will be able to view the reviews of each game and the ratings.
Preconditions	The customer should have a username and must be registered and logged in within the system to view and post reviews.
Post Conditions	The user is presented with a page showing the reviews of the game.
Action	The system checks if user is logged in or not, if yes then they can proceed to searching for a game and viewing the reviews of games.

REVIEWS: POST GAME REVIEW

Use Case ID: 1	Post Game Review
Description	The registered user (customer) will be able to view the reviews posted by other users and will also have the power to post their own review. They must be logged in with their username/password. The post review option provides them to post a game review for any game they would like to review. They will also be allowed to post a rating in which they get to pick out of 5 stars. Once the user fills in the review and hits submit, the review will then be held in a 'pending' state until the administrator approves or rejects the review.
Input	<ul style="list-style-type: none"> • Search query for the game • Title of review • Description of review • Ratings out of 5 stars
Output	The user will be able to see their own posted review of a particular game after it has been approved by the admin.
Preconditions	The customer should have a username and must be registered and logged in within the system to view and post reviews.
Post Conditions	The user is presented with a page showing the reviews of the game including their own. It will be visible to any registered user.
Action	The system checks if user is logged in or not, if yes then they can proceed to post a review on the game that they choose.

REVIEWS: ORGANIZE REVIEWS

Use Case ID: 1	Organize Reviews
Description	The administrator must approve or reject reviews that are posted by users. They can do that by reading the user reviews in the 'pending' list. If the user reviews are appropriate, the admin can approve them. If they are inappropriate, the admin can reject and delete the pending review.
Input	Approve or reject reviews.
Output	Updated 'Reviews' page with the latest added reviews.

Preconditions	The administrator must be logged in and should be able to read all the preexisting reviews and the 'pending' reviews.
Post Conditions	The users will be able to see the updated list of reviews that were previously there and the new reviews that were approved by the administrator.
Action	The system checks if there is an administrator logged in or not, if yes then the admin will be allowed to approve or reject the reviews.

MODULE – CREATE NEWSLETTER

REGISTER

Use Case ID:	Register
Description	The customer will register to receive newsletter from the website.
Input	<ul style="list-style-type: none"> • First Name • Last Name • Email
Output	If transaction was successful, then the system will return a Thank You message back to the user. If the transaction failed, the system will notify the user and return the appropriate message.
	The Email does not exists within the system at the time the user submits the form.
	If succeeded, the user will be registered to receive newsletter emails from the website.
	Many times a day

CREATE NEWSLETTER

Use Case ID: 2	Create Newsletter Emails
Description	The admin will create newsletters that will be sent off to people who are registered to receive newsletter emails.
Input	Email
Output	If the user has registered to receive emails for newsletters they will be receiving newsletters.
Preconditions	The admin must have access to creating newsletter emails as well as be able to see who is registered to receive newsletter emails.
Post Conditions	If the email was in the system, the user will receive emails from the website, if the user was not registered they will not receive emails.
Frequency of Use	Based on the admin, how many newsletter emails they send out.

EDIT NEWSLETTER

Use Case ID: 3	Edit Newsletter
Description	The admin will go to the newsletter email and make the changes needed.
Input	<ul style="list-style-type: none"> • First Name • Last Name • Email
Output	If the change was successful the admin will be notified. If the system fails, the admin will be notified to make changes, thereafter will be able to resend the emails to each register user.
Preconditions	The admin must has access to editing newsletter emails.

Post Conditions	If succeeded, the users who are registered will receive the edited version of the newsletter.
Frequency of Use	Based on the admin, how many newsletter emails they send out.

PREVIEW NEWSLETTER

Use Case ID: 4	
Description	Preview Newsletter
Input	The admin will create newsletter emails to send to people who are registered to receive them. Before the admin sends the emails out to the users, the admin will be able to preview the final product.
Output	<ul style="list-style-type: none"> Email
Preconditions	If the admin submits the newsletter correctly, the admin will be able to view the final product before sending out the emails.
Post Conditions	The Admin must have access to creating and previewing newsletter emails.
Frequency of Use	If succeeded, the admin will be able to view the final product of the newsletter before sending it out to registered users.
	Based on the admin, how may newsletter emails they send out.

MODULE - SUPPORT

SUPPORT: POST QUESTIONS

The following table outlines the details of use case # 1: Post Questions

Use Case ID:1	Post Questions
Description	As gamers encounter areas of perplexion, a question may be posted to gain insight from the GameOn community.
Input	Any question(s) related to gaming or game related activities
Output	When a question is answered, a message is triggered to notify the asker of an update.
Preconditions	A user must be logged in to post a question.
Post Conditions	Question becomes viewable on the website.
Action	The system will check that the User is currently logged in with valid credentials.

SUPPORT: POST COMMENTS ON OTHER QUESTIONS

The following table outlines the “Post Comments on Other’s Questions” use case.

Use Case ID:3	Post Comments on Other’s Questions
Description	As a GameOn user browses the support forum, a question is encountered that elicits a comment response interaction.
Input	Either a customer or an admin responds to a question with a comment readable by all other users.
Output	Upon successful update, the comment will become live on the site, visible by all users browsing the support area.

Pre-conditions	The user is registered with valid credentials.
Post Conditions	The comment is stored in the database.
Actions	The system will perform a valid user check before proceeding.

SUPPORT: BE NOTIFIED OF UPDATES

The following table outlines the “Be Notified of Updates” use case.

Use Case ID:3	Be Notified of Updates
Description	After someone has responded to a customer’s question, a notification will be generated and sent to the customer, alerting the customer to the new activity.
Input	A GameOn user inputs a response to a previously asked question.
Output	Upon update to the database, the system generates a notification to the originator of the question.
Pre-conditions	The user has generated a question on GameOn.
Post Conditions	The submitted response, upon storage in the data base, generates a response message to the question author.
Actions	The system generates a notification to notify a user of an update.

SUPPORT: RESPOND TO QUESTIONS

The following table outlines the “Respond to Questions” use case details:

Use Case ID:2	Respond to Questions
Description	<p>An administrator will successfully log into the GameOn site.</p> <p>The admin will encounter a question in the support area in which he or she is capable of answering intelligently.</p>

	<p>The admin will respond the support question with a posted response.</p> <p>The system will generate a notification to alert the question's author.</p>
Input	Admin uploads a comment to the GameOn support site.
Output	A notification will be sent to the question author.
Pre-conditions	User must be elevated to admin status.
Post Conditions	Question is updated with a comment answer.
Action	An admin adds a comment to a customer question. The system will notify the customer that a comment has occurred.

SUPPORT: UPDATE QUESTIONS

The following table outlines the "Update Questions" use case details:

Use Case ID:2	Update Questions
Description	<p>An administrator will successfully log into the GameOn site.</p> <p>The admin will encounter a question in the support area in that requires modification.</p> <p>The admin will update the question.</p> <p>The system will generate a notification to alert the question's author.</p>
Input	Admin updates a previously posted question on the GameOn support site.
Output	A notification will be sent to the question author.
Pre-conditions	User must be elevated to admin status.
Post Conditions	Question is updated with new wording added by admin.
Action	An admin modifies a customer posted question. The system will notify the customer that an update has occurred.

SUPPORT: VIEW QUESTION DETAILS

The following table outlines the “View Question Details” use case details:

Use Case ID:2	View Question Details
Description	<p>An administrator will successfully log into the GameOn site.</p> <p>The admin will select a question to view metadata.</p> <p>The admin will be presented with the question metadata.</p>
Input	Admin selects a previously posted question input by a customer in the support area of the site.
Output	A summation of the metadata will be presented to the admin.
Pre-conditions	Admin must select a previously asked question.
Post Conditions	Metadata is presented to the admin.
Action	An admin selects a question to view the inherent question metadata.

SUPPORT: RESPOND TO COMMENTS

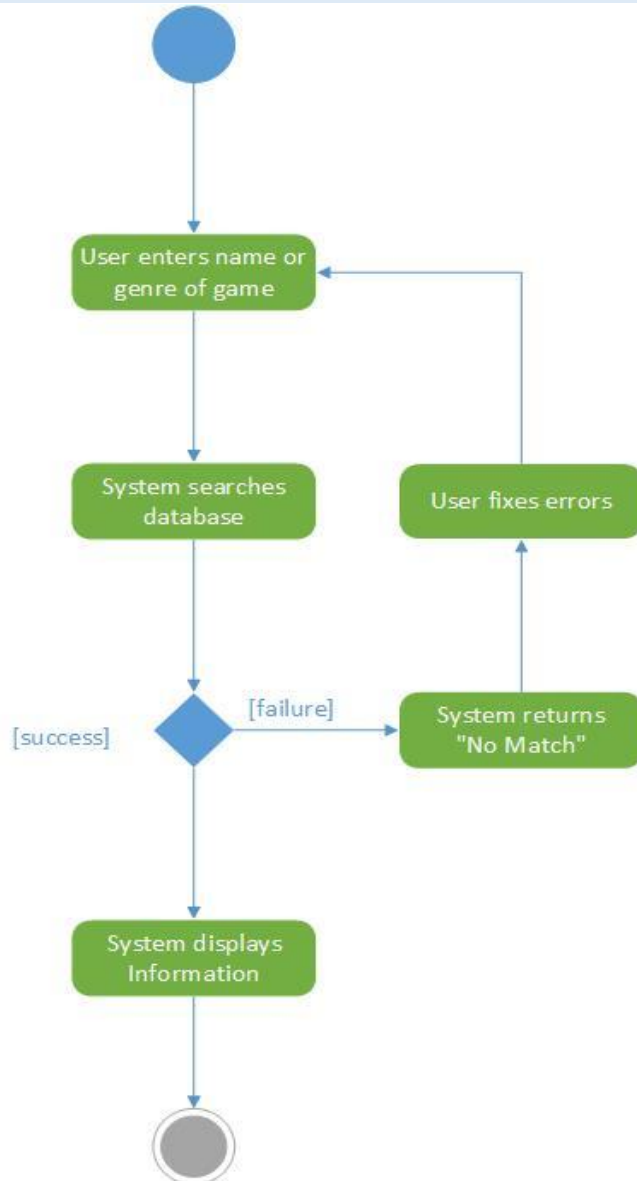
The following table outlines the “Respond to Comments” use case details:

Use Case ID:2	Respond to Comments
Description	<p>A GameOn user will successfully log into the GameOn site.</p> <p>The user will select a comment to respond to.</p> <p>A notification will be sent to the original poster.</p>
Input	A GameOn user will add a comment to a previous post.
Output	A comment will be uploaded to the database.
Pre-conditions	User must have successful GameOn credentials.
Post Conditions	A comment is posted to the GameOn database.
Action	A user posts a comment to support question, generating a system notification directed towards the original poster.

SYSTEM MODELS

MODULE - GAME SEARCH:

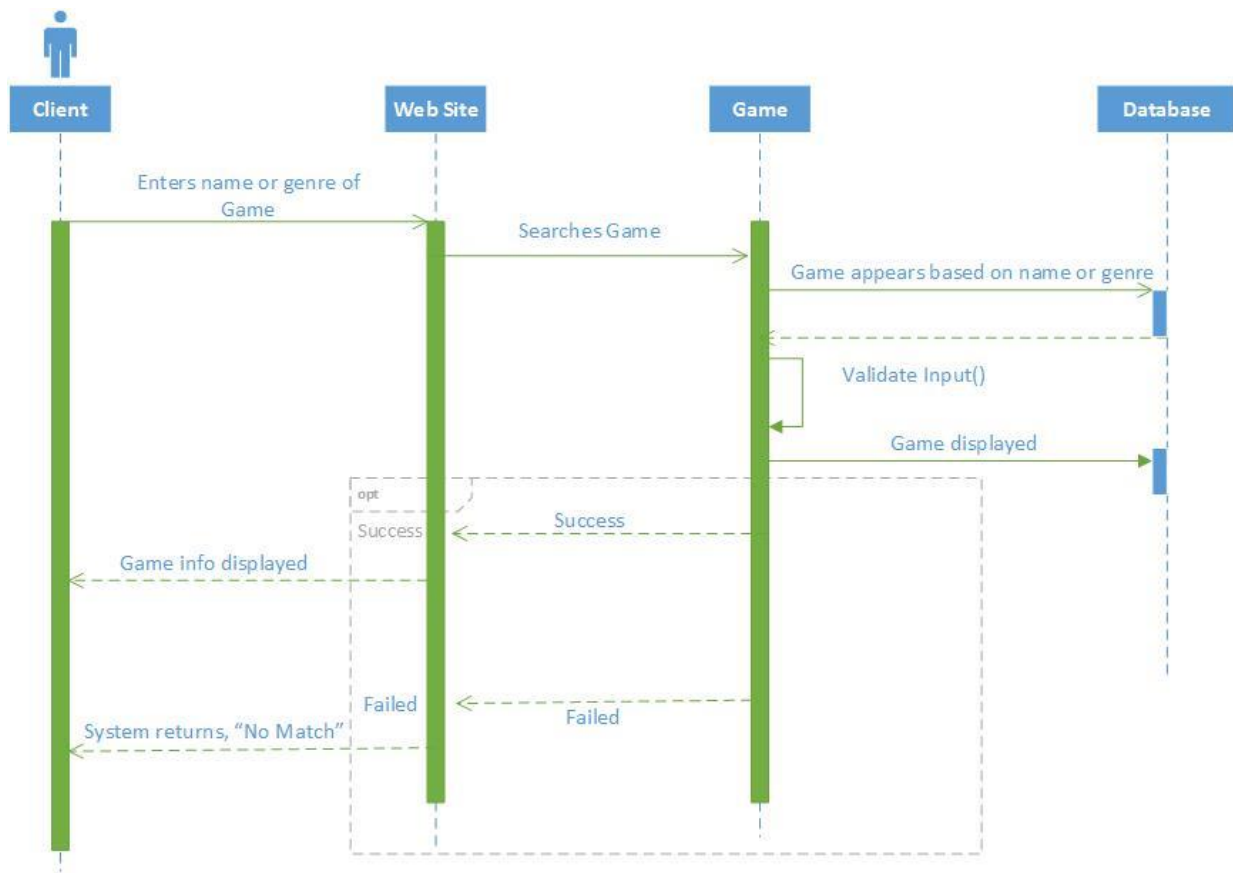
GAME SEARCH: SEARCH GAMES



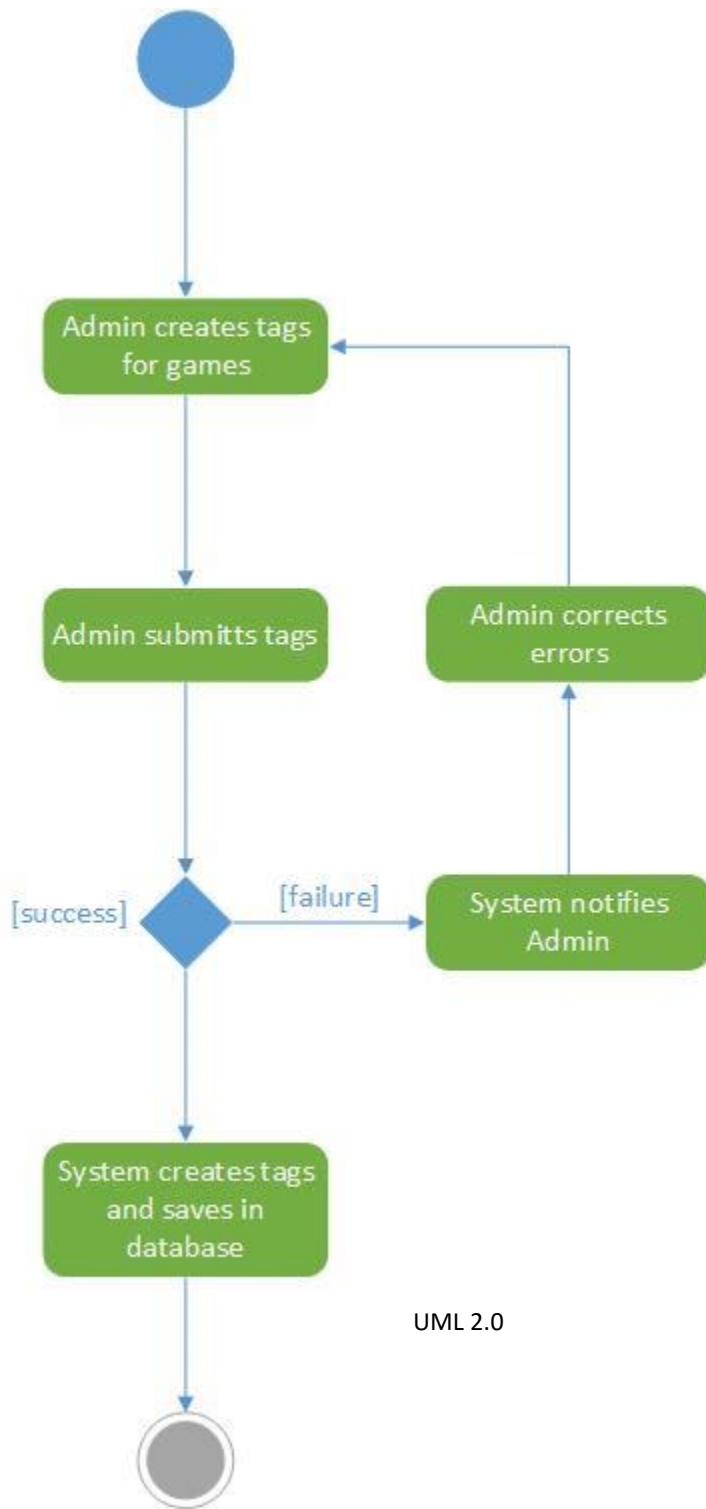
This diagram describes the process that takes place in the search games process. The user or customer will enter the name of the game or the genre of the game they would like to search. The user/ customer can enter words or partial words in the search bar.

Once the user/customer presses enter, the system will search the database. If the search was successful, the system will display the information of the game.

If the search failed, the system will return a message saying "No Match". The user/ customer will have to fix the errors and retype their search in the search bar.



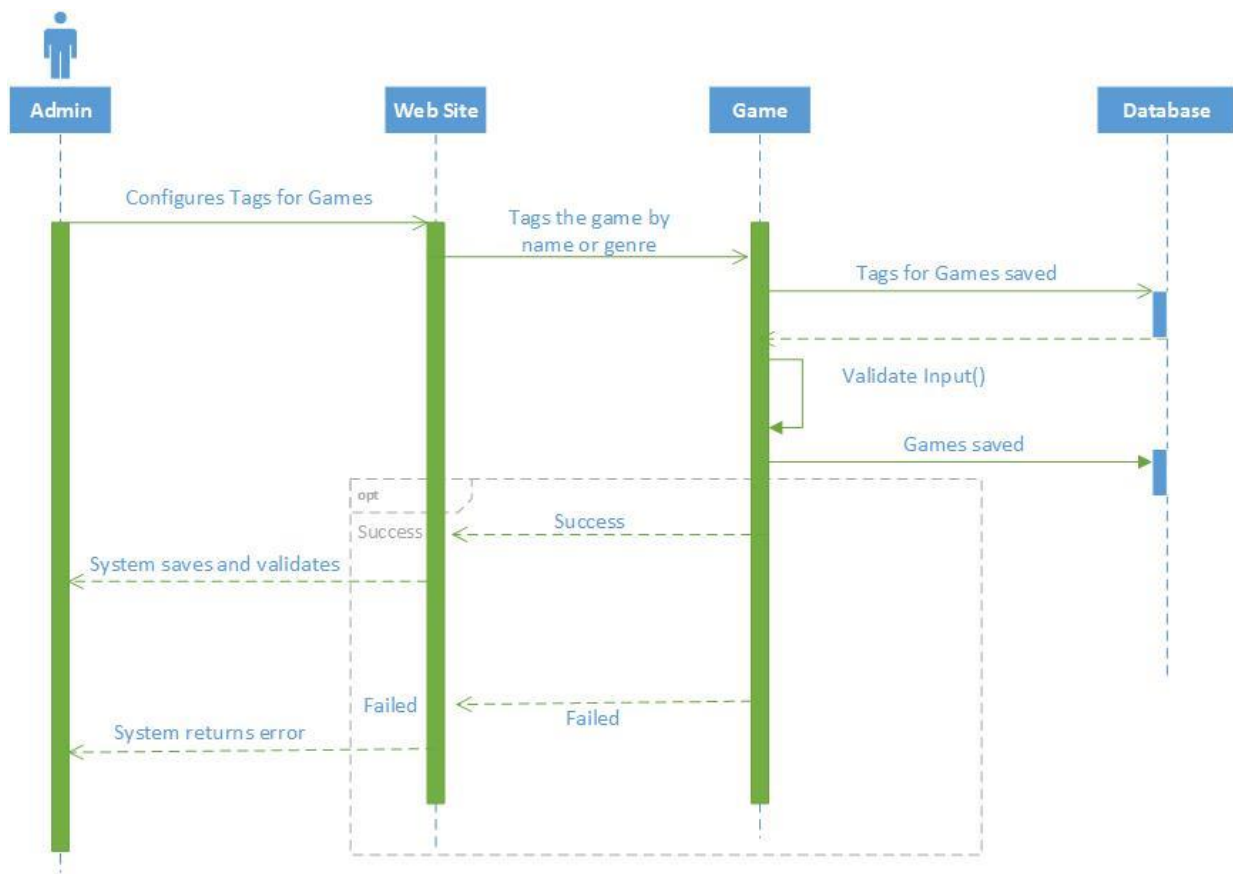
GAME SEARCH: CONFIGURE TAGS



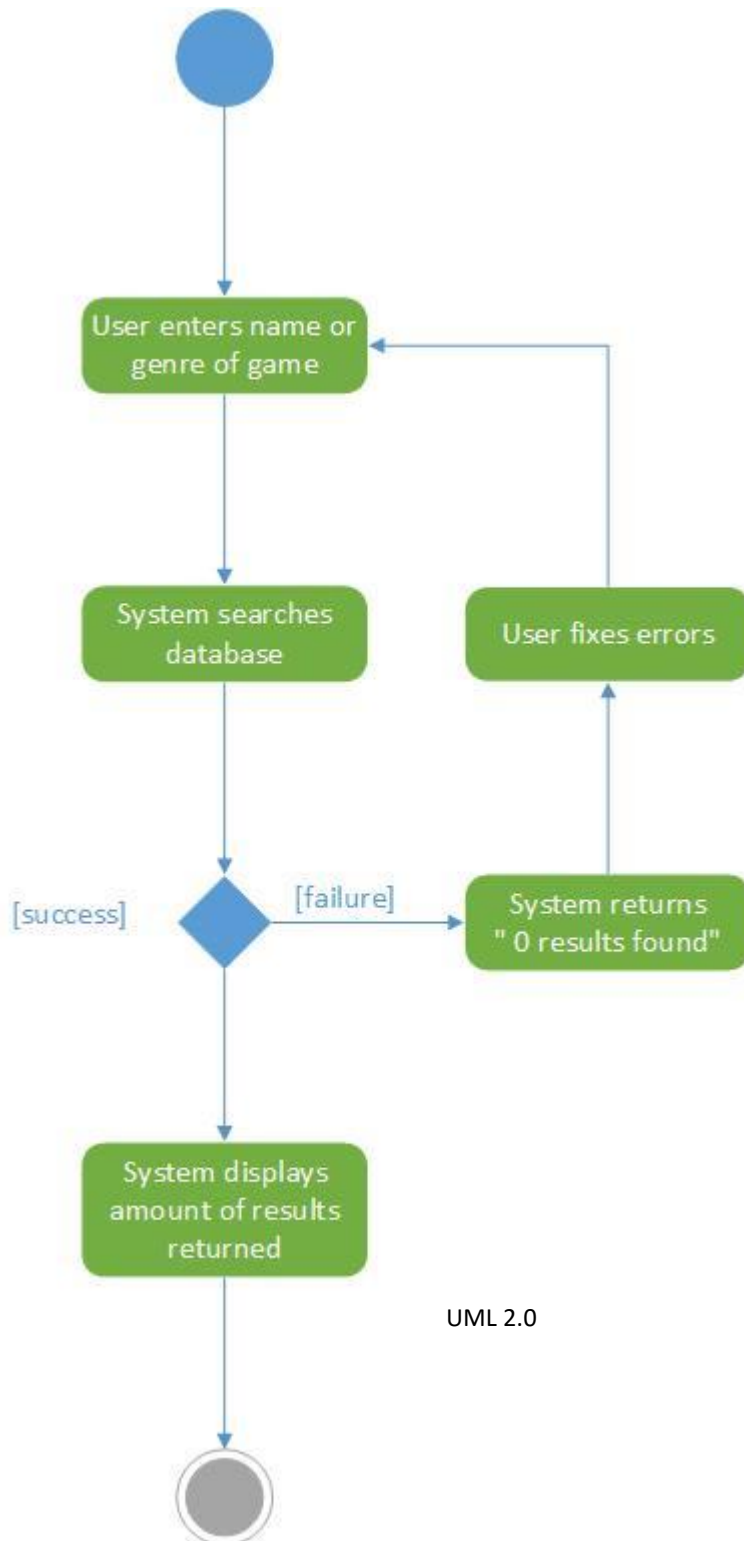
This diagram describes the process that takes place as part of the configure tabs process. An admin will create tags for the games they would like to create for. The admin will create tags based on the name of the game or the genre of the game. Then the admin will submit the tags.

If the tags are made successfully, the system creates tags and saves it in the database. If the tags are not made successfully, the system will notify the admin. The admin will correct the errors and the admin will recreate the tags for submission.

UML 2.0



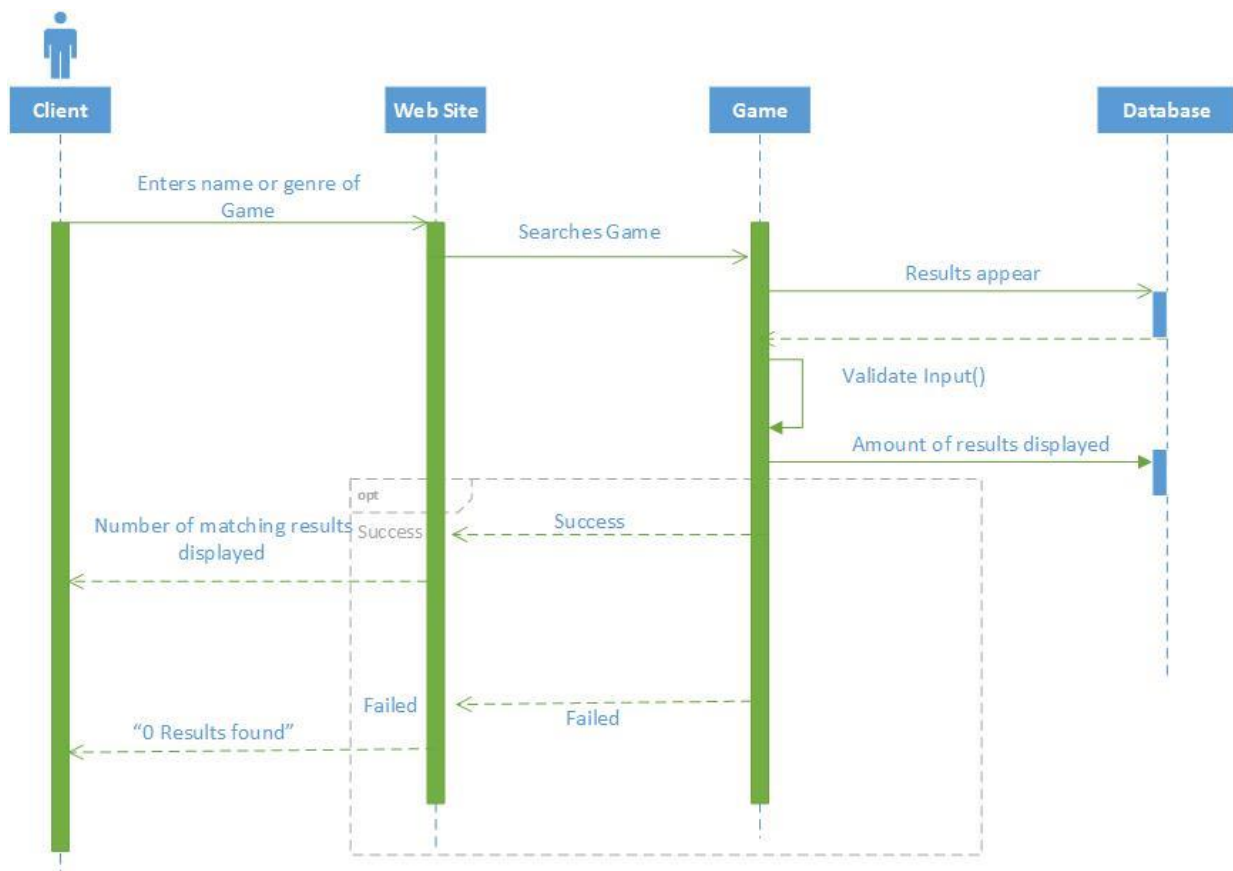
GAME SEARCH: SET RESULTS RETURNED



The diagram describes the process that takes place as parts of the set results returned process. The user/customer will enter the name or genre of the game they like to view. The system will search the database of games.

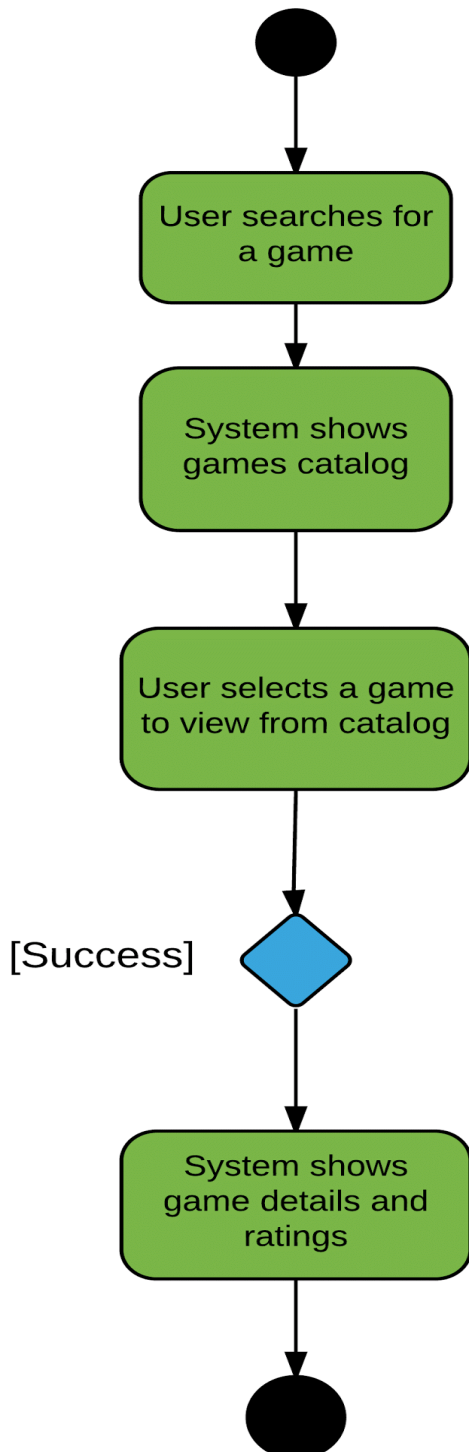
If the search was successful, the system will display the number of results that are returned. If the system fails, it will return a message saying, "0 results found". Thereafter, the user/customer will have to fix the errors and re-enter into the search bar again.

UML 2.0



MODULE - GAME CATALOG:

GAME CATALOG: VIEW DETAILS

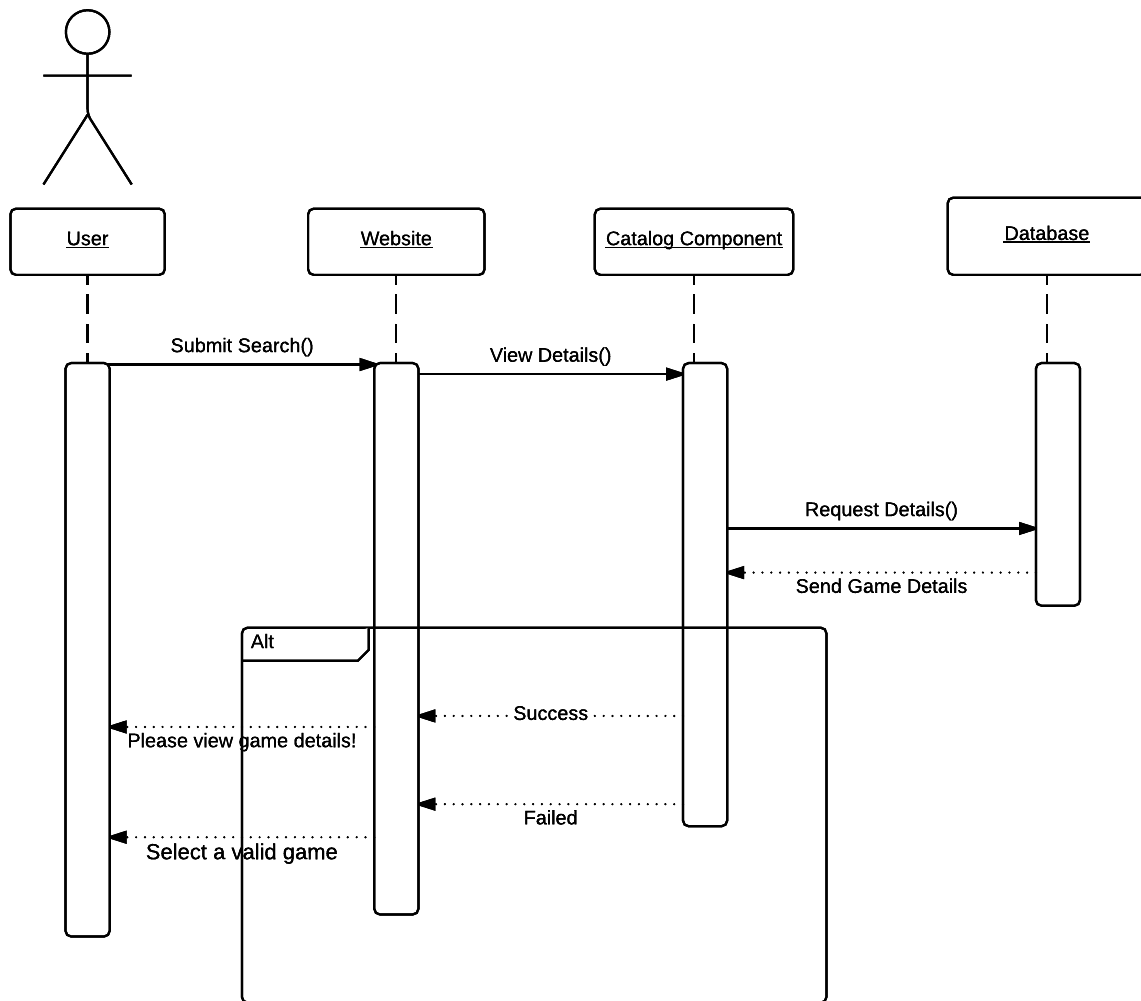


The diagram to the left shows the process that is ran when a user wishes to view the details of a game. As shown, the user first has to search for the game and make a query. The system then presents the user with the game list or search results. The user can apply different filters in the search to narrow down the games list.

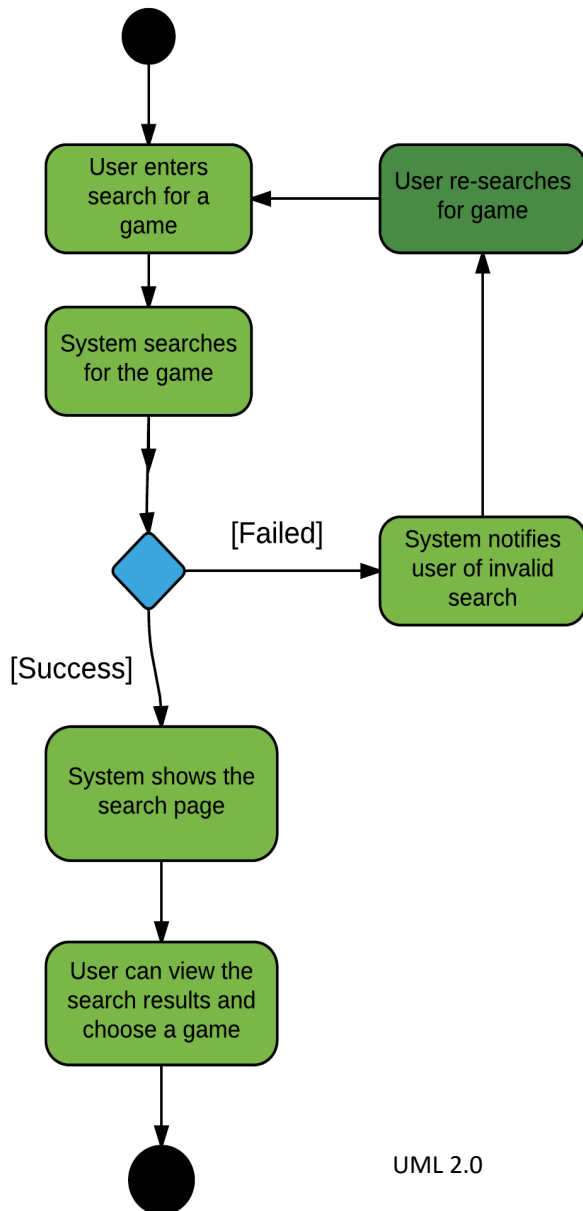
Once the system shows the user the list, the user must select a game from the list that they want to view and check the details of the game. Once the user selects the game and the page opens successfully, the system will then show the game details such as genre, platform, description, ratings and reviews by other users.

The sequence diagram below shows the process that takes place when a user wants to view the review. The process is called 'View Review'. A user must submit a search for a game in order to see the list of games and once the system shows the user the games list, they can click on the game and 'View Details' of the game.

As shown, the request is sent to the database for the game details and it sends back the game details which goes back if successful. The user is then able to view the details of the game. This is an alternative view of the process compared to the other diagram above about the 'View Review' Process.



UML 2.0



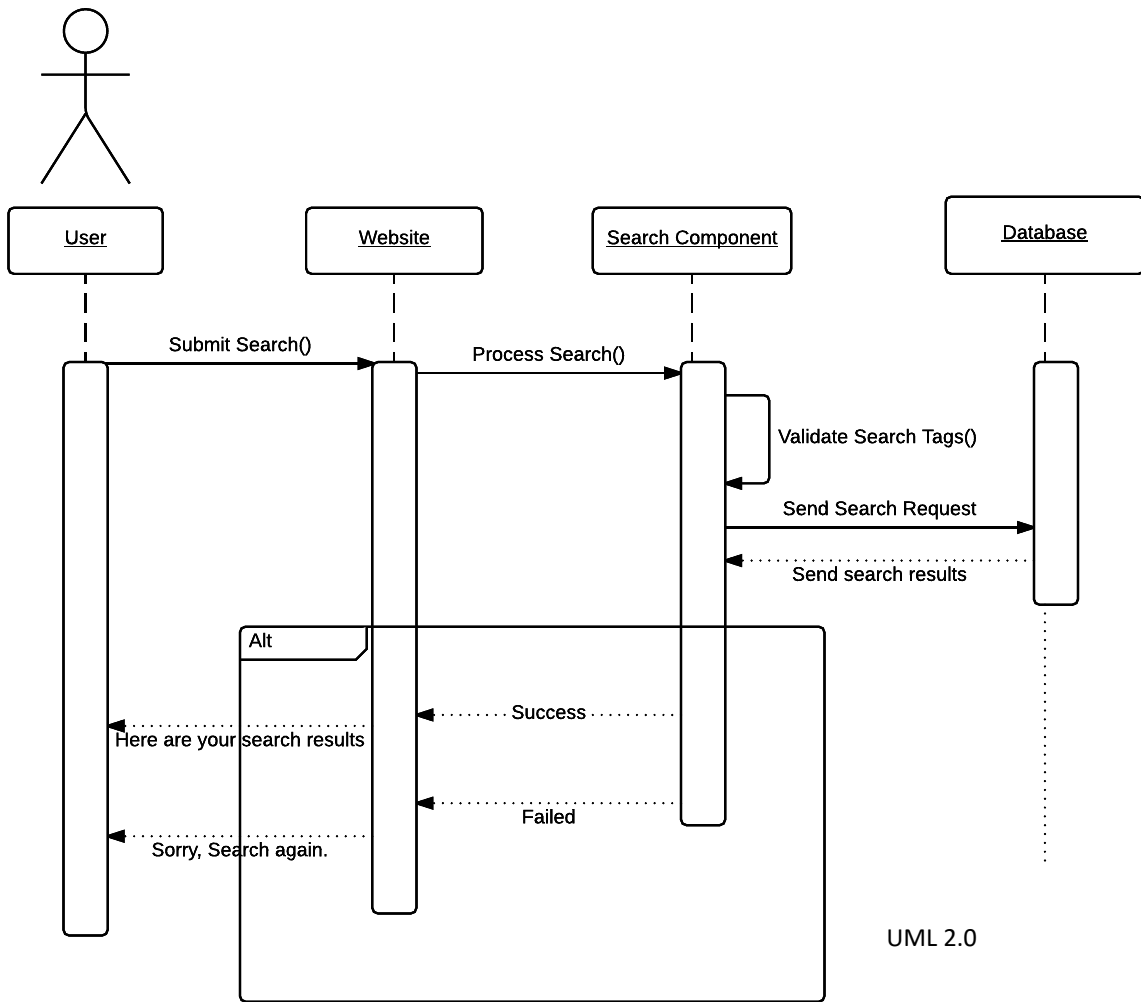
The diagram to the left shows the process when a user wishes to view the catalog. The catalog contains the games list and in order to that, the user must be registered and make a search.

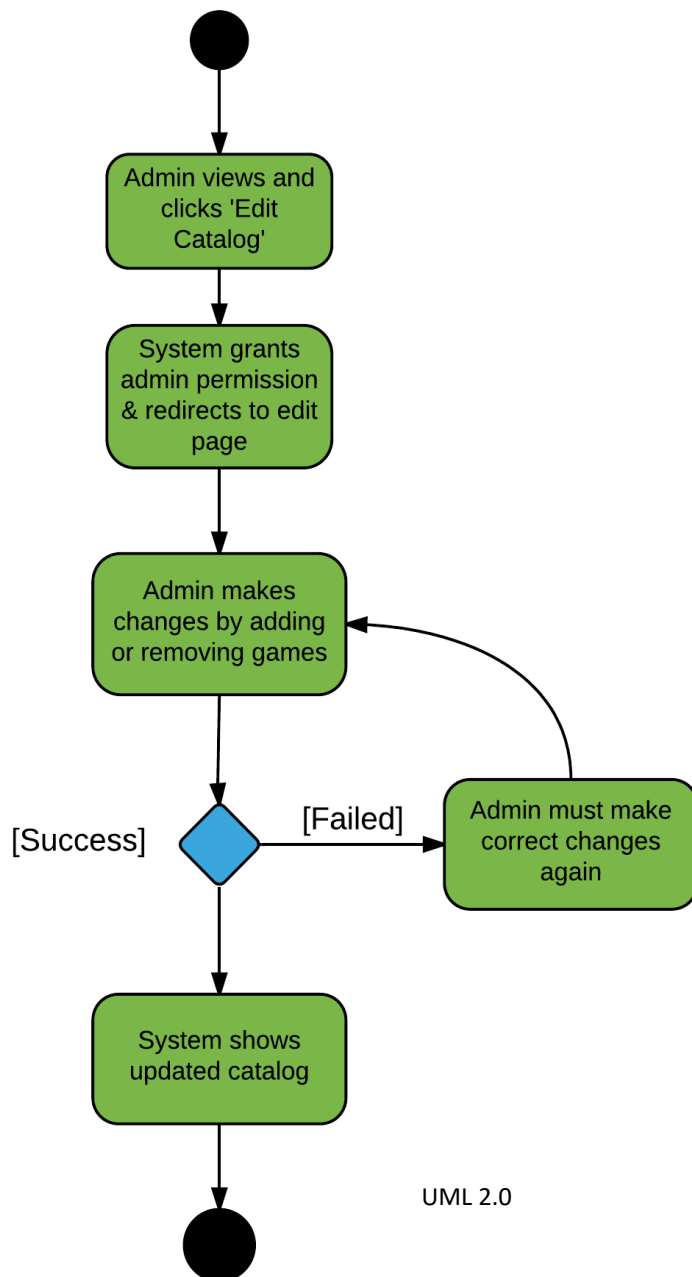
The search must be correct or else the results will not show. The user has to enter search tags that are connected to the game or else the search will not occur correctly. The user can also filter the search or sort the games by attributes such as genre, platform, ratings, name, etc.

As the diagram shows, the user if they fail to enter a valid search, they have to re-enter the search. If the search is successful, the system will show the search results and the list of all the games, which is the game catalog.

The sequence diagram below shows how the user has to submit the search and the system will process the search according to valid search tags. If the search tags are not valid, the system will notify the user of the error.

The system requests the database for the search and the database sends back the games list so that could be displayed to the user as they necessitated. Once successful, the user can view and select the game they wish to view.





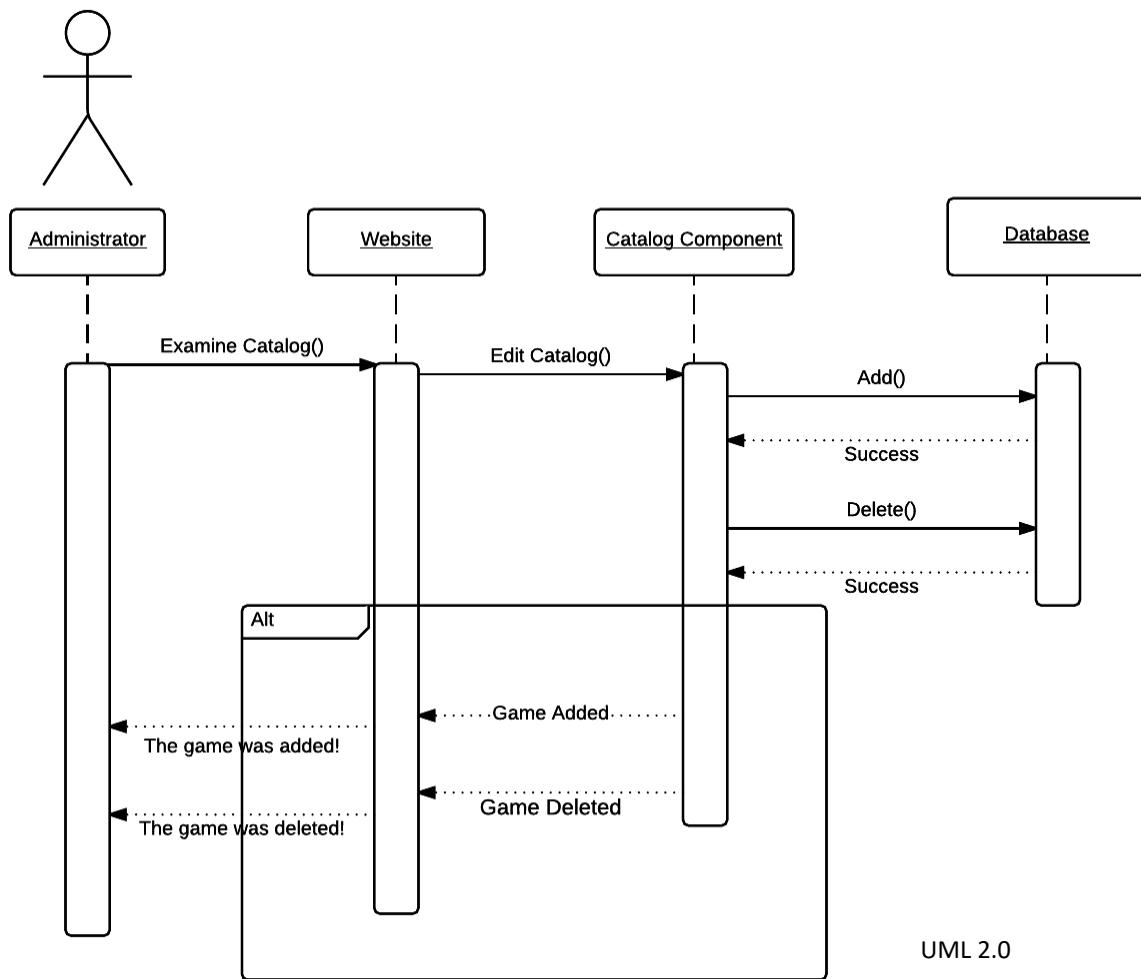
The configure catalog diagram to the left shows the steps that are taken in order for an administrator to apply changes. The administrator has the power to edit the catalog by adding or deleting games.

As shown in the steps, the admin first has to do the same steps as a regular user to get to the games list. Once they are there, they have to 'Edit Catalog' and then they are allowed to add or delete the games once the system grants them permission.

The admin must make proper changes. There are conditions for that, such as there can be no duplicate games. If the administrator fails matching those conditions, the system shows the admin the message to correct the errors. If successful, the changes take place.

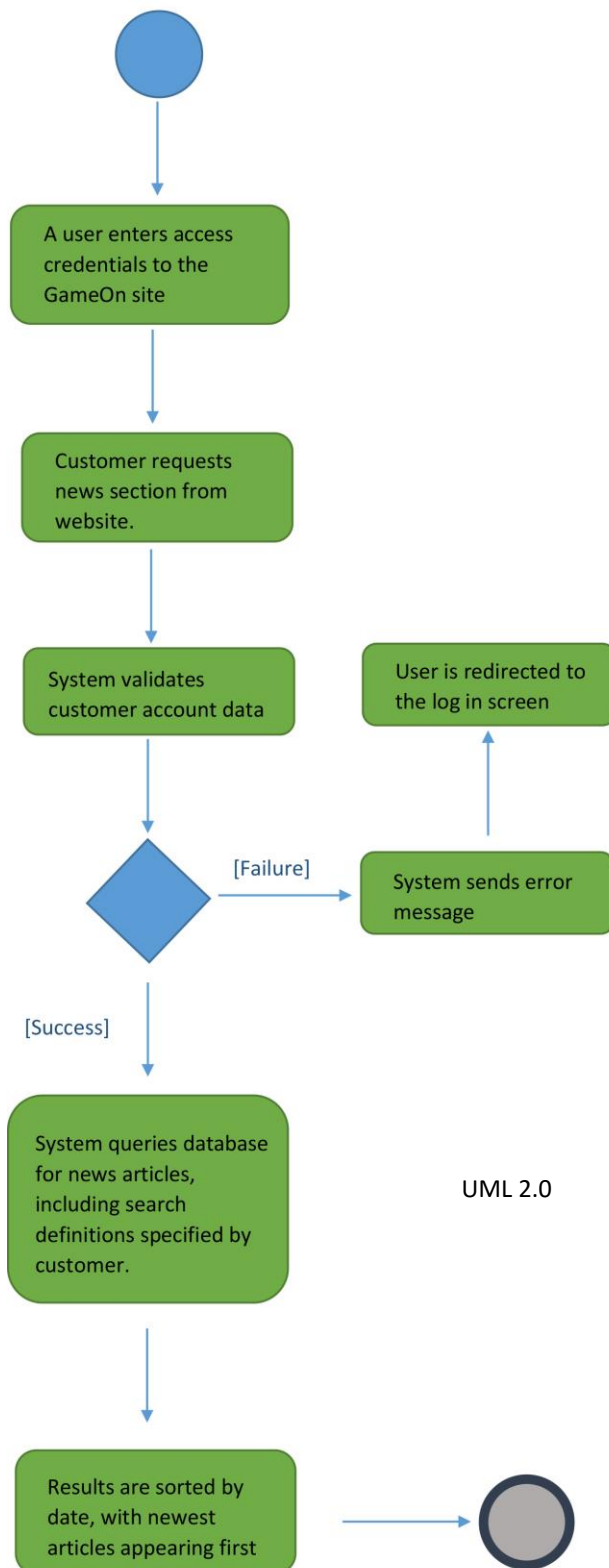
The sequence diagram below shows an alternative view of the process in the 'Configure Catalog' section. It shows that the administrator has to examine the catalog or view the catalog and then select the 'Edit Catalog' and once

that is done, they can add or delete the games from the database. The database responds and adds or deletes the game and if successful, the games catalog updates to the newest changes.



MODULE – NEWS ARTICLES

NEWS ARTICLES: BROWSE ARTICLES



UML 2.0

This diagram describes the process that takes place as part of the browse articles process. The customer accesses the news article page and inputs any additional search fields requested. Once the user submits the news request, the system first checks that the user is a valid customer.

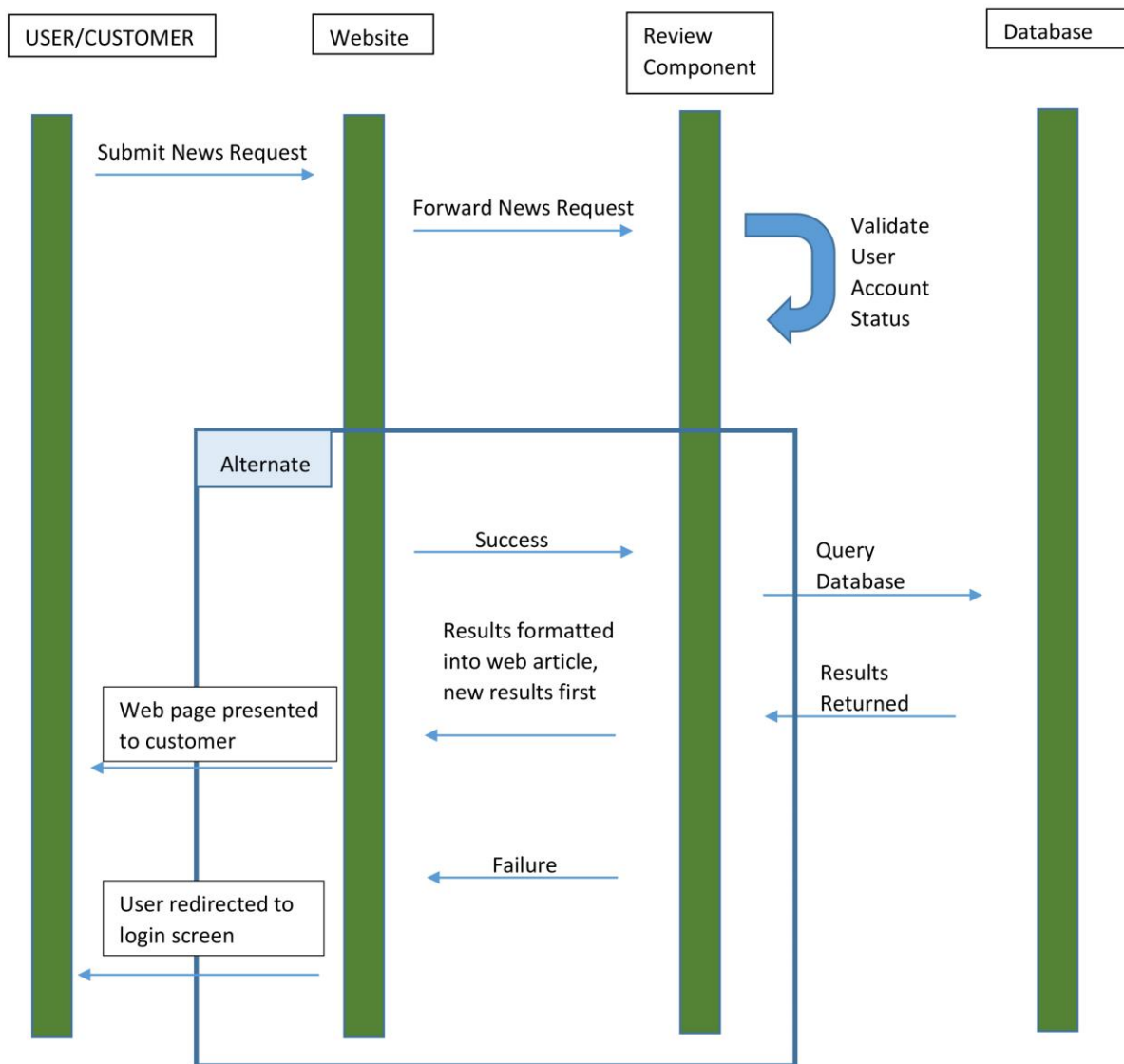
If the system determines an unauthorized news request, the system will return an "unauthorized user" error message with no news results. The user will be redirected to a login screen.

Upon input of a successful search query, the system will return all relevant news articles, sorted by date, with newest articles appearing at the top of the article list.

The same process can be seen in the sequence diagram below. The Web site component forwards all news requests to the review component, which determines eligibility to receive content. The diagram also describes an alternative path for successful and failed transactions.

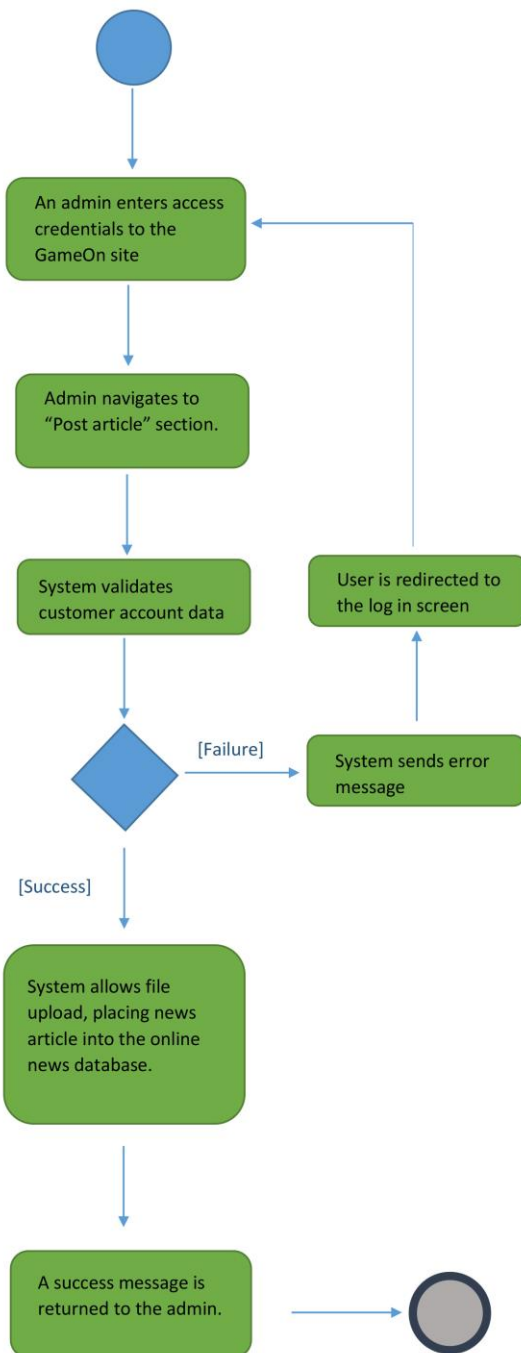
A successful account validation forwards a query to the database. The data is then returned from the data base and presented to the web site. The web site then formats the information into a dynamic web page custom to the user, with newest results sorted first.

A failed account validation returns no information from the data base, and redirects the user to the log in screen.



UML 2.0

NEWS ARTICLES: POST ARTICLES



This diagram describes the process that takes place as part of the posting articles process. An admin creates a news article page and inputs any additional definition tag fields relevant. Once the admin submits the news article, the system first checks that the admin is a valid account with proper privileges.

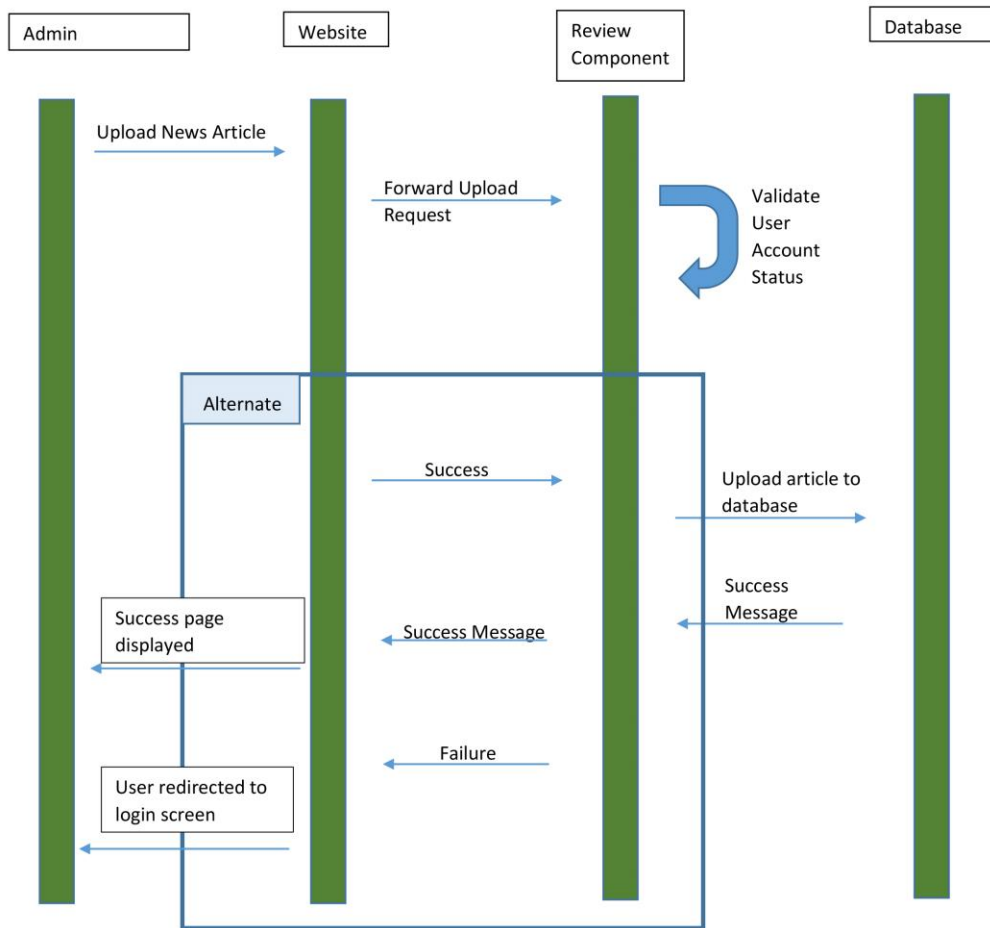
If the system determines an unauthorized upload, the system will return an "unauthorized user" error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will upload news articles, tagged by date and any related tags the admin has defined.

The same process can be seen in the sequence diagram below. The Web site component forwards all news uploads to the review component, which determines eligibility to upload content. The diagram also describes an alternative path for successful and failed uploads.

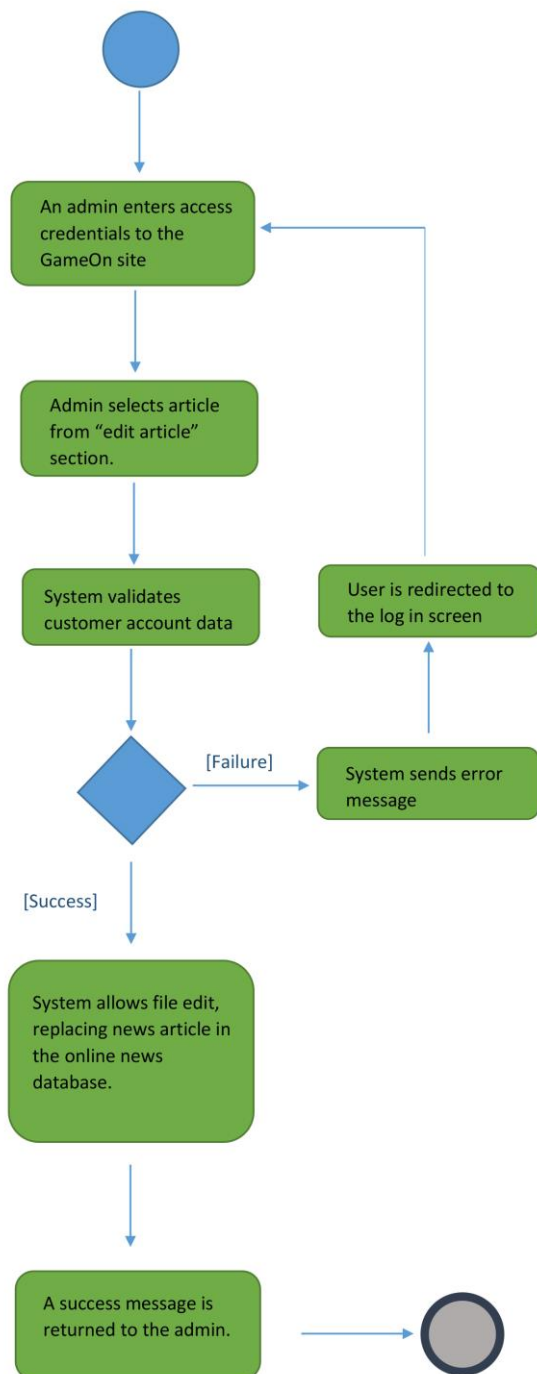
A successful account validation forwards an article to the database. The success message is then returned from the data base and presented to the web site. The web site then forwards the success message to the admin, letting them know the upload has been successful and the content is now live.

A failed account validation uploads no information to the data base, and redirects the user to the log in screen.



UML 2.0

NEWS ARTICLES: EDIT ARTICLES



This diagram describes the process that takes place as part of the edit articles process. An admin accesses the news article page and selects an article to edit. The system then checks that the admin has proper authorization.

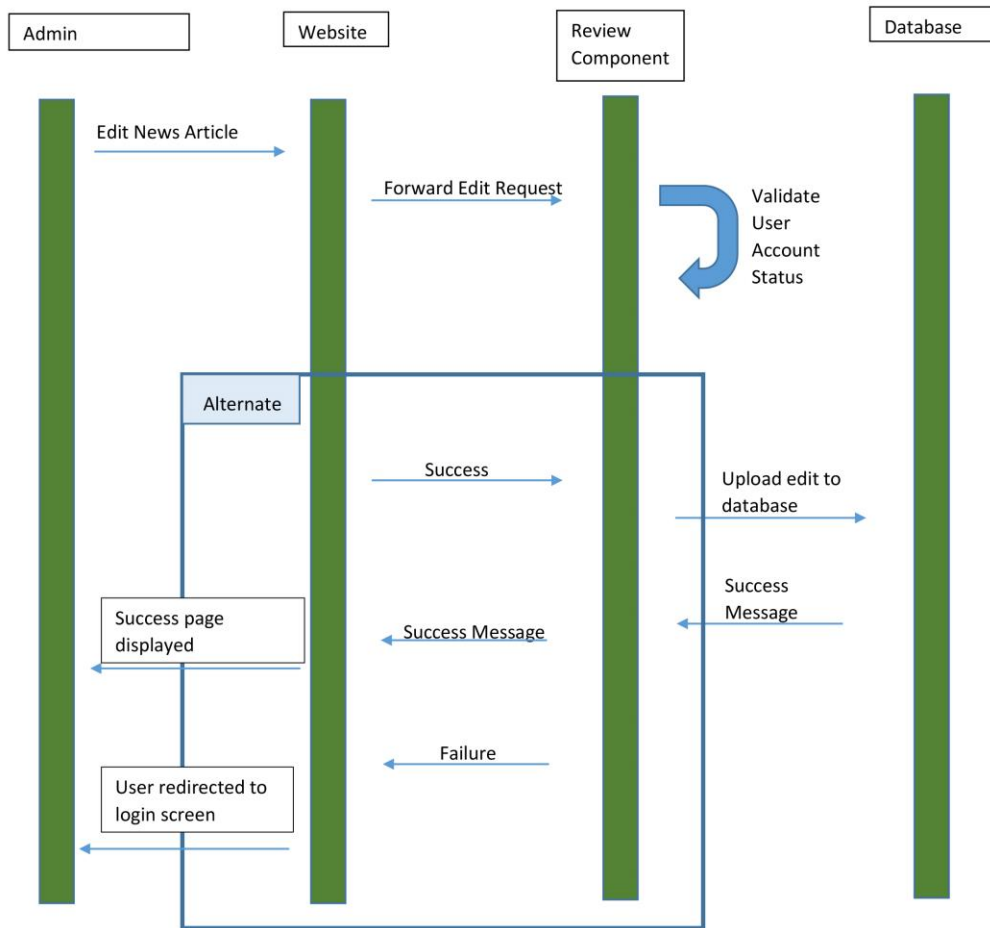
If the system determines an unauthorized edit request, the system will return an "unauthorized user" error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will upload all news articles updates.

The same process can be seen in the sequence diagram below. The Web site component forwards all edit requests to the review component, which determines eligibility to edit content. The diagram also describes an alternative path for successful and failed transactions.

A successful account validation forwards a edit to the database. A success message is then returned from the data base and presented to the web site. The web site then formats the information into a dynamic web page custom to the admin, informing the admin that the edit was successful.

A failed account validation edits no information in the data base, and redirects the user to the log in screen.



UML 2.0

REVIEWS: POST REVIEWS

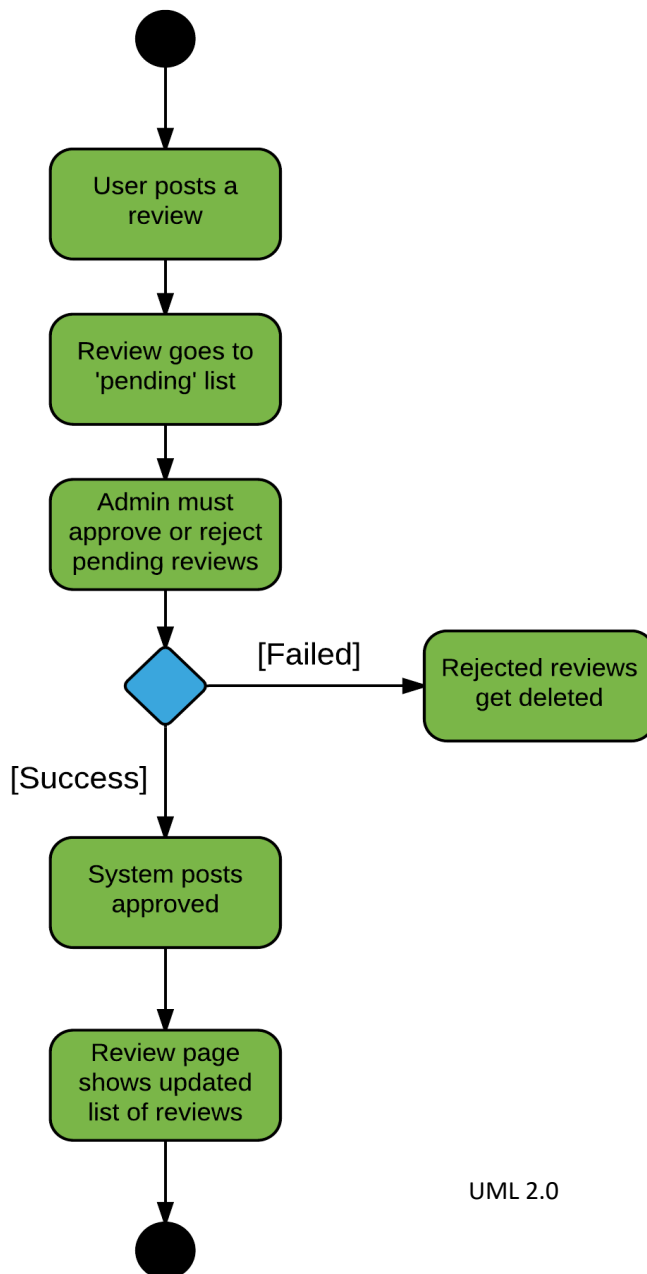
The diagram to the left represents the process of posting a review by the user. A user must be registered to perform this action. As the diagram shows, the user has to first select the game and then post a review. When a user posts a review, the system is set up to have a hold on reviews that are submitted.

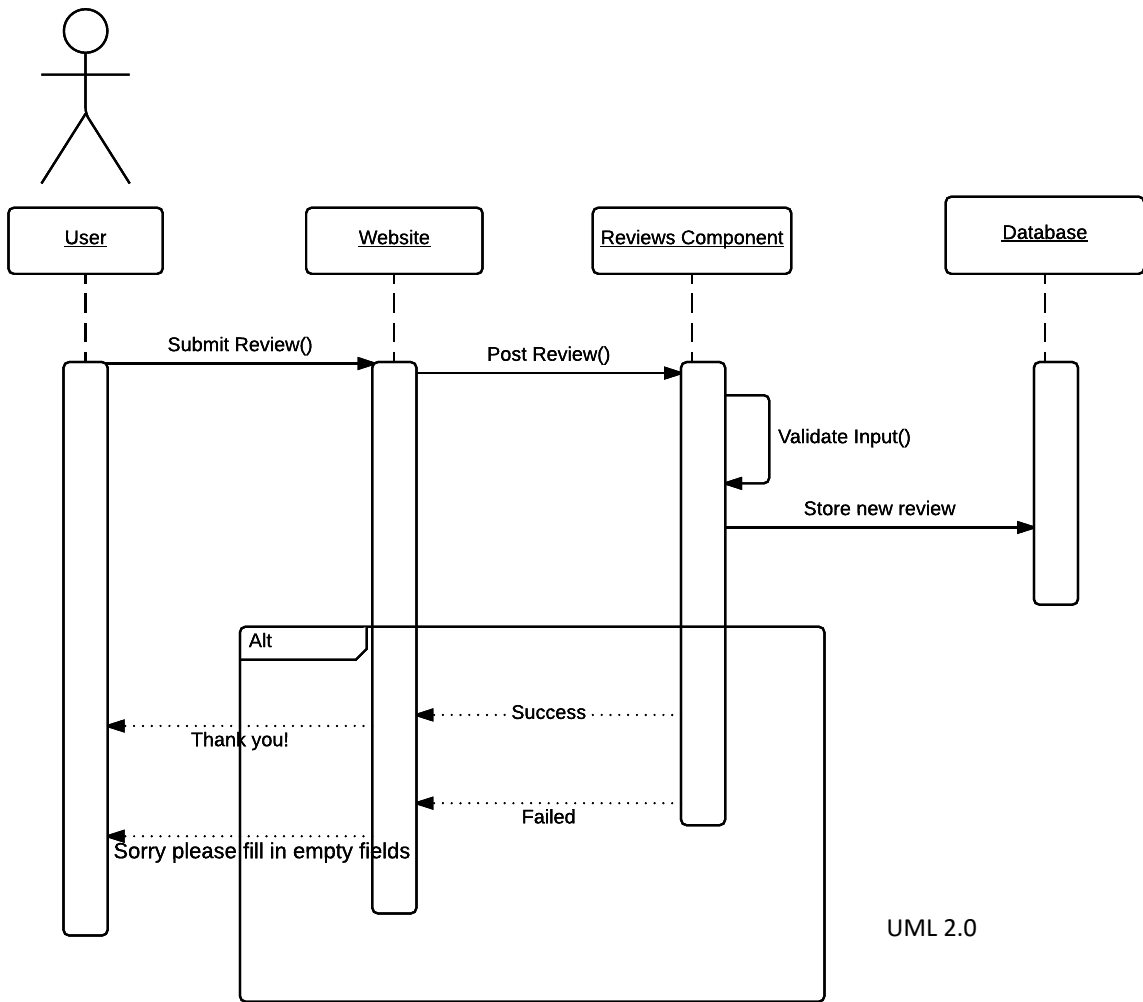
The 2nd stage of the process is the review going into the pending list. The pending list holds all the reviews which are not yet checked by the administrator.

The administrator has the rights to approve or reject the reviews if they are appropriate or inappropriate. Once the administrator approves the reviews, the system posts the latest review and updates the current list of reviews. If the administrator rejects the review, it gets deleted.

Below is the sequence diagram that shows a detailed process of how the process takes place. The user submits review and then the website posts it. The validate input section is the administrative part where it checks if its appropriate or not. If it is, the review goes to the database and gets

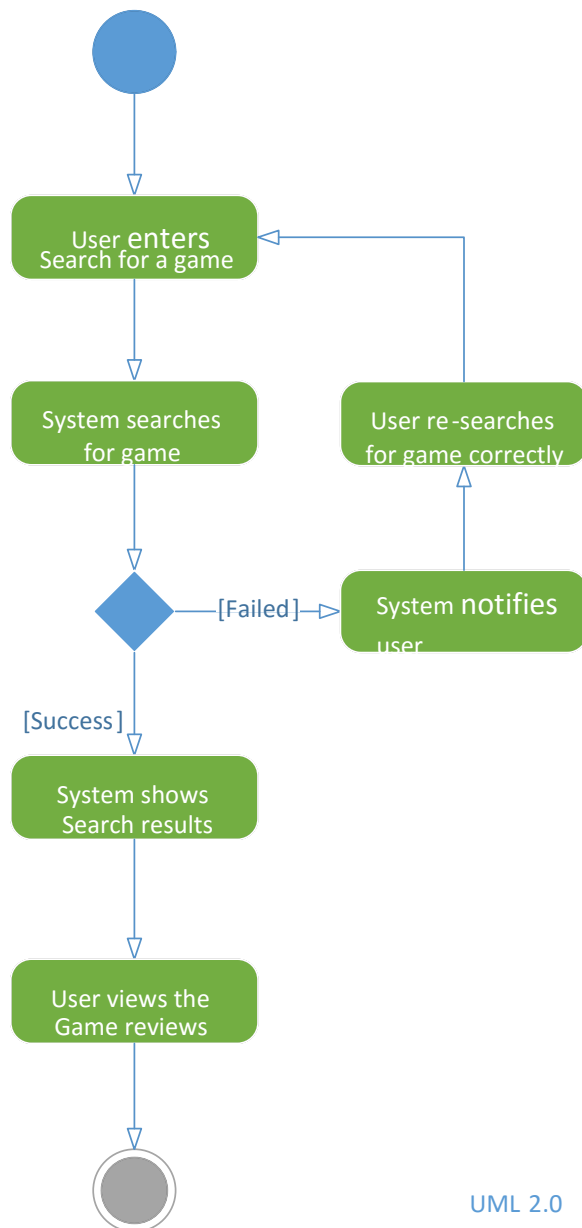
posted. The review page returns a message saying Thank You! After a user submits their review. If the user does not fill in correctly, the system automatically denies the post.





UML 2.0

REVIEWS: VIEW REVIEWS

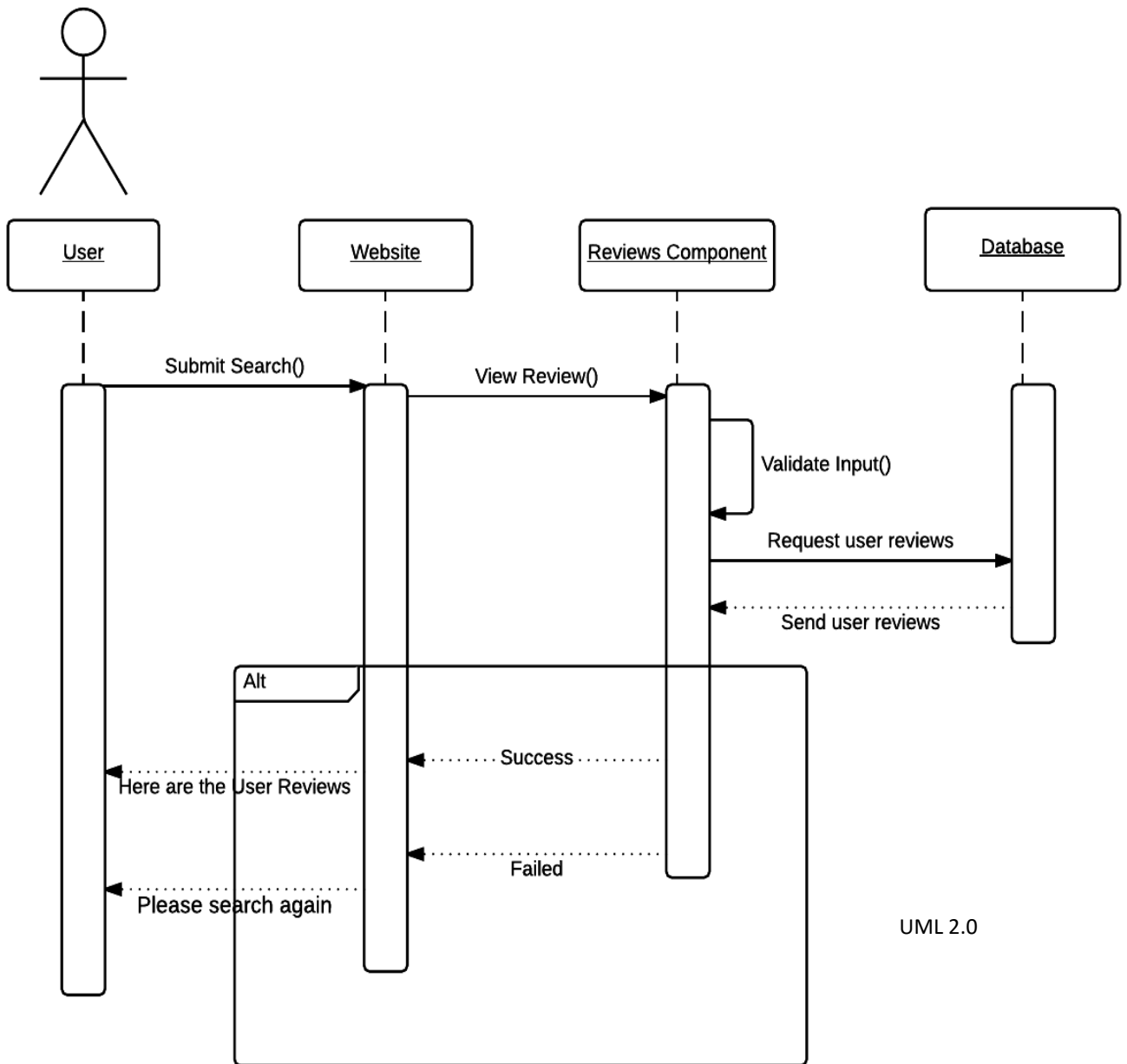


UML 2.0

The diagram to the left represents the 'View Reviews' process. As seen the first step is that the user must search for the game using the games catalog. Once the search is submitted, the system searches for the game from the database and if the user has not put a valid search tag in the search bar, the system notifies the user about the error. The user must re-enter the search.

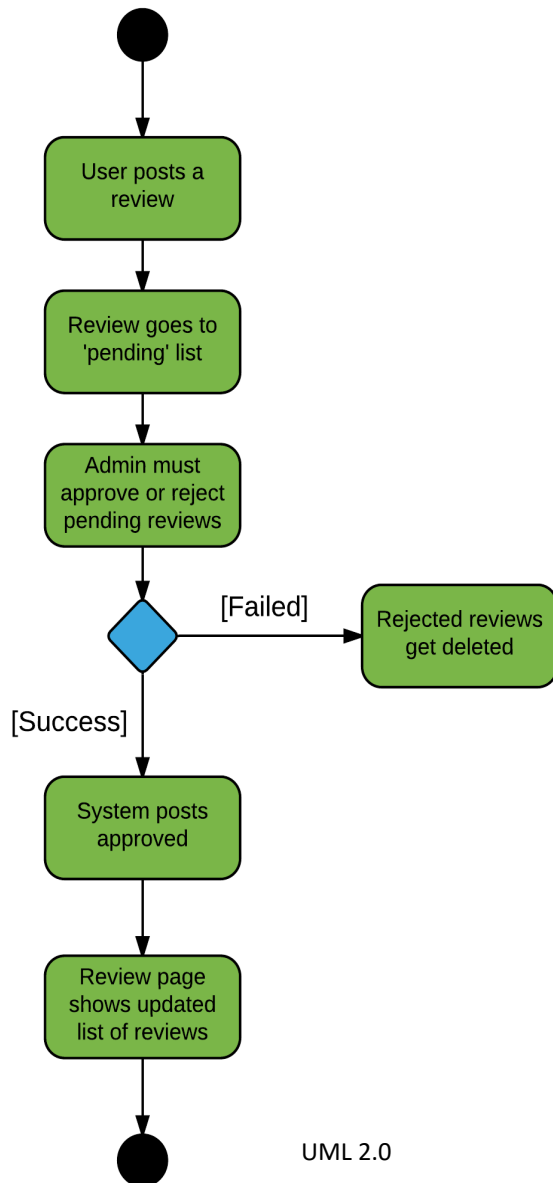
If the search is valid, the system shows the search results. The user will be able to select any game and click on the reviews to view reviews posted by other users.

Below is the sequence diagram that shows a much detailed view of the process. It shows how a user must search first then the system puts out results. The system has to first request the database for the user reviews and it replies back with the user reviews that are showed to the user on reviews page.



UML 2.0

REVIEWS: CONFIGURE REVIEWS

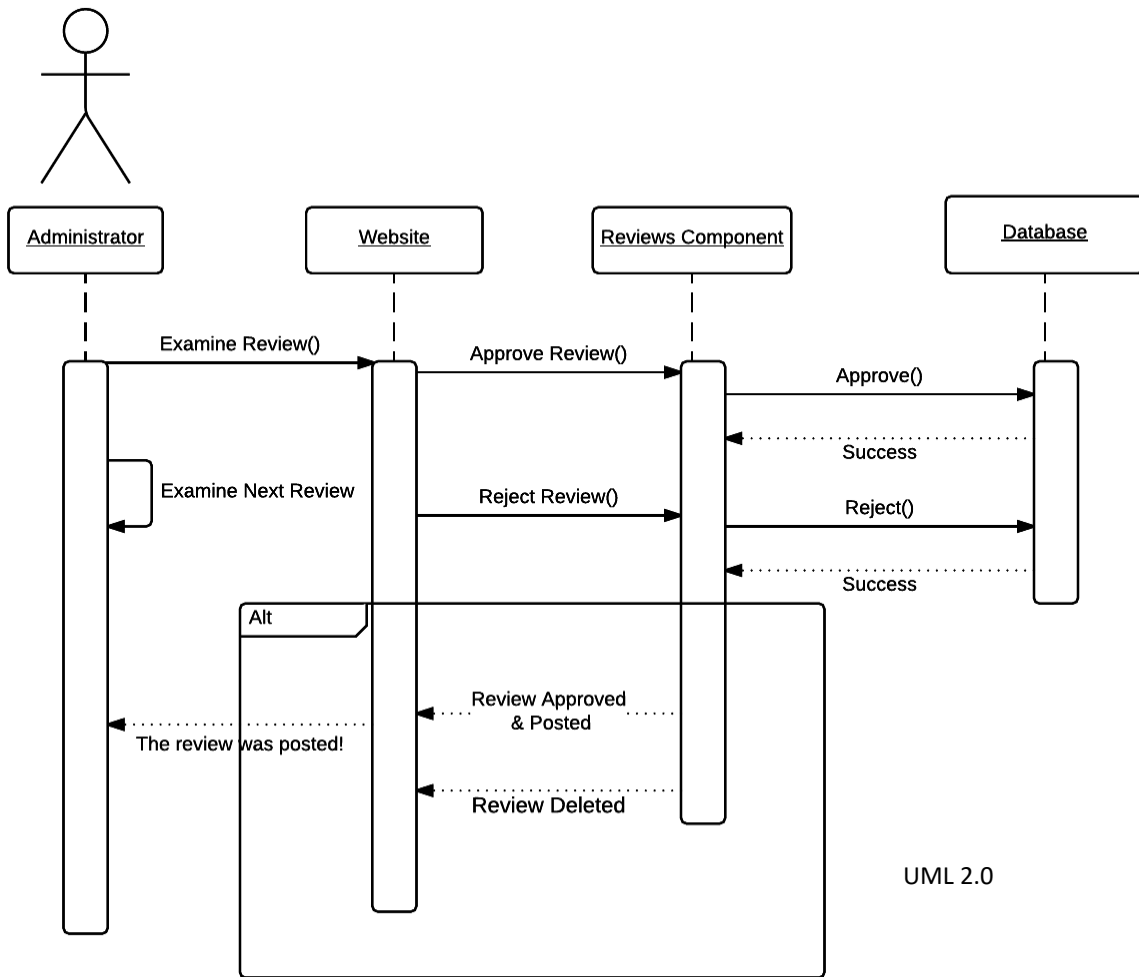


The diagram to the left is the process for the administrator in which the administrator can approve or reject posts. The process is called 'Configure Reviews'. The users post reviews for games but they get held back before they get published on the website. They need to be first administered to see if they are correct and appropriate to be posted. As the chart shows, the user posts a review, it goes to the 'pending' state and then it requires the admin's approval.

The administrator has the power to approve or reject the pending reviews. If the review is approved, the system goes on and processes it to update on the reviews page. It updates the reviews page and any registered user can now see the newly added review. If the review is rejected by the administrator, the review gets deleted.

The sequence diagram below shows a detailed view of the process. It shows how the administrator starts off by examining the reviews and if the review is appropriate, the administrator gets to approve the review. If the review is inappropriate, the administrator gets to reject the review. The database gets

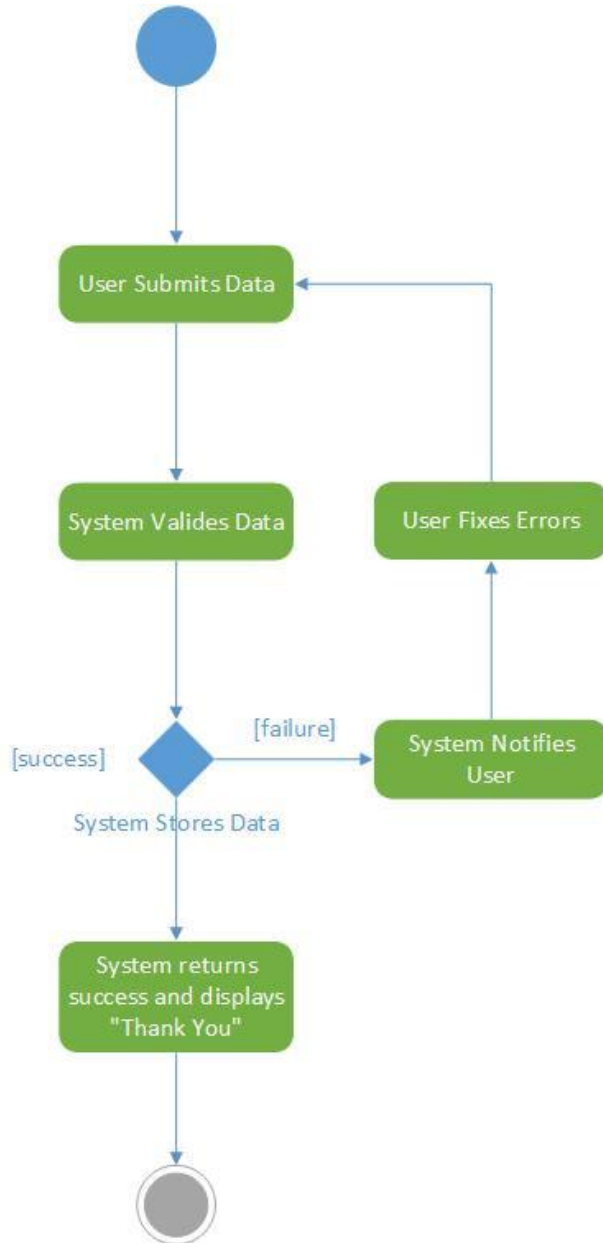
told if its approved or rejected and then the update goes back to the website and the system then updates the reviews page with the newly added review.



UML 2.0

MODULE - NEWSLETTER

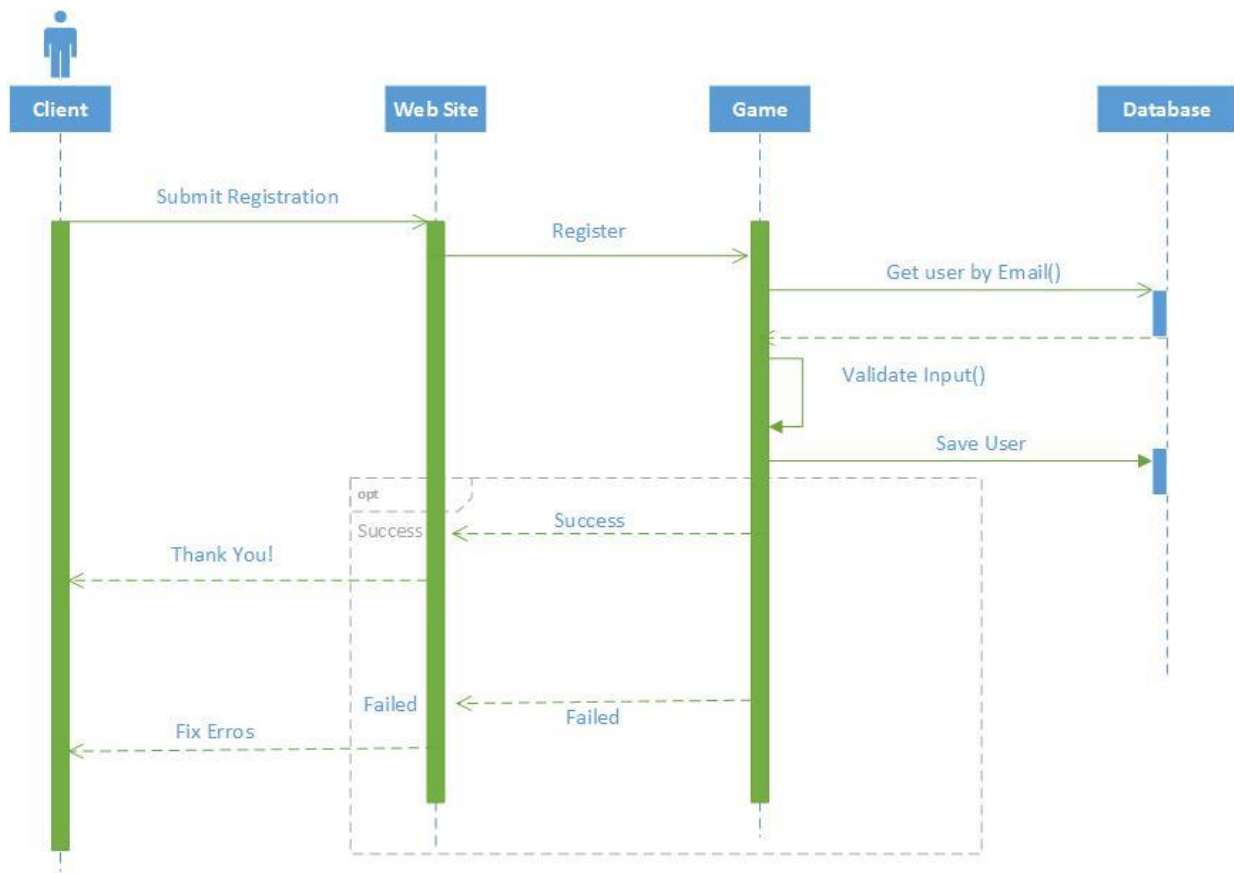
NEWSLETTER: REGISTER



This diagram describes the process that takes place in the registration process. The user will submit the required information that is needed to register in order to receive Newsletters. The system will validate the data.

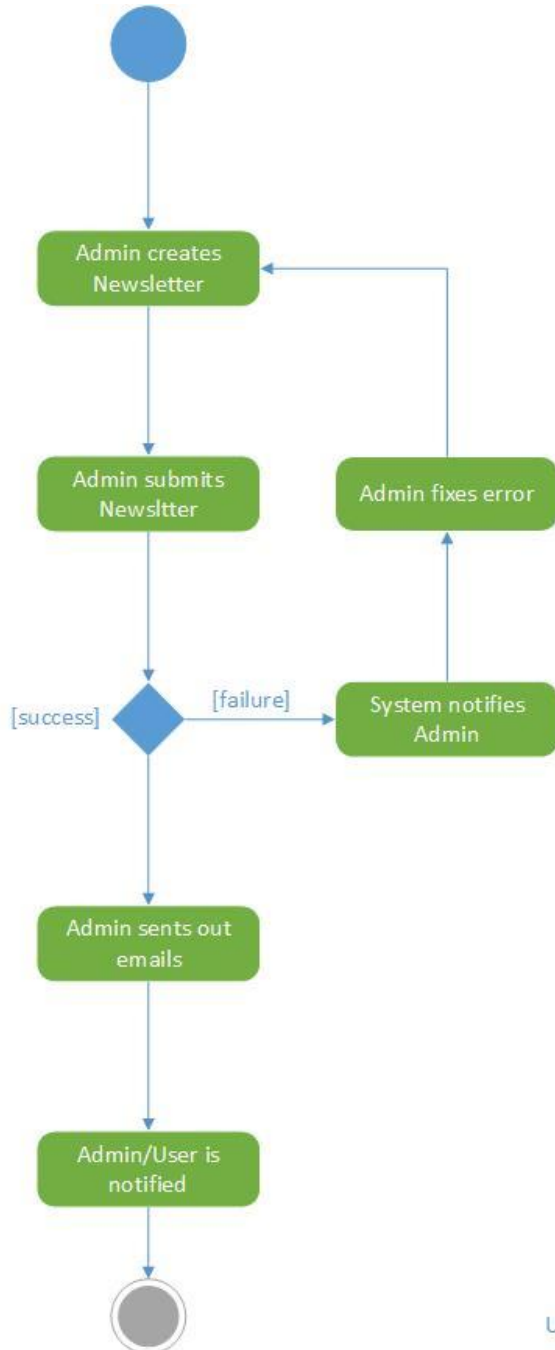
If the data is successful, the system will return success, and display a message saying "Thank You".

If the data fails, the system will notify the user, and the user will fix the required errors. Once the user fixes the errors, they will be able to resubmit the registration process.



UML 2.0

NEWSLETTER: CREATE NEWSLETTER

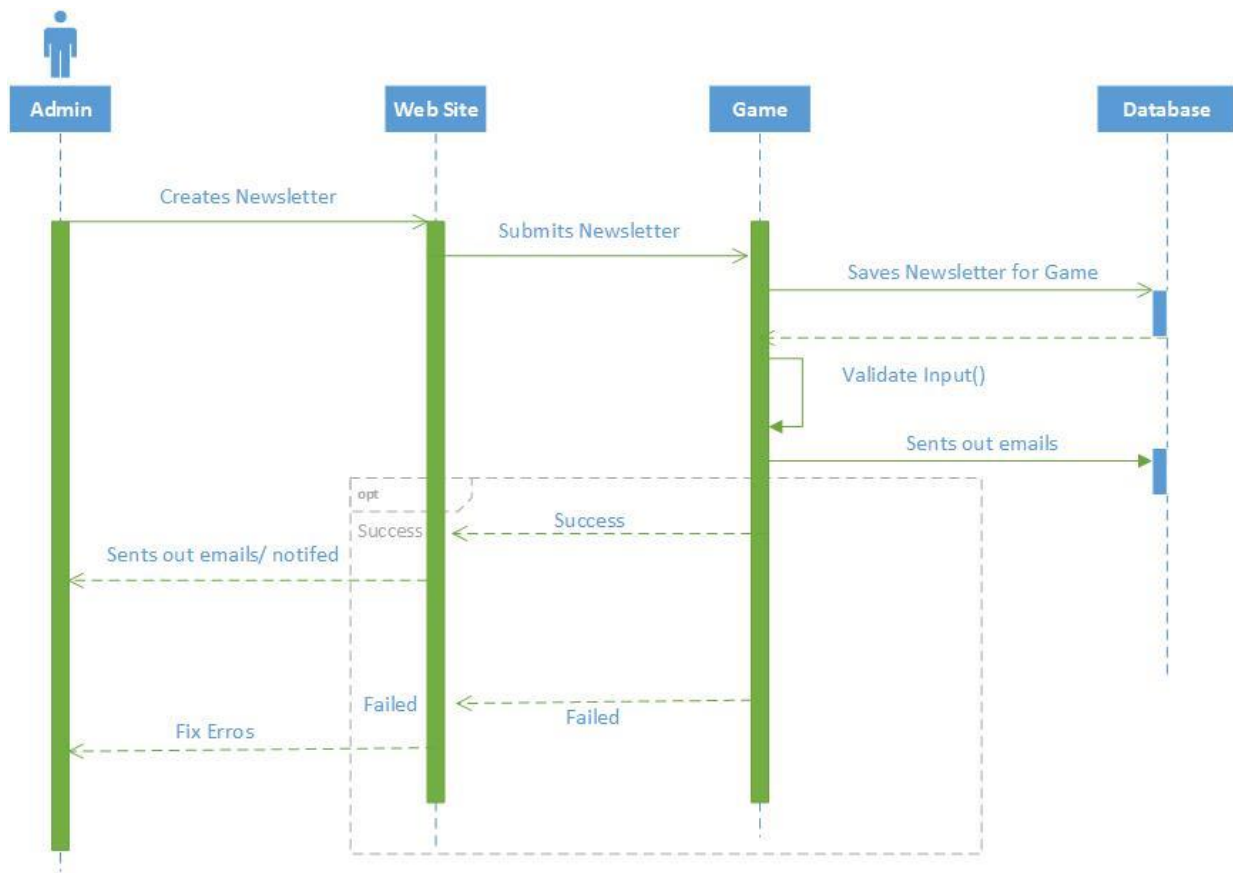


UML 2.0

This diagram describes the process that takes place in creating the Newsletter. The admin will create the Newsletter and submit the Newsletter upon completion.

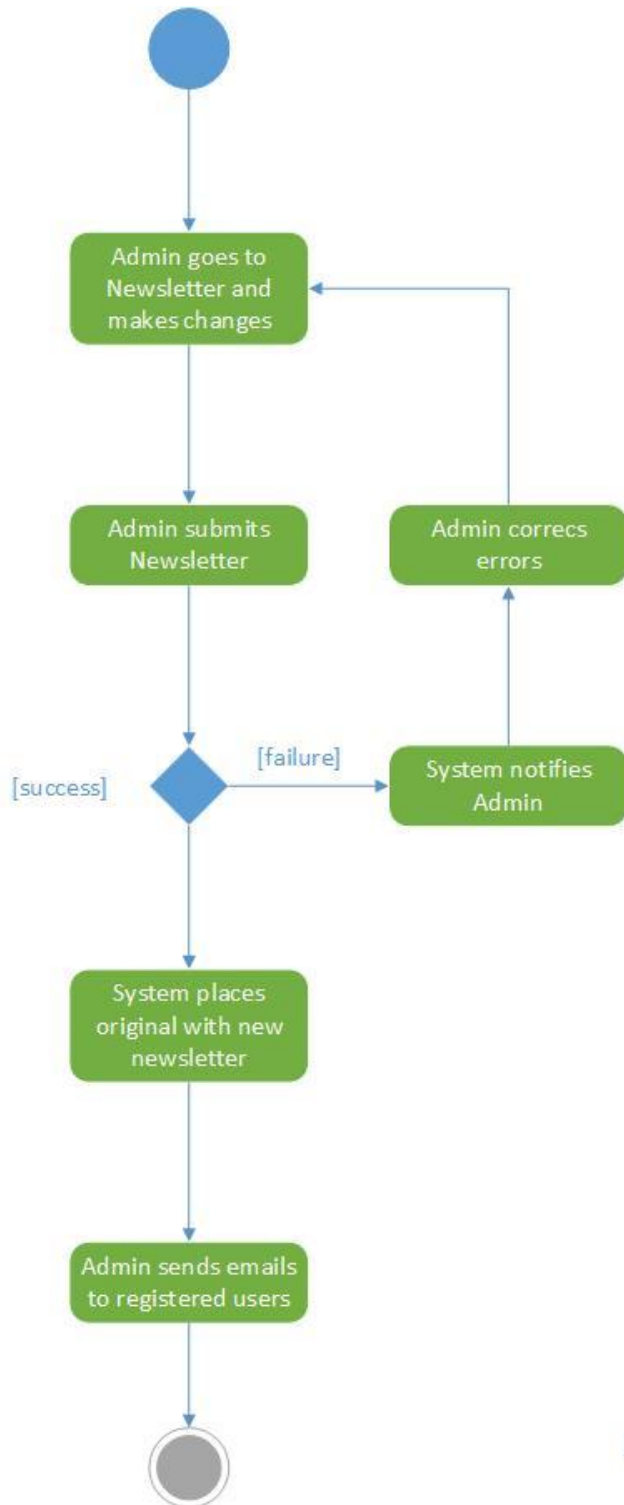
If the submission is successful, the admin will be able to dispense the newsletters to the subscriber list.

If the submission fails, the system will notify the admin, and the admin will fix the required errors. Upon completion, the admin will resubmit the Newsletter.



UML 2.0

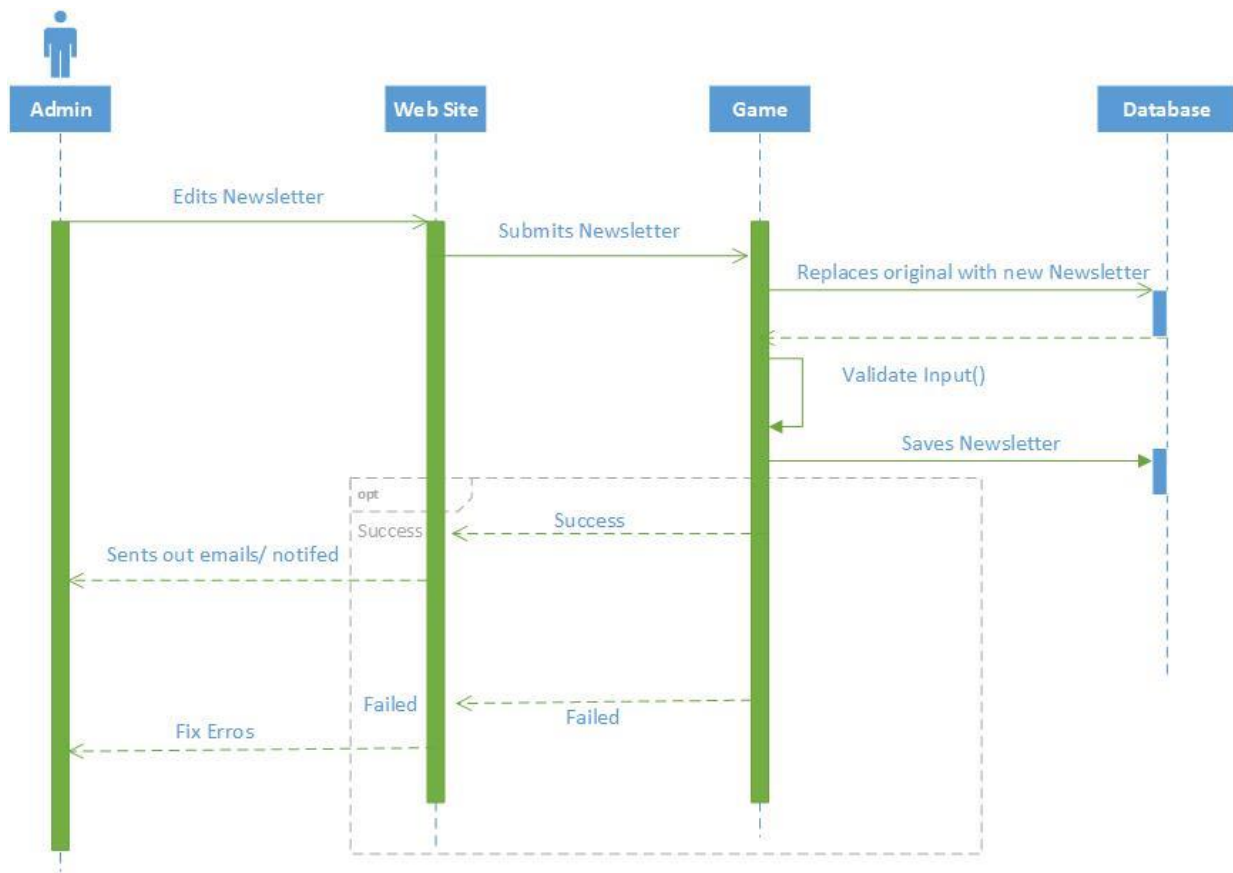
NEWSLETTER: EDIT NEWSLETTER



This diagram describes the process that takes place in editing the newsletter. The admin will go to the desired newsletter and make the changes needed. The admin will then submit the Newsletter.

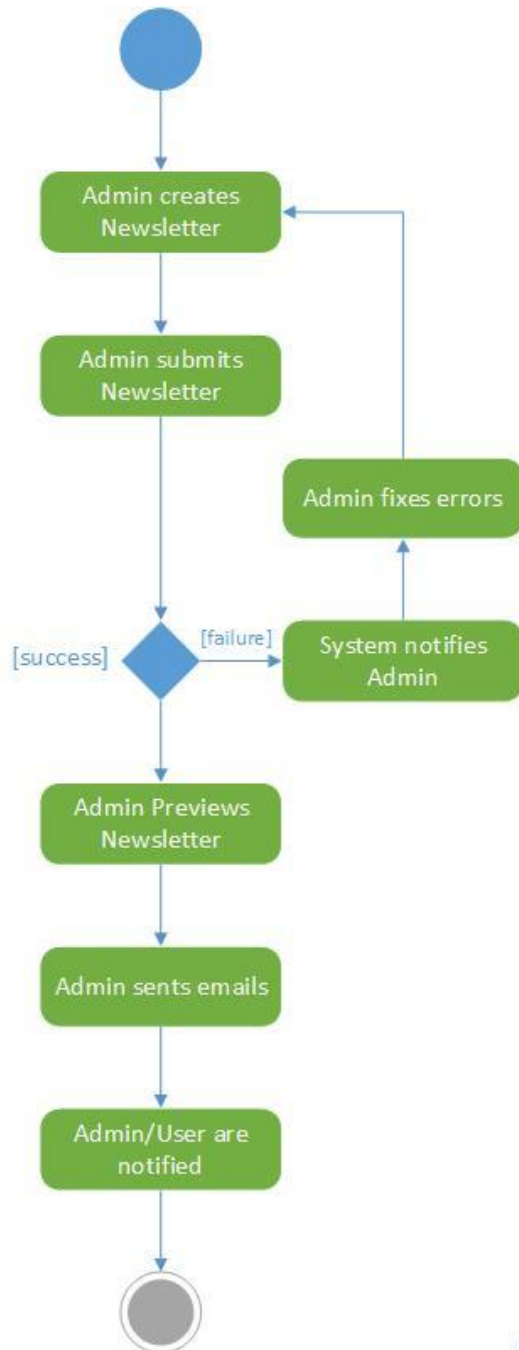
If the submission is successful, the system will place the original Newsletter with the new Newsletter. Thereafter, the admin will send out emails to registered users.

If the submission fails, the system will notify the admin, and the admin will correct the errors. Upon completion, the admin will be able to resubmit the new Newsletter.



UML 2.0

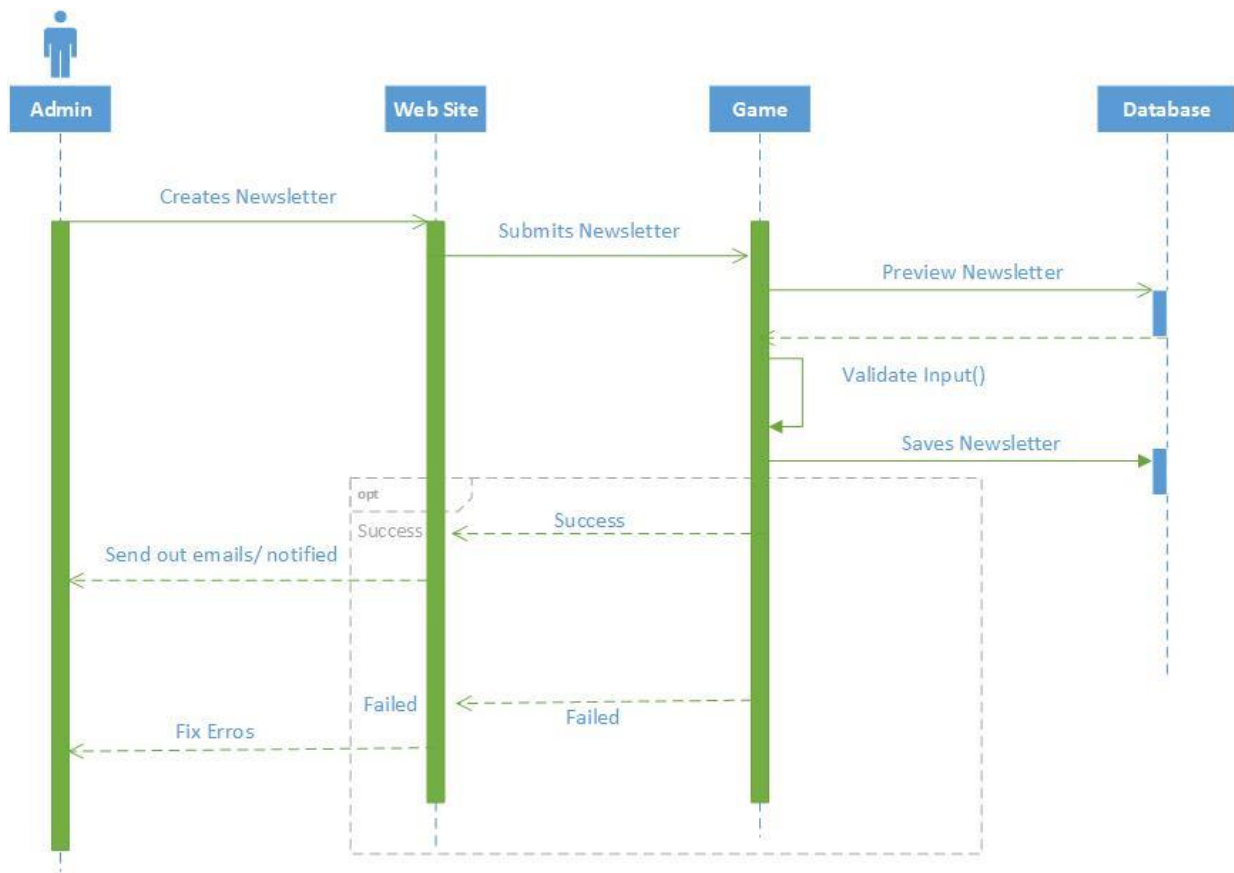
NEWSLETTER: PREVIEW NEWSLETTER



This diagram describes the process that takes place in previewing the Newsletter. The admin will create a Newsletter and submit it.

If the submission is successful, the admin will have the opportunity to preview the Newsletter before sending out emails to registered users.

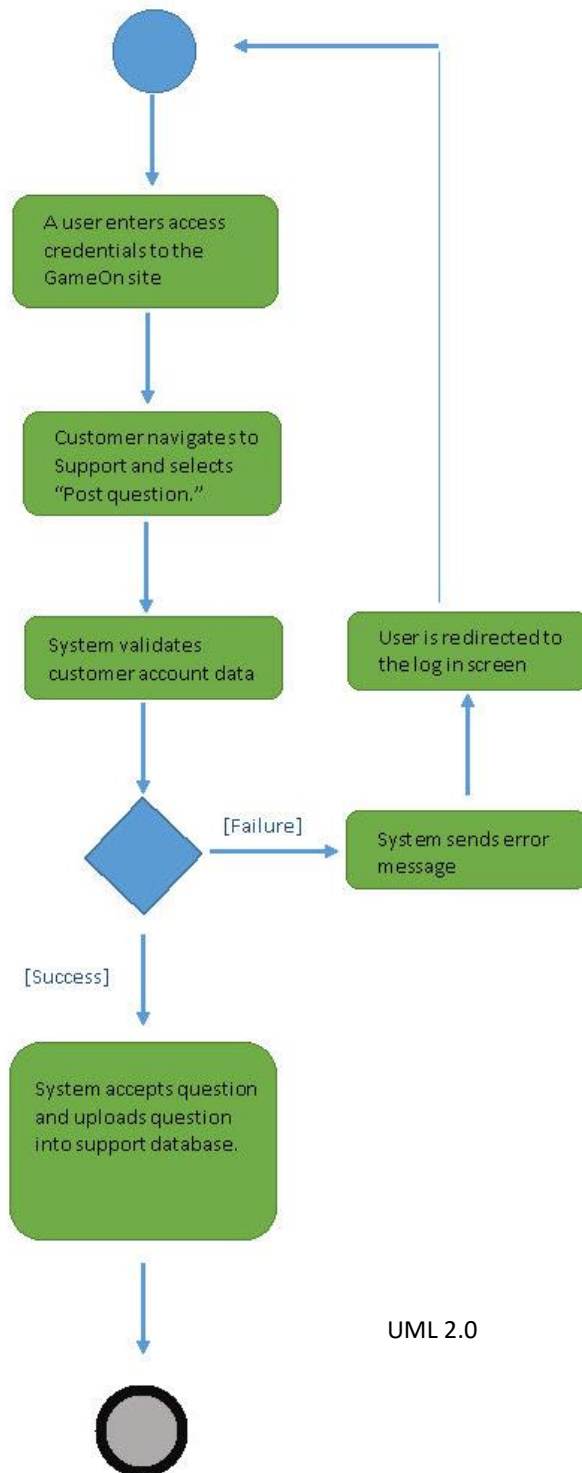
If the submission fails, the system will notify the admin. The admin will fix the errors, and resubmit the Newsletter again.



UML 2.0

MODULE - SUPPORT

SUPPORT: POST QUESTIONS



UML 2.0

This diagram describes the process that takes place for a user to post questions. The customer accesses the support page and submits a new support question. The system will first check that the user is a valid customer.

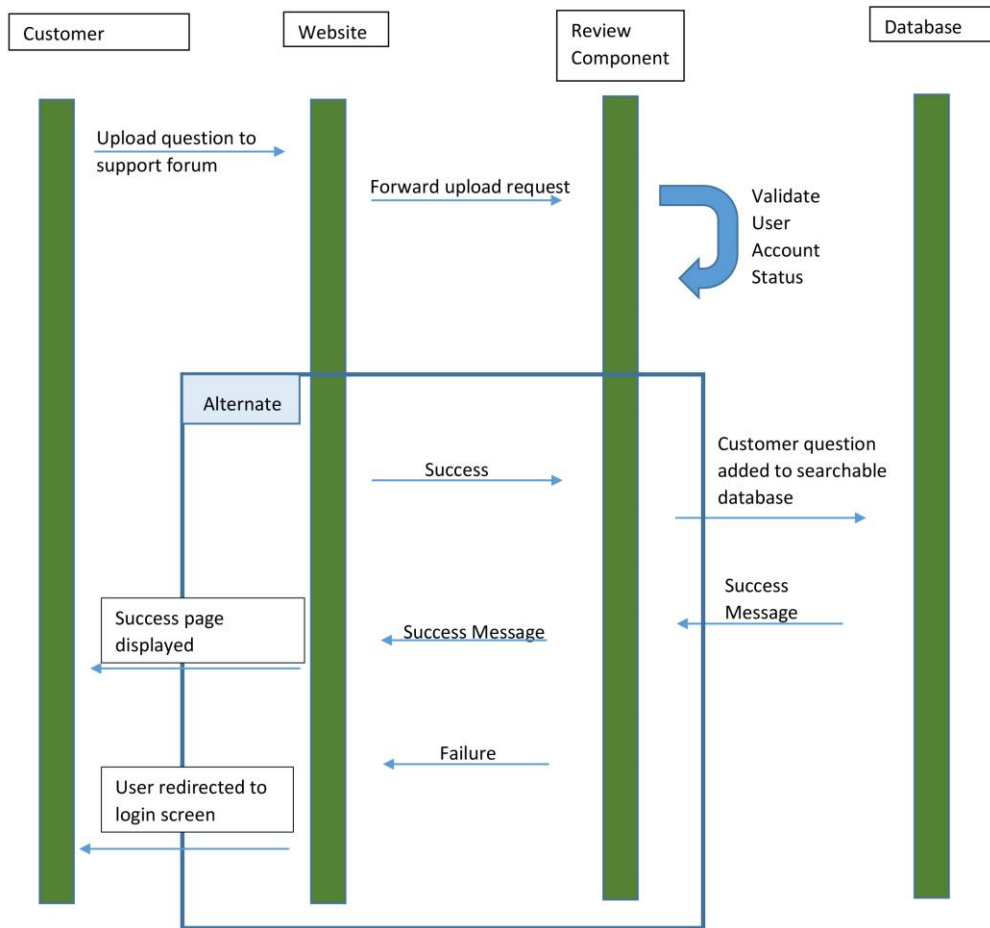
If the system determines an unauthorized user request, the system will return an "unauthorized user" error message with no question posted. The user will be redirected to a login screen.

Upon input of successful credentials, the system will post the user's question into the database and show the question in the support forum.

The same process can be seen in the sequence diagram below. The Web site component forwards all post requests to the review component, which determines eligibility to receive content. The diagram also describes an alternative path for successful and failed transactions.

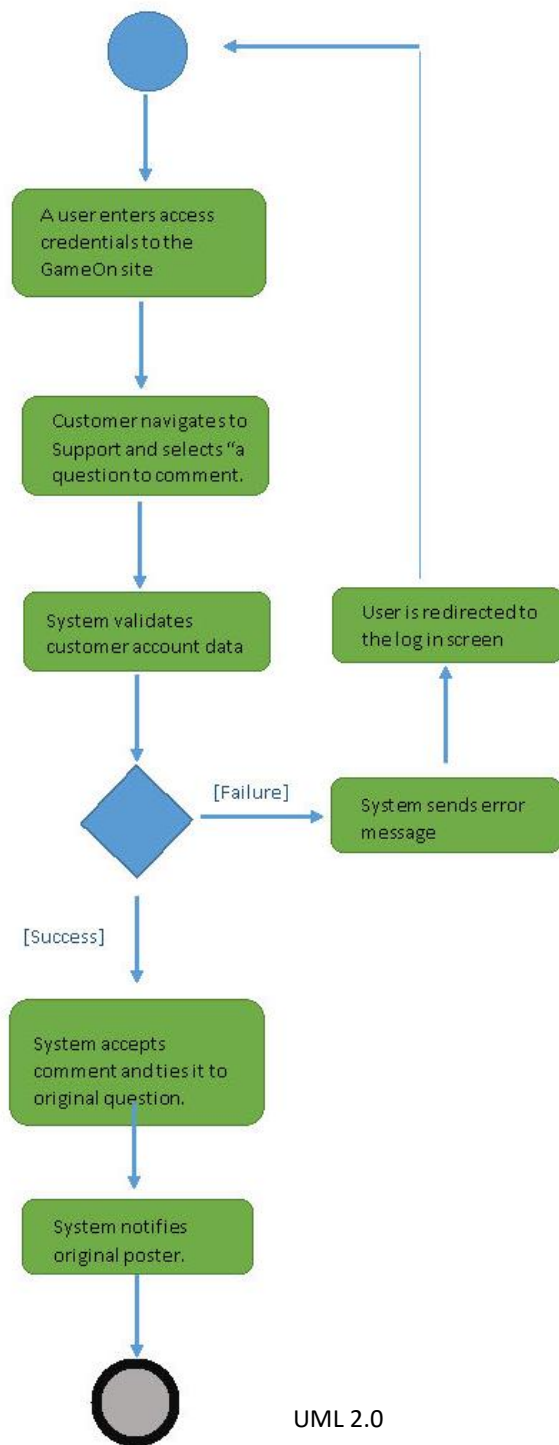
A successful account validation forwards the post to the question database where it is searchable by all users.

A failed account validation posts no information to the data base, and redirects the user to the log in screen.



UML 2.0

SUPPORT:POST COMMENTS ON OTHER’S QUESTIONS



This diagram describes the process that takes place as part of the posting comments onto other user’s questions. A user creates a news article page and inputs any additional definition tag fields relevant. Once the admin submits the news article, the system first checks that the admin is a valid account with proper privileges.

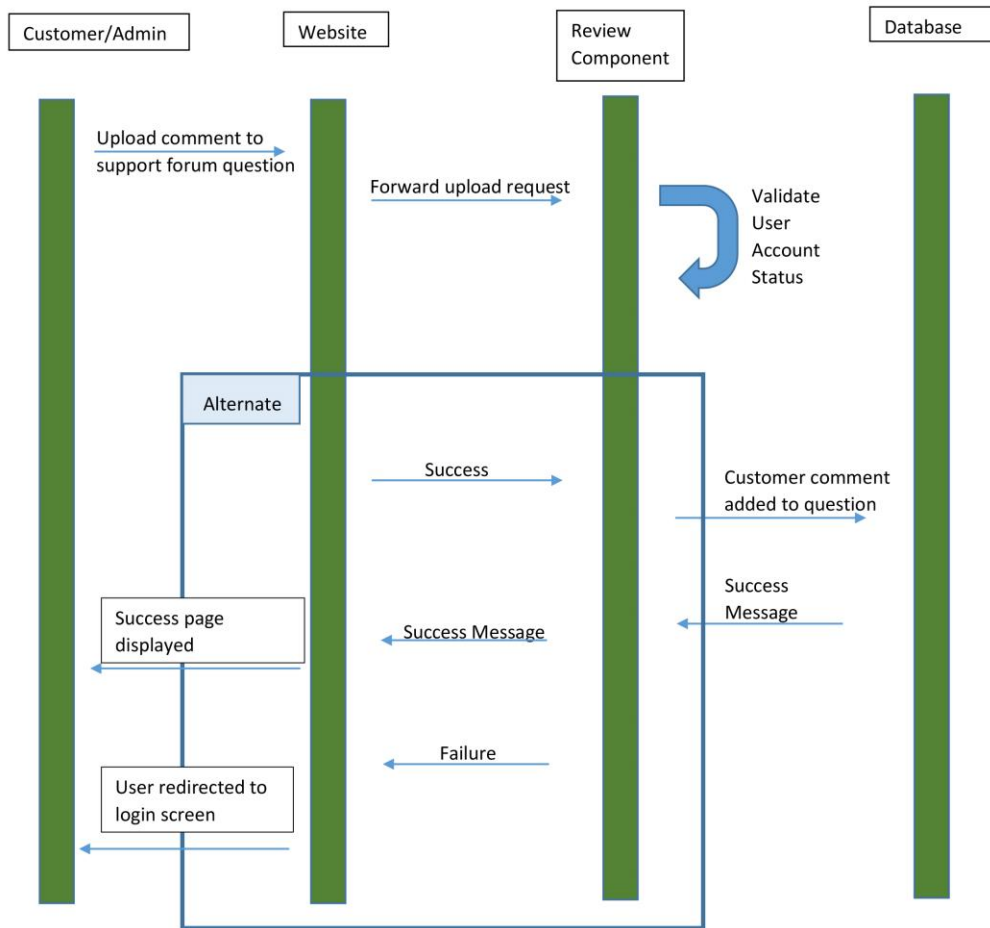
If the system determines an unauthorized upload, the system will return an “unauthorized user” error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will upload news articles, tagged by date and any related tags the admin has defined.

The same process can be seen in the sequence diagram below. The Web site component forwards all news uploads to the review component, which determines eligibility to upload content. The diagram also describes an alternative path for successful and failed uploads.

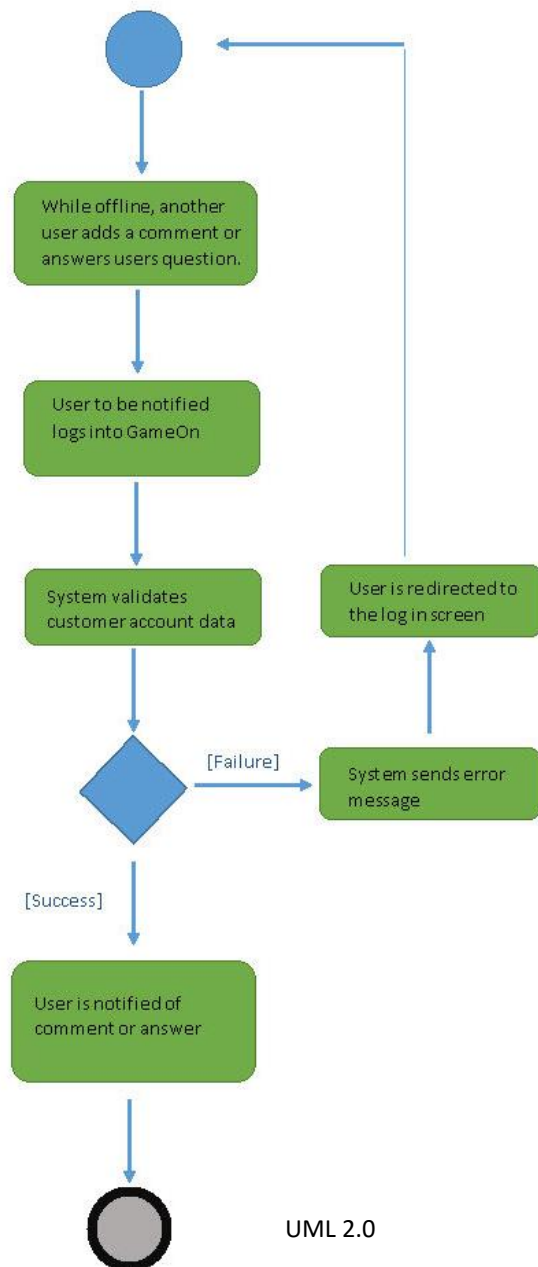
A successful account validation forwards an article to the database. The success message is then returned from the data base and presented to the web site. The web site then forwards the success message to the admin, letting them know the upload has been successful and the content is now live.

A failed account validation uploads no information to the data base, and redirects the user to the log in screen.



UML 2.0

SUPPORT: BE NOTIFIED OF UPDATES



This diagram describes what happens when an update has occurred and a notification is triggered.

An admin will enter their credentials and log into the GameOn site.

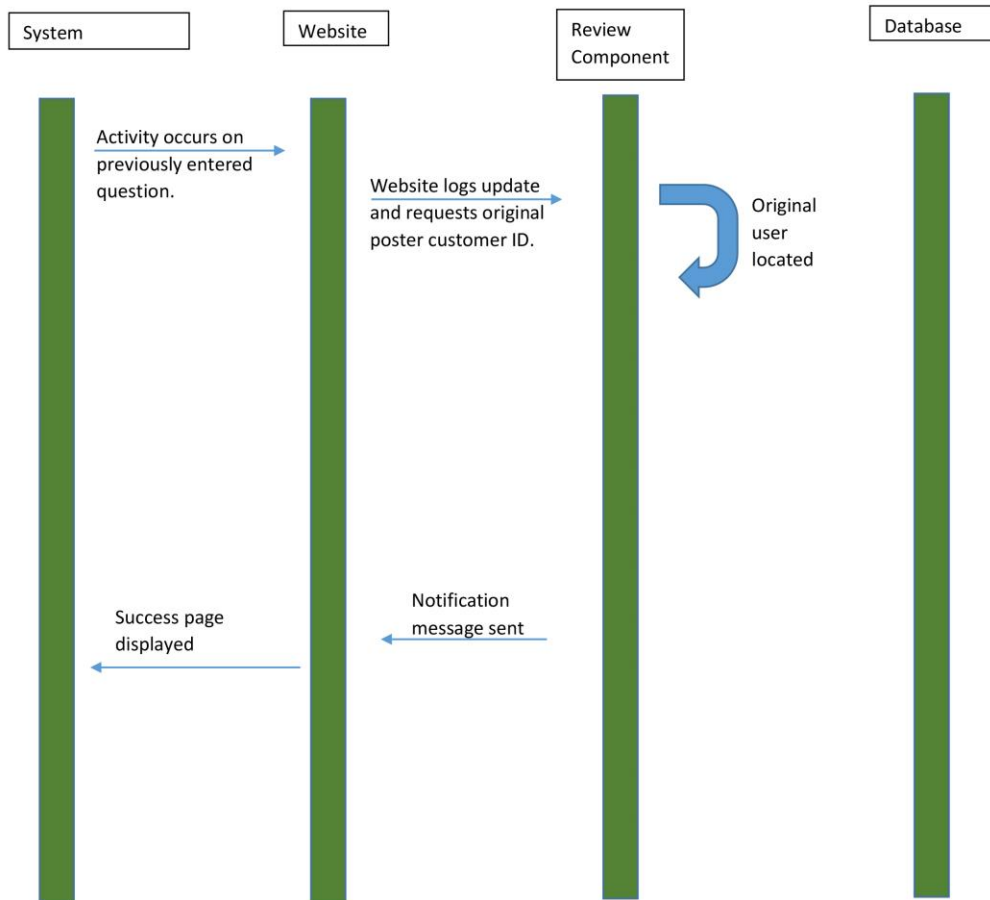
If the system determines an unauthorized edit request, the system will return an "unauthorized user" error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will upload all news articles updates.

The same process can be seen in the sequence diagram below. The Web site component forwards all edit requests to the review component, which determines eligibility to edit content. The diagram also describes an alternative path for successful and failed transactions.

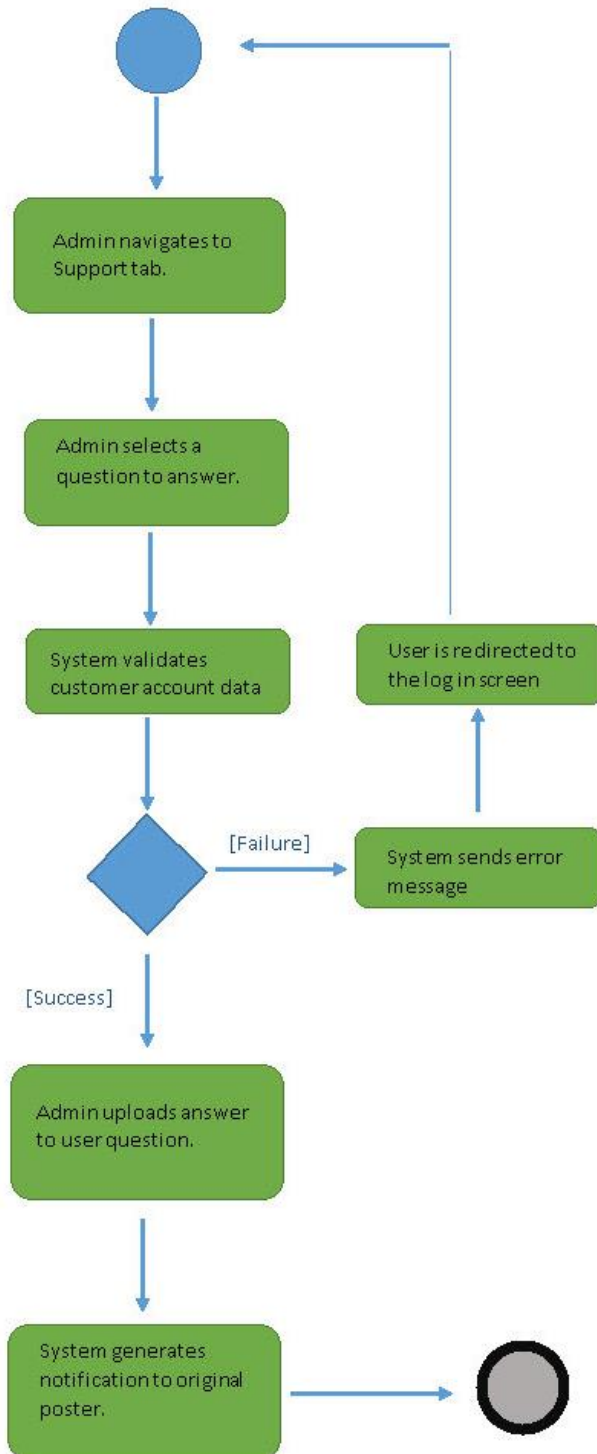
A successful account validation forwards an edit to the database. A success message is then returned from the data base and presented to the web site. The web site then formats the information into a dynamic web page custom to the admin, informing the admin that the edit was successful.

A failed account validation edits no information in the data base, and redirects the user to the log in screen.



UML 2.0

SUPPORT: RESPOND TO QUESTIONS



UML 2.0

This diagram describes the process that takes place when an admin responds to questions. An admin navigates to the support tab. From there, the admin will select a question from the database to answer. Before proceeding with uploading the answer, the system checks that the admin has proper authorization.

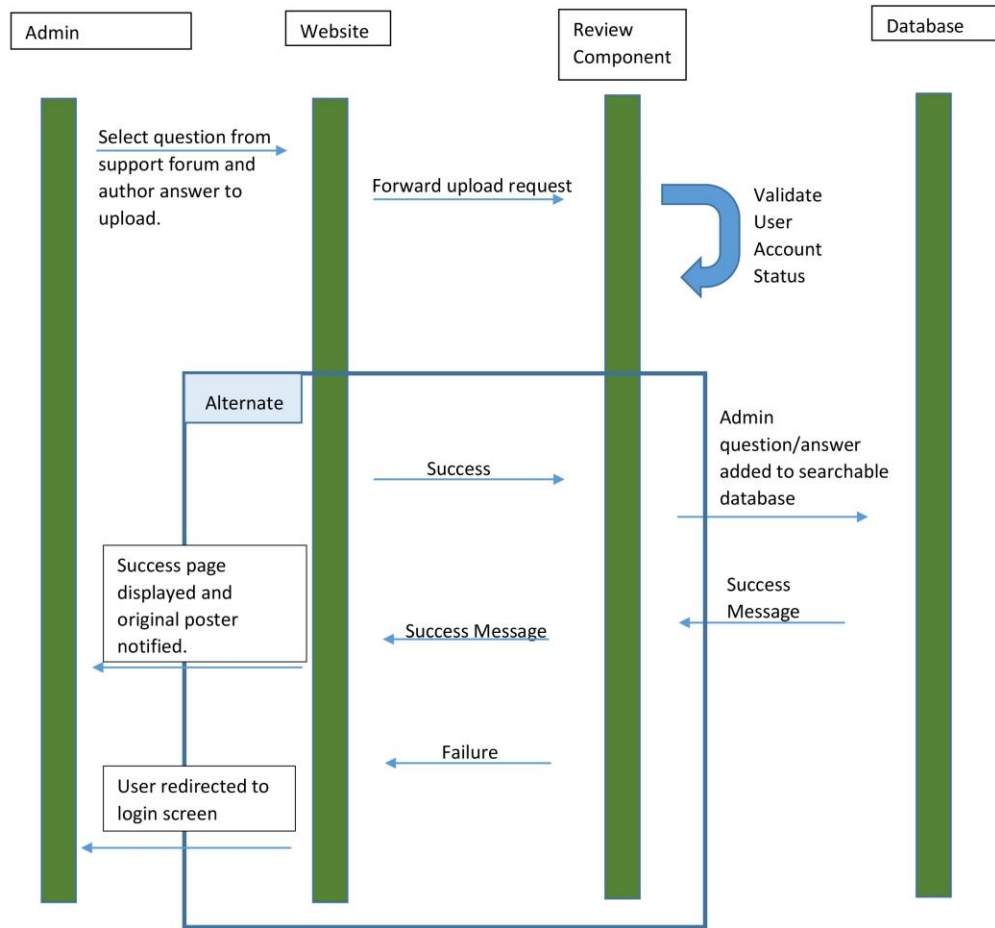
If the system determines an unauthorized answer, the system will return an “unauthorized user” error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will upload the answer to the question a customer has posted. Once an answer has been uploaded, the user will be notified of question activity.

The same process can be seen in the sequence diagram below. The Web site component forwards all edit requests to the review component, which determines eligibility to edit content. The diagram also describes an alternative path for successful and failed transactions.

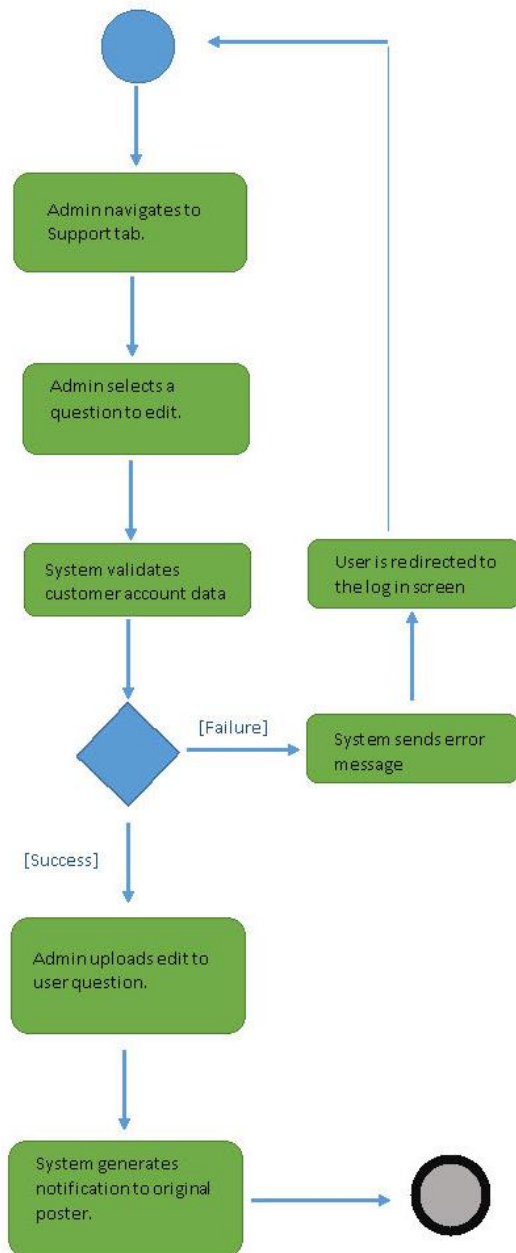
A successful account validation forwards an answer to the database.

A failed account validation adds no information in the data base, and redirects the user to the log in screen.



UML 2.0

SUPPORT: UPDATE QUESTIONS



UML 2.0

This diagram describes the process that takes place as part of the update questions process. As an admin browses questions, occasionally the admin will find errors within questions that need to be addressed. This occurs with the following steps. An admin accesses the support page and selects an article to edit. The system then checks that the admin has proper authorization.

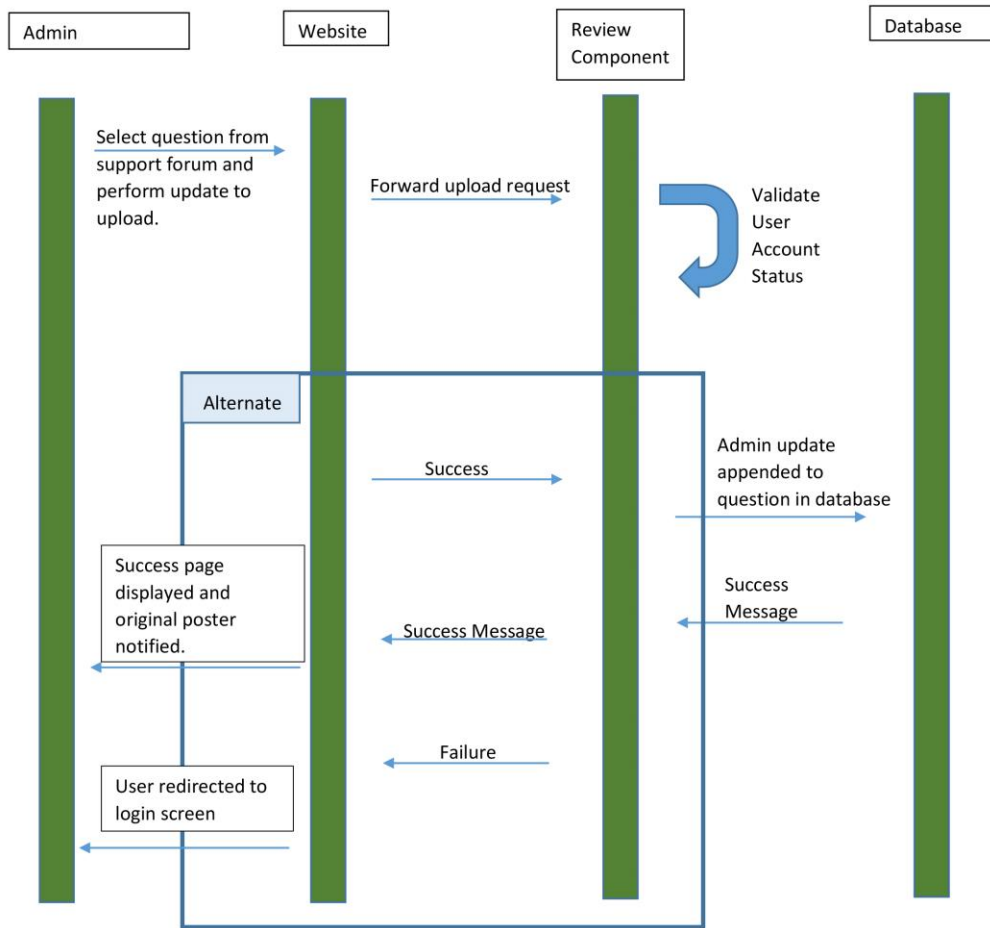
If the system determines an unauthorized edit request, the system will return an “unauthorized user” error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will upload all edits to the database.

The same process can be seen in the sequence diagram below. The Web site component forwards all edit requests to the review component, which determines eligibility to edit content. The diagram also describes an alternative path for successful and failed transactions.

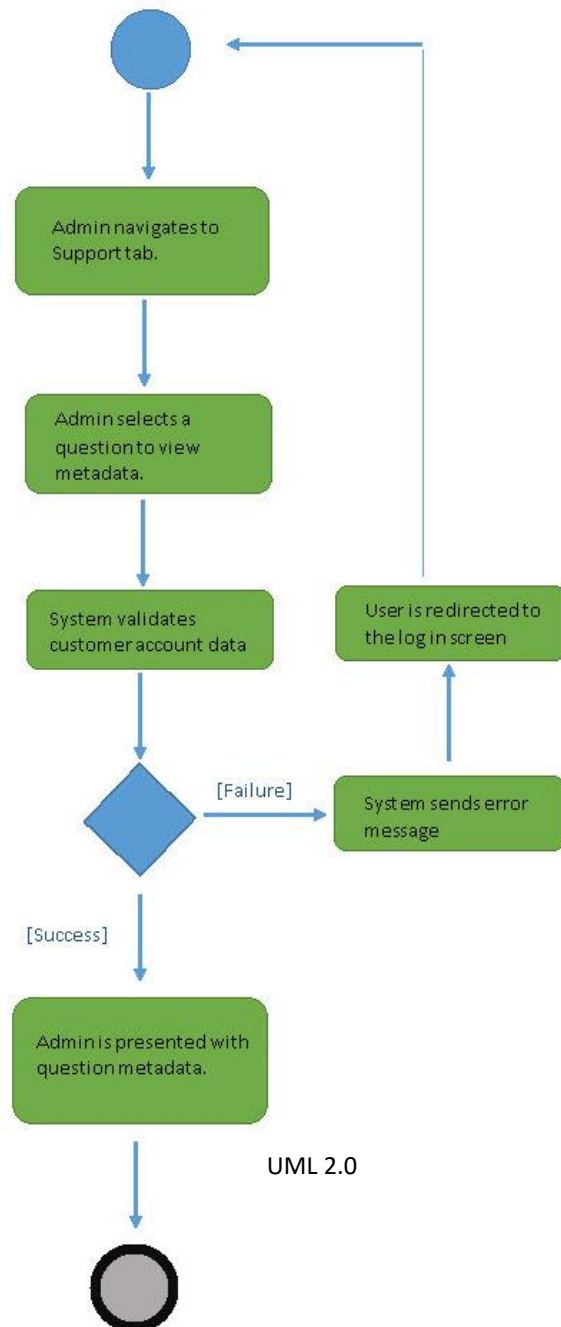
A successful account validation forwards an edit to the database. The website then generates a system notification to inform the original poster of an edit occurrence.

A failed account validation edits no information in the data base, and redirects the user to the log in screen.



UML 2.0

SUPPORT: VIEW QUESTION DETAILS



This diagram describes the process that takes place as part of the view question details process. An admin accesses the support page and selects a question to view details. The system then checks that the admin has proper authorization.

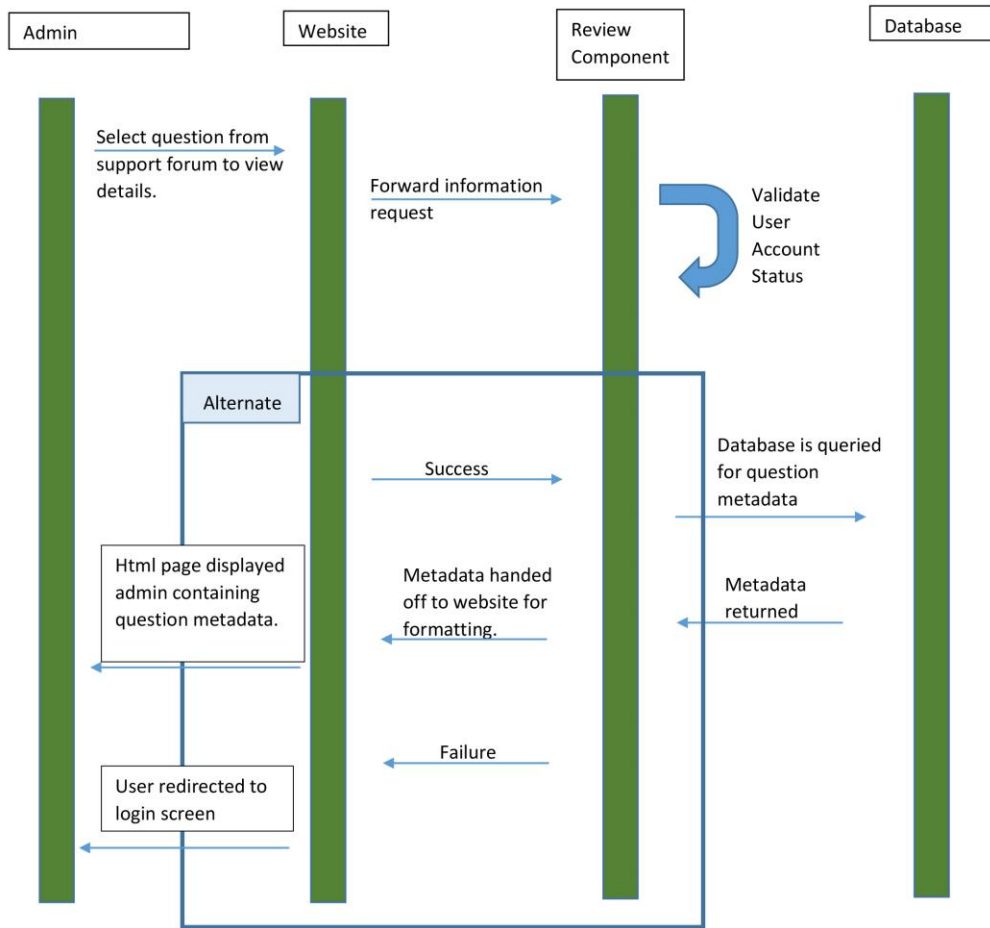
If the system determines an unauthorized detail request, the system will return an “unauthorized user” error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will display the metadata of a question to the admin.

The same process can be seen in the sequence diagram below. The Web site component forwards all edit requests to the review component, which determines eligibility to view content. The diagram also describes an alternative path for successful and failed transactions.

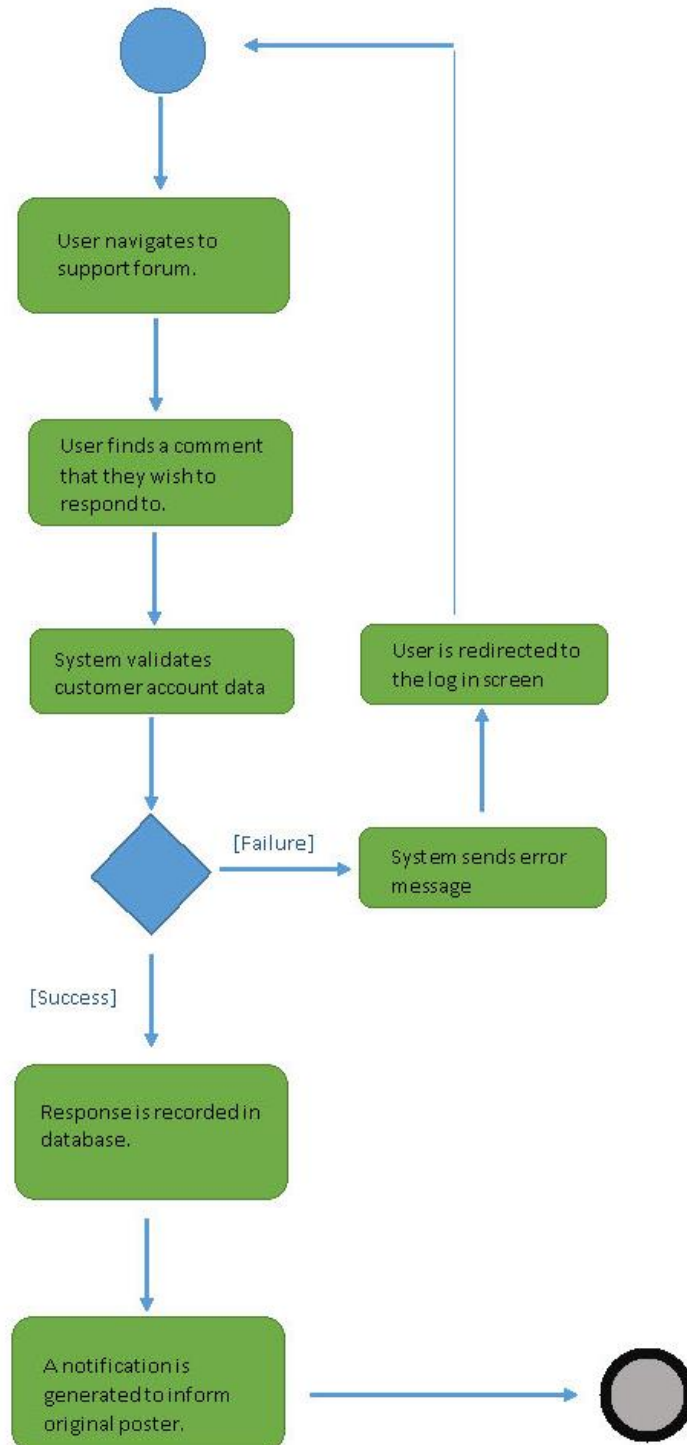
A successful account validation forwards a request to the database. The metadata is then returned to the website for presentation to the admin.

A failed account validation displays no information from the data base, and redirects the user to the log in screen.



UML 2.0

RESPOND TO COMMENTS



UML 2.0

This diagram describes the process that takes place as part of the respond to comments process. A user accesses the support page and selects a comment to respond to. The system then checks that the admin has proper authorization.

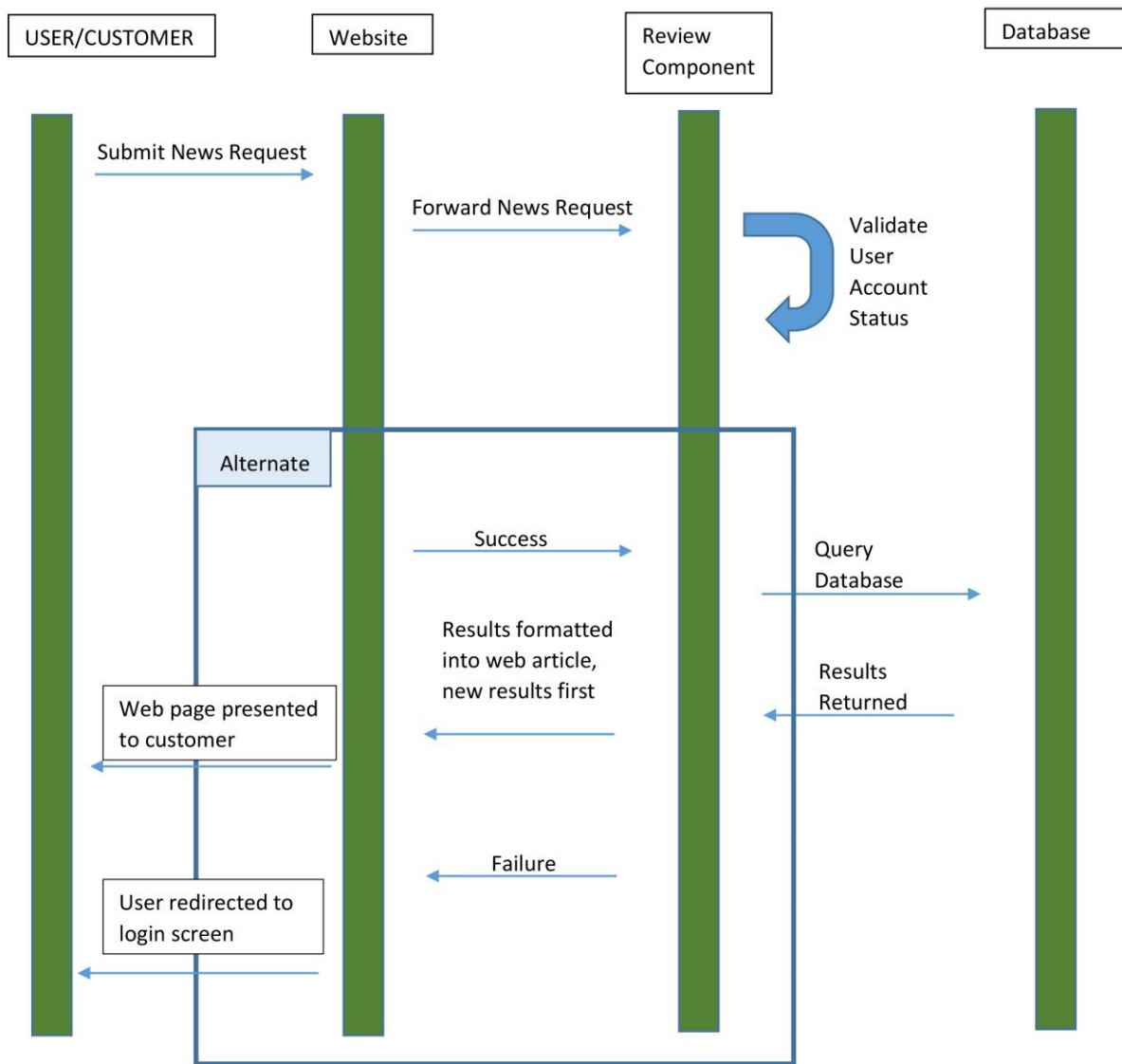
If the system determines an unauthorized response request, the system will return an "unauthorized user" error message. The user will be redirected to a login screen.

Upon input of successful credentials, the system will allow a comment response.

The same process can be seen in the sequence diagram below. The Web site component forwards all edit requests to the review component, which determines eligibility to comment. The diagram also describes an alternative path for successful and failed transactions.

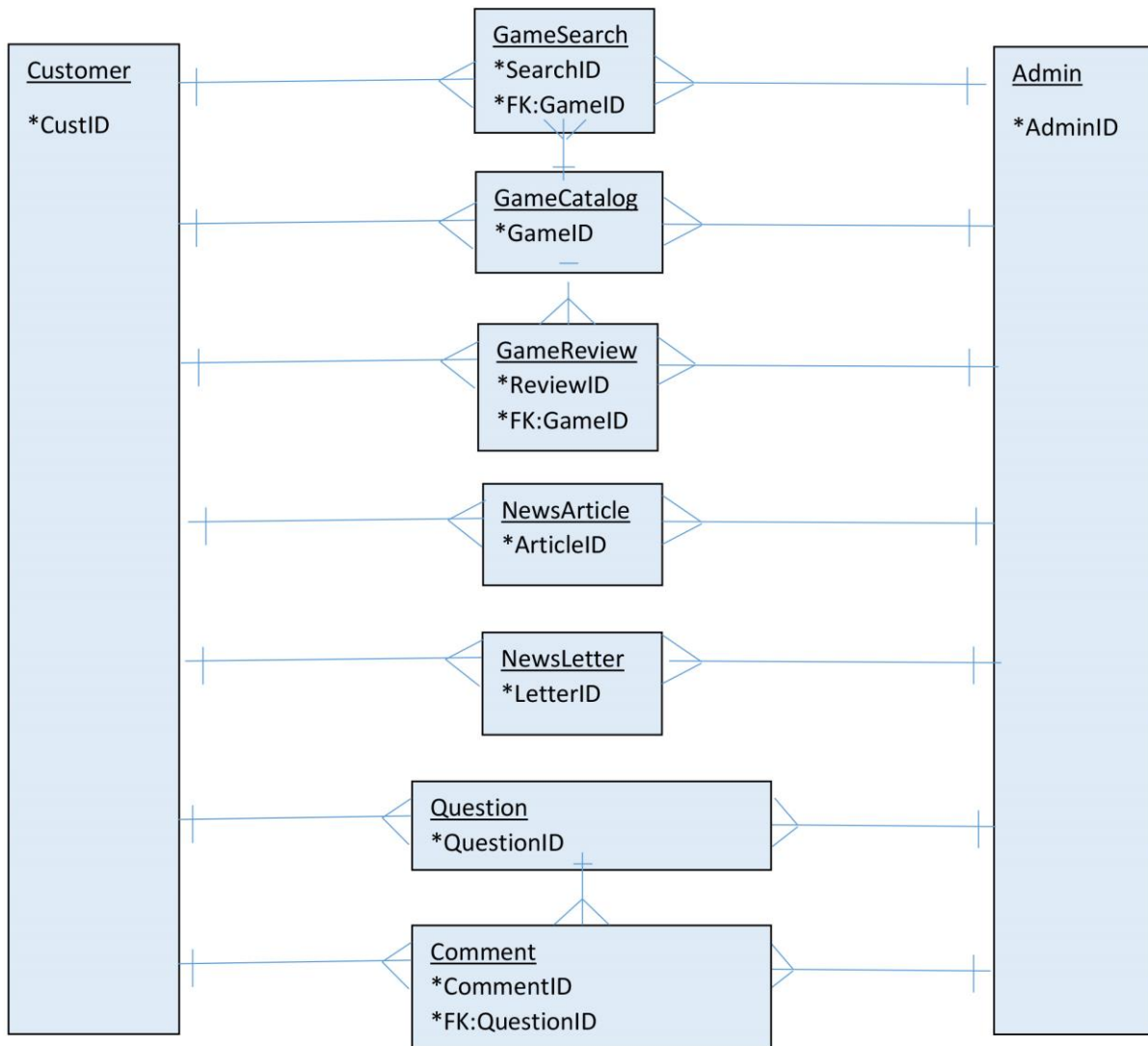
A successful account validation forwards the comment to the database. A success message is then returned from the data base and presented to the web site. This prompts a notification to the original poster informing them that a response has occurred.

A failed account validation edits no information in the data base, and redirects the user to the log in screen.



UML 2.0

DOMAIN ANALYSIS



The class diagram above identifies the main entities for the Registration/Authentication module.

The following entities have been identified:

Customer: This entity represents a single customer. It contains a unique attribute of "CustID."

Admins: Only admins have the ability to create other Administrator type of users. Admin contains the unique attribute "AdminID."

GameCatalog: This database represents all games that have had an entry into the GameOn database. All games have a unique "GameID."

GameSearch: This database is a query to sort all games in GameOn database.

GameReview: This database represents all games that have had an entry into the GameOn database. All game reviews tie to a foreign key "GameID."

NewsArticle: A database of all news articles written about games. Each news article has a unique "ArticleID."

NewsLetter: This database stores all periodic newsletters. Each newsletter contains a unique "LetterID."

Question: Each question is authored by a customer and has a unique "QuestionID."

Comment: All comments have a unique "CommentID," and each comment is tied to a "QuestionID" foreign key.

NON-FUNCTIONAL REQUIREMENTS

Reliability

Our site will strive for reliability and failover by utilizing cloud based internet providers coupled with redundant data storage.

Availability

Due to our cloud based internet provider, our availability strives for 99.99%.

Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to:

- a) Utilize certain crypto-graphical techniques;*
- b) Keep specific log or history data sets;*
- c) Assign certain functions to different modules;*
- d) Restrict communications between some areas of the program;*
- e) Check data integrity for critical variables.*

Maintainability

As we are a cloud based company, at GameOn, all maintenance and security is handled by our third party hosting company. Data redundancy and file accessibility are hallmarks of any hosting corporation.

Portability

As we are a web based service, any operating system or web browsing device should be compatible with our forum.