Software Requirements Specification

for

“Critically Underrated Movies”

Version 1.0 approved

Prepared by Group 8

LUMS

3rd October 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.Introduction

1.1 Purpose

This document aims to provide the software requirements of Critically Underrated Movies. Its objective includes helping the creator, administrator and editor of the software to understand and manage the various functions of the software.

1.2 Document Conventions

This SRS document for Critically Underrated Movies uses the following conventions: to make the document more effective and readable; ‘Arial’ font style is used for paragraph text with font size 11, headings and subheadings are in bold and use the font ‘Times New Roman’.

1.3 Intended Audience and Reading Suggestions

This document is divided into sections to aid the reader in finding the information relevant to them; readers are encouraged to jump to any section they find relevant. Below is a brief overview of each part of the document:

1. Introduction: This section gives an overview of the MovieBase project, including the document’s purpose and its conventions, reading suggestions, project scope and references the document uses.
2. Overall Description: This covers program features, user classes and program constraints.
3. Use Cases and System Features: Specifies the different roles and access rights the users and administrators of the software will have.
4. External Interface Requirements: Describes the hardware, software and communications interfaces this software will deal with.

5. Non Functional Requirements: Lays out the different requirements this software will follow including those related to safety and security among others.
6. Other Requirements: Covers all those details not mentioned in the previous sections.

1.4 Project Scope

Critically Underrated Movies is a website providing movie lovers a platform where they can find information about their desired movies and helps them share and comment on their favourite projects with their social circle. Apart from providing an ideal place to movie watchers to read and share their experience about a popular movie project, it also acts as a tool for movie production houses to gauge their audience's response and feelings regarding their project.

2. Overall Description

2.1 Product Perspective

This software was created to aid users in finding movies that suit their taste and liking. It is designed to be a social media for movies.

2.2 Product Features

Users will be able to create an account, which will let them create their personal lists, leave reviews and interact with other users. Once an account is made, they will be able to edit their profile, where they can write about themselves, the genres they like, etc .They shall be able to send connection requests to other users, which will allow them to share their lists of movies. Moreover, when 2 users are connected, they will also be able to recommend each other specific movies. Apart from this, users will have the ability to leave reviews on movies. Users will also be able to vote on pre-existing reviews, which will make relevant reviews more visible to users. User's whose reviews are well-liked by other users, will also be able to apply for verification as a 'Critic'.

For safety reasons, users will also be able to block other users which will limit their interaction with the blocked user. Moreover, users will have the option to report profiles/reviews which will then be reviewed by the moderator, who will take necessary measures.

2.3 User Classes and Characteristics

- User:

  Our users will compromise of casual movie watchers, and movie fanatics alike. Finding new and exciting movies will be made easier through our platform, where users will have the option of reading past reviews and new recommendations for movies. An average person watches 2 movies per week which will in turn make him a recurring user of our website. Apart from this,

people who watch more movies and want to give their feedback shall use our services almost daily.

- Moderators:

Moderators will be voluntary helpers of the community who shall be passionate about creating such an environment to help other people to make informed decisions regarding the movies they consume. While it is not necessary that they themselves must be movie enthusiasts, they need be people who uphold the terms of service.

- Administrator:

The administrator shall be a person who is up to date about current movie releases and events. They will have the responsibility to regularly update the movie database and also assign moderators.

2.4 Operating Environment

Our service will operate as a web-application which can be accessed by both desktops and mobile. It shall be able to run with the latest versions of google chrome, firefox and other internet browsers.

2.5 Design and Implementation Constraints

There shall be no constraints with respect to its development and use.

2.6 User Documentation

There shall be a help menu where the user will be able to read up on how the site is to be used. The user shall also be given a terms of service agreement at the time of account creation.

2.7 Assumptions and Dependencies

There shall be no externalities that affect the use of our service, and this no assumptions or dependencies are made.

3. Use Cases and System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You must organize this section by use case, and explain each system feature through the Use Case List and Use Case Template table for each use case provided i.e. if you have 15 use cases, there should be 15 use case template tables>

3.1 Use Case ID

3.1.1

<Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X.Y. Related use cases can be grouped in the hierarchy.>

3.2 Use Case Name

<State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

• View part number information.

• Manually mark hypertext source and establish link to target.

• Place an order for a CD with the updated software version>

3.3 Use Case History

3.3.1 Created By

<Supply the name of the person who initially documented this use case.>

3.3.2 Date Created

<Enter the date on which the use case was initially documented.>

3.3.3 Last Updated By

<Supply the name of the person who performed the most recent update to the use case description.>

3.3.4 Date Last Updated

<Enter the date on which the use case was most recently updated.>

3.4 Use Case Definition

3.4.1 Actors

<An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.>

3.4.2 Trigger

<Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.>

3.4.3 Description

<Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.>

3.4.4 Preconditions

<List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. *User's identity has been authenticated.*
2. *User's computer has sufficient free memory available to launch task.>*

3.4.5 Postconditions

<Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. *Document contains only valid SGML tags.*
2. *Price of item in database has been updated with new value.>*

3.4.6 Normal Flow

<Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?"

This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered "X.0", where "X" is the Use Case ID.>

3.4.7 Alternative Flows

<Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the Use Case ID and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow for use case number 5.>

3.4.8Exceptions

<Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form "X.Y.E.Z", where "X" is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, "E" indicates an exception, and "Z" is a sequence number for the exceptions. For example "5.0.E.2" would indicate the second exception for the normal flow for use case number 5.>

3.4.9Includes

<List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.>

3.4.10Priority

<Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.>

3.4.11 Frequency of Use

<Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.>

3.4.12 Business Rules

<List any business rules that influence this use case.>

3.4.13 Special Requirements

<Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.>

3.4.14Assumptions

<List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.>

3.4.15Notes and Issues

<List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.>

Use Case List

Primary Actor	***Use Cases***
Admin	1. Update movie database 2. Appoint moderators 3. Assign moderator score
Moderator	4. Remove user content 5. Limit access of users 6. Grant verification status
User	7. Create profile and Register User 8. Create lists 9. Make recommendations to other users 10. Make reviews on movies 11. Make connections with other users 12. Add personal information 13. Apply for verification 14. Rate reviews 15. Rate movies 16. Share favourite Movie Quotes 17. Block other users 18. Award other users 19. Vote Best Movie of All Time 20. Share Movie of The Day

Use Case Template

Use Case ID:	1		
Use Case Name:	Update movie database		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	Admin
Description:	Admin can determine what movies to add or remove from the database based on certain criteria such as time-relevance, popular opinion, new releases etc
Trigger:	Presses the “add/remove movie” button
Preconditions:	- Currently logged in as admin
Postconditions:	- Movie is either added or removed from database based on selected option
Normal Flow:	1. Enters admin login information 2. System verifies admin status 3. Admin selects “add/remove movie” 4. System displays data entry form 5. Admin enters movie details 6. System adds movie to the database
Alternative Flows:	1. If the Admin wants to delete the movie 2. Admin searches for movie_title

	3. System displays movie results 4. Admin selects delete button 5. System removes movie from database
Exceptions:	Admin attempts to add existing movie - Admin adds a movie that already exists in the database - System displays error message saying “Movie already exists”
Includes:	- Authenticates Admin identity - Verifies entered information with database
Priority:	High
Frequency of Use:	After initial database entry, once every few months to update new movie titles as they are released
Business Rules:	None
Special Requirements:	Admin can cancel data entry at any point after selecting “Add/remove movie”
Assumptions:	System displays/inserts movie within a few seconds
Notes and Issues:	

Use Case ID:	2

Use Case Name:	Appoint Moderators		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	Admin
Description:	Administrators will have the ability to appoint and remove moderators from the website
Trigger:	Presses add/remove moderator button
Preconditions:	- Users to add/remove must pre-exist in the system - User must be logged in as administrator
Postconditions:	- There will be an updated list of moderators
Normal Flow:	**Moderators Menu** 1. User logs in as Administrator 2. System verifies and shows the main menu to Administrator 3. Administrator clicks on 'Current Moderators' button 4. System shows current list of moderators 5. Administrator clicks on 'Add' button. 6. System asks for user id to add. 7. Administrator enters user id. 8. System shows a confirmation message to administrator. 9. Administrator confirms selection. 10. System updates the list and shows the list of current moderators.
Alternative Flows:	**Moderators Menu** 11. User logs in as Administrator

	12. System verifies and shows the main menu to Administrator 13. Administrator clicks on ‘Current Moderators’ button 14. System shows current list of moderators 15. Administrator clicks on a moderator to remove him. 16. System shows a confirmation message to the administrator. 17. Administrator confirms selection. 18. System shows updates the list and shows the list of current moderators.
Exceptions:	- Administrator tries to add a user that doesn't exist - System shows error
Includes:	- Authenticates Admin identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- After appointing initial moderators, changes to moderators will be infrequent
Business Rules:	- None
Special Requirements:	- Admin can cancel the change after clicking the add or remove button
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	3

Use Case Name:	Assign Moderator Score		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	Admin
Description:	Administrators will have the ability to score the performance of moderators on the website, so that the moderators know how well they are moderating
Trigger:	Presses moderator score button
Preconditions:	- Users to add/remove must pre-exist in the system - User must be logged in as administrator
Postconditions:	- There will be an updated score of moderators
Normal Flow:	**Moderators Menu** 11. User logs in as Administrator 12. System verifies and shows the main menu to Administrator 13. Administrator clicks on ‘Current Moderators’ button 14. System shows current list of moderators 15. Administrator clicks on ‘Score’ button. 16. System shows score scale from 1 to 10. 17. Administrator selects moderator score. 18. System shows a confirmation message to administrator. 19. Administrator confirms selection. 20. System updates the list and shows the updated score of moderator(s).
Alternative Flows:	**Moderators Menu** 11. User logs in as Administrator

	19. System verifies and shows the main menu to Administrator 20. Administrator clicks on ‘Current Moderators’ button 21. System shows current list of moderators 22. Administrator clicks on a moderator to score. 23. System shows a confirmation message to the administrator. 24. Administrator confirms selection. 25. System shows updates the list and shows the list of current moderators.
Exceptions:	- Administrator tries to add a user that doesn't exist - System shows error
Includes:	- Authenticates Admin identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- After appointing initial moderators, changes to moderators will be infrequent
Business Rules:	- None
Special Requirements:	- Admin can cancel the change after clicking the score button
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	4

Use Case Name:	Remove user content		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	Moderator
Description:	If a moderator determines that a user has posted content that goes against community standards, they can remove the content
Trigger:	Moderator presses “Remove content” button
Preconditions:	- Moderator must be logged into the system as a moderator
Postconditions:	- The content is removed from the site
Normal Flow:	1. Moderator enters login information 2. System verifies moderator status and logs in 3. Moderator finds offensive content from content window and presses “Remove content” button 4. System displays alert window confirming the action 5. Moderator selects “Yes” 6. System removes content from site
Alternative Flows:	In step 4 1. Moderator selects “No” 2. System returns to content window

Exceptions:	None
Includes:	- Authenticate moderators identity
Priority:	High
Frequency of Use:	Fairly often, a few times a week
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	5		
Use Case Name:	Limit user access		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	Moderator
Description:	If a moderator determines that a user has posted content that goes against community standards, they can restrict the user from voting and/or leaving or permanently banning the user's account
Trigger:	Moderator presses "Users access" button
Preconditions:	- Moderator must be logged into the system as a moderator
Postconditions:	- The specified user is no longer allowed to leave reviews - The specified user is no longer allowed to vote

	- The specified user is permanently banned
Normal Flow:	7. Moderator enters login information 8. System verifies moderator status and logs in 9. Moderator clicks on ‘Users’ button 10. System displays list of registered users 11. Moderator clicks on specified user 12. System shows user granted permissions (ability to vote, ability to leave review) and a ‘Delete user’ button 13. Moderator unchecks either one (or both) of the boxes 14. System alerts the moderator to confirm his changes 15. Moderator confirms 16. System updates the user’s privileges.
Alternative Flows:	Moderator enters login information 17. System verifies moderator status and logs in 18. Moderator clicks on ‘Users’ button 19. System displays list of registered users 20. Moderator clicks on specified user 21. System shows user granted permissions (ability to vote, ability to leave review) and a ‘Ban user’ button 22. Moderator clicks on ‘Ban user’ 23. System alerts the moderator to confirm his changes 24. Moderator confirms 25. System updates the current list of allowed users
Exceptions:	
Includes:	- Authenticate moderators identity
Priority:	- High
Frequency of Use:	- Fairly often, couple of times a week
Business Rules:	None

Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	6		
Use Case Name:	Grant verification access		
Created By:	Hamza	Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	Moderator
Description:	Moderators can grant verification status to those users that have applied for verification and have the credentials to be verified
Trigger:	Presses “Verify” button
Preconditions:	- Moderator must be logged into the system
Postconditions:	- User will be successfully verified
Normal Flow:	1. Moderator enters login credentials in the login screen 2. System verifies the credentials and logs in 3. Moderator goes to “Verification Application” tab 4. System will display the tab 5. Mod will search in tab 6. System will show the list of users matching the search 7. User will select the user to verify 8. System will display an alert window to confirm verification 9. Mod will verify the user

	10. The blocked user can not recommend the movies 11. The blocked user can not send the friend request
Alternative Flows:	1. In step 6, User will select the user 2. System will display that user is blocked 3. System will also show an unblock button 4. User will press the button 5. System will unblock the user 6. Now, both can send requests and recommend movies
Exceptions:	If the searched user does not exist, system will take the Mod back to main
Includes:	- Authenticates Mod identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- Mod will use it frequently to verify users
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	7		
Use Case Name:	Create profile and Register User		
Created By:	Tehzeer	Last Updated By:	

Date Created:	3rd October 2021	Date Last Updated:	

Actors:	User
Description:	Users will have the ability to create an account, add about me section, add list, leave reviews.
Trigger:	- Clicks on ‘Sign up’ button
Preconditions:	- Users must have a valid username and email. - Users must have a strong password
Postconditions:	- Users will be able to log in to their account. - Users will have access to multiple functions which are available for all users.
Normal Flow:	1. User clicks on sign-up button 2. System will take them to a page where they will add their credentials 3. User will confirm their credentials 4. System will send them a verification email 5. User will confirm his account through email 6. System will update their credentials in database
Alternative Flows:	IN STEP 3, 7. User enters username which pre-exist 8. System will give a warning about ‘user name already exist’ 9. User enters invalid emails 10. System will give a warning about email doesn’t exist 11. User enters a valid email, but weak password 12. System will give a warning ‘password is weak’
Exceptions:	- If the user enters wrong credential, account won’t be created

Includes:	- Unique username - Verified email - Strong password (not username itself, and some common patterns)
Priority:	- High
Frequency of Use:	- Only once
Business Rules:	- None
Special Requirements:	- Admin can remove the user from the website
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	8		
Use Case Name:	Create Lists		
Created By:	Shaheer	Last Updated By:	
Date Created:	3rd October 2021	Date Last Updated:	

Actors:	User
Description:	Users can create lists of movies that they like.
Trigger:	User presses “Create List” button
Preconditions:	- User must be logged into the system
Postconditions:	- The movie is added to the user’s list of liked movies
Normal Flow:	1. User enters login information 2. System verifies user status and logs in 3. User presses “Create list” button 4. System displays data form to add list name, description, movies etc. 5. User presses “Done” button 6. System saves the user’s list in the database.
Alternative Flows:	7. User finds a movie and presses “Add to my list” button. 8. System displays existing lists to which the movie can be added to. 9. User selects the list which he wants the movie to be added to. 10. System adds movie to the user’s list.
Exceptions:	- The user presses “Add to my list” button for a movie that already exists in the user’s list - System display message “Movie already exists in this list”
Includes:	- Authenticates User identity - Verifies entered information with database

Priority:	High
Frequency of Use:	Used on a daily basis by the user.
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	9		
Use Case Name:	Make recommendations to other users		
Created By:	Shaheer	Last Updated By:	
Date Created:	3rd October 2021	Date Last Updated:	

Actors:	User

Description:	Users can recommend movies to their social circle.
Trigger:	User presses “Recommend” button
Preconditions:	- User must be logged into the system - User must be friends with the person he is recommending the movie to.
Postconditions:	- The movie is shown to the person who it is recommended to by the user
Normal Flow:	1. User enters login information 2. System verifies user status and logs in 3. User presses “Recommend” button 4. System displays user’s friends to select who to recommend to 5. User selects one or more friend to recommend the movie to 6. User presses “Done” button 7. The system notifies the user’s friend that he is being recommended the movie by that user
Alternative Flows:	7. User recommends the same movie to one person multiple times 8. System displays messages “Movie already recommended to the user, recommend again?” 9. User selects presses “yes” or “no” button. 10. System re-notifies the user’s friend or closes the system message depending on the user’s selection.
Exceptions:	- The user recommends movies to a friend that is not old enough to watch the movie. - System display message “Movie is not appropriate as your friend is under age”

Includes:	- Authenticates User identity - Verifies entered information with database
Priority:	High
Frequency of Use:	Used on a daily basis by the user.
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	10		
Use Case Name:	Make reviews on movies		
Created By:	Shaheer	Last Updated By:	
Date Created:	3rd October 2021	Date Last Updated:	

Actors:	User

Description:	Users can review movies that they have watched.
Trigger:	User presses “Review” button
Preconditions:	- User must be logged into the system
Postconditions:	- The review is added to the particular movie’s profile
Normal Flow:	1. User enters login information 2. System verifies user status and logs in 3. User presses “review” button 4. System displays data form to type out review. 5. User types out their review 6. User presses “Done” button 7. System saves the user’s review in the database.
Alternative Flows:	8. User presses “review” button on a movie that they have already reviewed. 8. System displays existing review and allows user to edit or delete their review. 9. User makes edits or presses “delete” button. 11.System saves the updated review or removes the review from the database depending on the user’s selection.
Exceptions:	None
Includes:	- Authenticates User identity - Verifies entered information with database

Priority:	High
Frequency of Use:	Used on a daily basis by the user.
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	11		
Use Case Name:	Make connections with users		
Created By:	Hassan	Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users are able to send users connection requests and consequently accept user connection requests as well. When 2 people are connections, they will be able to view each others' lists and send each other movie recommendations.
Trigger:	User presses "Send Connection Request" button
Preconditions:	- User must be logged into the system - User must not have blocked the other user, or be blocked by the other user
Postconditions:	- The user has sent a connection request - The user has accepted/declined a connection request
Normal Flow:	1. User enters login information 2. System verifies the user and logs in 3. User presses 'Make Connection' button 4. System asks user to input user id 5. User clicks 'Send connection request' 6. System sends a connection request to specified user.
Alternative Flows:	7. User enters login information 8. System verifies the user and logs in 9. User presses 'Connection Requests' button 10. System shows list of pending connection requests 11. User clicks on either accept/decline. 12. System confirms the selection and updates the user's connections list.
Exceptions:	User to send connection request to does not exist
Includes:	- Authenticates User identity - Verifies entered information with database

Priority:	High
Frequency of Use:	Used on a daily basis by the user.
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	12		
Use Case Name:	Add personal information		
Created By:	Tehzeer	Last Updated By:	
Date Created:	3rd October 2021	Date Last Updated:	

Actors:	User
Description:	Users can create their personal profile where they can add a little bit about their personal suggestions, favourite movies, genres they like and about their personalities.

Trigger:	User have to press ‘Add Personal Information’ button
Preconditions:	- Users must have a profile on the website. - Users must be logged in first.
Postconditions:	- User’s personal information will be updated on their profile and database.
Normal Flow:	1. User logged in with id and password 2. System confirms their credentials from database 3. User will go to their profile 4. System will take them to their profile 5. User clicks on add personal information 6. System will allow them to edit that portion 7. User will add their personal information and saves it 8. System will update that information in the database and display that under their profile.
Alternative Flows:	- None
Exceptions:	- None
Includes:	- Authenticate user’s identity
Priority:	- Medium
Frequency of Use:	- After initial; use, it will be used infrequently
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None

Notes and Issues:	- Users must not add inappropriate information which can be harmful.

Use Case ID:	13		
Use Case Name:	Apply for verification		
Created By:	Tehzeer	Last Updated By:	
Date Created:	3rd October 2021	Date Last Updated:	

Actors:	User
Description:	When users will have certain amount of votes, their profile will be updated to a verified account and their votes and suggestions will be shown on top
Trigger:	- Clicks on ‘Apply for verification’ button
Preconditions:	- Users must have a profile. - Users must be logged in. - Users must have more than a certain number of votes.
Postconditions:	- User’s profile will be shown on top. - User’s suggestions will be public.
Normal Flow:	1. User enters login information 2. System logged them in after confirming it from database 3. User goes to their profile 4. System will take them to their profile

	5. User clicks on apply for verification 6. System will check the database and ask the moderator for the user's verification.
Alternative Flows:	- None
Exceptions:	- User doesn’t meet the criteria to apply for verification
Includes:	- Authenticate user’s identity - User must have account in database
Priority:	- Medium
Frequency of Use:	- Infrequently
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- There must be a record of user’s vote history
Notes and Issues:	- None

Use Case ID:	14		
Use Case Name:	Rate Reviews		
Created By:	Tehzeer	Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able to rate reviews out of 10
Trigger:	Presses' movie rating' button
Preconditions:	- User must be logged into the system
Postconditions:	- User's rating will be visible in 'your rating' tab in the system
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. User will search the movie from the system 4. System will return the movies list matching user's search 5. User will select the movie from the list 6. System will display the 'movie rating' button 7. User will press the button 8. System will display the numbering options from 1 to 10 9. User will select one of the numbers 10. System will display the User's rating number .
Alternative Flows:	11. In step 6, User can now see the movie reviews 12. User can rate the reviews 13. User can now select a new number for the rating 14. System will display the new rating for review
Exceptions:	1. In step 6, user can cancel the process and can return back to menu without giving the rating

Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- User will use it on the daily basis and can add/edit the reviews
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	15		
Use Case Name:	Rate movies		
Created By:	Talal	Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able to rate movies out of 10
Trigger:	Presses' your rating' button
Preconditions:	- User must be logged into the system
Postconditions:	- User's rating will be visible in 'your rating' tab in the system
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. User will search the movie from the system 4. System will return the movies list matching user's search

	5. User will select the movie from the list 6. System will display the 'Your rating' button 7. User will press the button 8. System will display the numbering options from 1 to 10 9. User will select one of the numbers 10. System will display the User's rating number .
Alternative Flows:	11. In step 6, User can now see the rating 12. User can now change the rating 13. User will press again the 'your rating button' 14. System will display the previous rating 15. User can now select a new number for the rating 16. System will display the new rating
Exceptions:	2. In step 6, user can cancel the process and can return back to menu without giving the rating
Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- User will use it on the daily basis and can add/edit the rating
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	16		
Use Case Name:	Share Favourite Movie Quotes		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able to share their best movie quotes
Trigger:	Presses ‘Add Movie Quotes’ button
Preconditions:	- User must be logged into the system
Postconditions:	- User’s quotes will be visible in ‘favourite movie quotes’ tab in the system
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. User presses ‘Add Movie Quotes’ button 4. System will open text box

	5. User will type the quote along with movie name 6. System will display the confirm button. 7. User will press the button 8. System will display the quote in the list
Alternative Flows:	9. In step 4, User can now see the previous quotes 10. User can now remove previous quotes 11. User will press again the ‘remove button’ next to quote 12. System will display the ‘confirm button’ 13. User can now press the button 14. System will remove the quote
Exceptions:	2. In step 6, user can cancel the process and can return back to menu without sharing the quote
Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- infrequent after first use
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	17		
Use Case Name:	Block Users		
Created By:	Talal	Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able to block/unblock the users to avoid them from recommending movies or making connections
Trigger:	Presses' block' button
Preconditions:	- User must be logged into the system
Postconditions:	- User will be successfully blocked
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. System will display the 'friends' tab 4. User will search in friends 5. System will show the list of friends matching the search 6. User will select the user to block 7. User can also add multiple friends to block 8. System will show a block button 9. System will display an alert window to confirm blocking 10. User will block the user 11. The blocked user can not recommend the movies 12. The blocked user can not send the friend request
Alternative Flows:	13. In step 6, User will select the user 14. System will display that user is blocked 15. System will also show an unblock button 16. User will press the button 17. System will unblock the user 18. Now, both can send requests and recommend movies
Exceptions:	If the searched user does not exist, system will take the user back to main
Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- User will use it frequently to block or unblock the users
Business Rules:	- None
Special Requirements:	- None

Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	18		
Use Case Name:	Award Other Users		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able to award other users to appreciate their contributions to the platform
Trigger:	Presses ‘Award’ button
Preconditions:	- User must be logged into the system
Postconditions:	- User will be successfully Awarded
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. User will select the user from their review to Award 4. System will show an award button 5. System will display a selection of silver/gold award 6. User will award the other user 7. The award will show on the other user’s profile
Alternative Flows:	8. User will select the other user from their movie list award 9. System will show an award button 10. System will display a selection of silver/gold award 11. User will award the other user 12. The award will show on the other user’s profile
Exceptions:	If the selected user no longer exists, system will take the user back to main
Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- User will use it frequently to award other users
Business Rules:	- None
Special Requirements:	- None

Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	19		
Use Case Name:	Vote Best Movie Of All Time		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able to vote for the best movie of all time
Trigger:	Presses ‘Your BMOAT’ button
Preconditions:	- User must be logged into the system
Postconditions:	- User’s rating will be visible in ‘your rating’ tab in the system
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. User will search the movie from the system 4. System will return the movies list matching user’s search

	5. User will select the movie from the list 6. System will display the 'Vote BMOAT' button 7. User will press the button 8. System will add to the movie's votes for BMOAT and remove your vote from any previous movie you may have voted for.
Alternative Flows:	9. User can now remove the vote 10. User will press again the 'vote BMOAT button' 11. System will remove your vote for that movie
Exceptions:	2. In step 6, user can cancel the process and can return back to menu without giving the vote
Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- User will use it on the daily basis and can add/edit the rating
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None
Notes and Issues:	- None

Use Case ID:	20		
Use Case Name:	Share Movie of The Day		
Created By:		Last Updated By:	
Date Created:	26th September 2021	Date Last Updated:	

Actors:	User
Description:	Users will be able share their movie of the day

Trigger:	Presses ‘MOTD’ button
Preconditions:	- User must be logged into the system
Postconditions:	- User’s MOTD will be visible in ‘MOTD’ tab in the system
Normal Flow:	1. User enters login credentials in the login screen 2. System verifies the credentials and logs in 3. User will search the movie from the system 4. System will return the movies list matching user’s search

	5. User will select the movie from the list 6. System will display the 'MOTD' button 7. User will press the button 8. System will display the movie as MOTD
Alternative Flows:	9. In step 8, User can now see the MOTD 10. User can now remove the MOTD 11. User will press again the 'MOTD button' 12. System will remove your selection for MOTD
Exceptions:	2. In step 6, user can cancel the process and can return back to menu without giving the rating
Includes:	- Authenticates user identity - Verifies entered information with database
Priority:	- High
Frequency of Use:	- User will use it on the daily basis and can add/edit the rating
Business Rules:	- None
Special Requirements:	- None
Assumptions:	- None
Notes and Issues:	- None

4. External Interface Requirements

4.1 User Interfaces

The user interface is a simple to use and intuitive web page, where data is presented in a nondescript way. All the options for the use cases shall be made into UI elements and high priority items shall be presented at the home page after login.
There will be helpful options for movies displayed at the start, giving ideas for popular movies that could be added to lists and user created lists that are also popular. A call to action button shall be made at the center for a “Create a list” action at the top of the page.
Options for help, privacy and contact us shall be provided at the footer of each page.

4.2 Hardware Interfaces

There are no hardware interface requirements apart from the keyboard and mouse or mobile phone required to interact with the web page.

4.3 Software Interfaces

There are no 3rd party software with which our product must communicate with, as this is a standalone web page with a singular database to run queries through.

4.4 Communications Interfaces

There will be an email communication method set up in order to confirm the user account created.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. Any request for the page shouldn't take more than 10 seconds.
2. Upon loading of the page, the end user should be prompted to give access to their location within 5 seconds. The end user needs to have functional GPS in their device for the system to be able to filter the veterinary profiles around their location.
3. The system should be able to cater to a high number of simultaneous users (Up to 10000 users). Any more users would depend on the performance of the server the software is deployed on.
4. The data entered by admin should be visible within 15 seconds
5. The data returned by any search filter used by the end user should be available within 3 seconds and should be visible in the end user view within 15 seconds.
6. After placing the reset password request, the user should receive a reset code on their primary contact number/email within 2 minutes.
7. As per standard memory consumption, our application will consume (130 - 450) MB of memory.

5.2 Safety Requirements

1. Users will have the ability to report harmful content to moderators, who will then have the ability to restrict the content posted by the specified user.
2. Users will have the ability to block other users from seeing their publicly posted content and send connection requests.
3. Users will only be able to view movies according to their age and respective MPA ratings
4. Lists created by users shall only be visible to those they are connected with (and administrator)

5.3 Security Requirements

1. Certain checks won't allow anyone to perform admin's tasks, not even authenticated users which means admin's rights are protected.
2. Admin and user's account are protected through an encrypted password.
3. User creation and password reset requests will be done through protected emails.

5.4 Software Quality Attributes

1. Software's test cases will make it reliable.
2. Software is easy to use as it is built on a friendly framework.
3. Users can easily adapt the usage environment of the software.
4. Every function is easy and defined in the description for easy use.

Appendix A: Glossary

- **Movie:** Long formed video content listed on the website for users to react and comment on among other things.
- **Comment:** a verbal or written remark expressing an opinion or reaction of a user on the website.
- **Review:** a critical appraisal of the movie by a user on the website.
- **MPA Rating:** The **Motion Picture Association film rating system** is used to rate a movie's suitability for certain audiences based on its content.