Use Cases

for

MANGA/COMIC WEB APPLICATION

Version 1.0 approved

Prepared by M. Oskhar Mub

UIN Syarif Hidayatullah Jakarta

4 January 2024

Source Code: github.com/kreatfqode/api-web-comic

Table of Contents

1. Introduction

The Use Case Document is a business document which provides a story of how a system, and its actors, will be utilized to achieve a specific goal. An effective Use Case should provide a detailed step-by-step description of how the system will be used by its actors to achieve the planned outcome. The purpose of the Use Case is to tie the business needs of the system to the design parameters of the system to ensure that the completed system achieves the goals established by the business requirements. The level of detail in Use Cases may vary greatly depending on the size and complexity of the system being designed.

This Use Case has been developed for ABC Corporation's new system for ordering material based on the design team's gathering of business and functional area requirements. The Material Ordering System will replace the manual material ordering processes currently utilized by ABC Corp. ABC Corp. has identified business needs for reducing man hours for material ordering and leveraging existing software platforms (i.e. SAP) to help manage material ordering and inventory management. The new Material Ordering System will be designed to meet these business needs and improve ABC Corp.'s overall business strategy.

2. Explanation of Use Case Contents

Use Case formats and contents may vary based on system requirements, organizational standards, or unique situations. However, a majority of Use Cases consist of some fundamental contents which may be applied across a wide range of system types. This section will provide explanations for each section of the Use Case.

Name of Use Case: Provide a short name for the use case which should lend itself to the objective of the system.

Description: This section should provide a description of both the reason for using the use case and the expected outcome of the use case.

Actors: Actors may be primary or secondary. Primary actors are the people who will be initiating the system described in the use case. Secondary actors are those will participate in the completion of the use case.

Precondition: This section should describe any conditions that must be true or activities that must be completed prior to executing the use case.

Postcondition: This section should describe the state of the system at the conclusion of the use case. Postconditions may include conditions for both successful and unsuccessful execution of the use case.

Flow: This section should describe all actions of the user and the expected system responses for planned normal execution of the use case. The description should be sequential and provide adequate detail to understand all user actions and system responses.

Alternative Flows: Many use cases have varying or special extensions or conditions which are separate from the main flow but also necessary. Alternative flows are usually the result of options or exceptions built into the use case which may alter the primary flow.

Exceptions: When use cases are executed, there may be various conditions which result in errors. This section should describe any errors that may result during use case execution and how the system will react or respond to those errors.

Requirements: This section should describe any non-functional or special requirements for the system as the use case is executed. These requirements may consist of legal or regulatory requirements, quality standards, or organizational requirements that are outside of the functional requirements the system is expected to perform.

3. Product Use Case

3.1. User Registration

Name of Use Case:	User Registration		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	20/1/2024
Description:	This use case describes the process of a new user registering on the web comic/manga application.		
Actors:	Guest (New User), System		
Preconditions:	The user does not have an existing account.		
Postconditions:	The user's account is successfully created, and they are ready to log in.		
Flow:	1. User provides a unique username, a valid email, and a secure password. 2. The application validates the provided information. 3. The user is registered with a default role set as 'User'. 4. The system sends a confirmation email to the user. 5. User confirms the registration by clicking the link in the confirmation email.		
Alternative Flows:	1. If the email confirmation link is not clicked within a certain time, the registration may expire, and the user needs to register again.		
Exceptions:	1. If the email confirmation link is not clicked within a specified time, the registration becomes invalid, and the user must re-register.		
Requirements:	1. The system must validate the uniqueness of usernames and ensure that email addresses are valid.		

	2. Confirmation emails must be sent securely and promptly.

3.2. Authentication and Authorization

<table>
<tr><td><b>Name of Use Case:</b></td><td colspan="3">Authentication and Authorization</td></tr>
<tr><td><b>Created By:</b></td><td>M. Oskhar Mub</td><td><b>Last Updated By:</b></td><td>M. Oskhar Mub</td></tr>
<tr><td><b>Date Created:</b></td><td>20/1/2024</td><td><b>Last Revision Date:</b></td><td>20/1/2024</td></tr>
<tr><td><b>Description:</b></td><td colspan="3">This use case involves the processes of authentication and authorization to ensure secure and controlled access to the manga platform.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">User, Admin, Guest</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">The user or admin interacts with the platform.</td></tr>
<tr><td><b>Postconditions:</b></td><td colspan="3">1. The user gains appropriate access to their account features.
2. The admin gains access to the administrative features.</td></tr>
<tr><td><b>Flow:</b></td><td colspan="3">User Authentication:
1. Users access the platform.
2. User initiates the login process.
a. Enter Credentials: The user provides a valid combination of username/email and password.
3. System verifies user credentials.
a. Check against Database: Verify the entered credentials against stored user data.
b. Two-Factor Authentication (optional): If enabled, the system prompts the user for a second form of authentication.
4. Users gain access to their account.
User Authorization:
5. System checks user roles and permissions.
a. User Role Assignment: Assign appropriate roles (e.g., regular user, premium user) to the authenticated user.
b. Check Permissions: Ensure the user has the necessary permissions for the requested action.
6. User performs actions within authorized scope.
Admin Authentication:</td></tr>
</table>

	7. Admin accesses the platform. 8. Admin initiates the login process. a. Enter Credentials: The admin provides valid credentials. 9. System verifies admin credentials. a. Check against Database: Verify the entered credentials against stored admin data. b. Two-Factor Authentication (optional): If enabled, the system prompts the admin for a second form of authentication. 10. Admin gains access to administrative features. Admin Authorization: 11. System checks admin roles and permissions. a. Admin Role Assignment: Assign appropriate roles (e.g., content moderator, super admin) to the authenticated admin. b. Check Permissions: Ensure the admin has the necessary permissions for the requested administrative action. 12. Admin performs actions within authorized scope.
Alternative Flows:	1. User Forgot Password: a. If a user forgets their password, the system provides a password reset option through email or other verification methods.
Exceptions:	1. Invalid Credentials: a. If credentials are invalid, the system informs the user or admin and prompts for correct credentials. 2. Account Lockout (optional): a. After multiple unsuccessful login attempts, the system may temporarily lock the account for security reasons.
Requirements:	1. Secure Storage of Credentials: Ensure passwords are securely stored using encryption. 2. Two-Factor Authentication (optional): Provide an option for an additional layer of authentication. 3. Role-Based Access Control (RBAC): Implement RBAC for both users and admins. 4. Password Reset Mechanism: Allow users to reset their password through secure methods.

3.3. User Recommender System

Name of Use Case:	User Recommender System

Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	20/1/2024
Description:	This use case involves the implementation of a User Recommender System to provide personalized manga recommendations based on user preferences and behavior.		
Actors:	User		
Preconditions:	1. The user is registered and logged into their account. 2. The user has interacted with manga content on the platform.		
Postconditions:	The system generates and displays personalized manga recommendations for the user.		
Flow:	1. User navigates to the homepage or recommendation section. 2. User initiates the Recommender System. 3. System analyzes user behavior and preferences. a. User Interaction Tracking: The system considers the user's history, including viewed titles, liked/disliked content, and previous searches. b. User Genre Preferences: Analyzing the genres the user has engaged with the most. 4. System generates personalized manga recommendations. a. Collaborative Filtering: Utilizing collaborative filtering algorithms to suggest manga based on similar users' preferences. b. Content-Based Filtering: Considering characteristics of manga the user has liked to recommend similar content. 5. Recommendations are displayed to the user. a. Option to Provide Feedback: User can provide feedback on individual recommendations (like/dislike) for further refinement.		
Alternative Flows:	1. aIf the user has not interacted with enough content: a. The system informs the user that more interactions are needed to improve recommendations.		
Exceptions:	1. Insufficient User Interaction Data: a. If the user has not interacted with enough content, the system informs the user and encourages more interactions for better recommendations.		

Requirements:	5. User Interaction Tracking: The system must track and analyze user interactions with manga content. 6. Collaborative Filtering Algorithm: Implementation of collaborative filtering to suggest manga based on similar users. 7. Content-Based Filtering Algorithm: Consideration of manga characteristics for personalized recommendations. 8. Feedback Mechanism: Users should have the option to provide feedback on recommendations.

3.4. Manga Discovery with Advanced Filters

Name of Use Case:	Manga Discovery with Advanced Filters		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	20/1/2024
Description:	This use case involves an advanced search functionality known as "Catch all manga," allowing users to discover manga titles based on various filters such as search keywords, length, start status, genre, and release year.		
Actors:	User		
Preconditions:	User has accessed the manga discovery section.		
Postconditions:	Users are presented with a refined list of manga titles based on selected filters.		
Flow:	1. User navigates to the manga discovery section. 2. User initiates the "Catch all manga" advanced search. a. View Available Filters: User views a list of available filters, including search keywords, length, start status, genre, and release year. 3. User sets desired filters for manga search. a. Search Keywords: User enters keywords for the manga title. b. Length Filter: User sets the desired length range (e.g., short, medium, long). c. Start Status Filter: User selects whether to include ongoing, completed, or both types of manga. d. Genre Filter: User selects one or multiple genres from the available options. e. Release Year Filter: User specifies the release year or range.		

	4. System processes user-selected filters and refines the manga list. 5. User is presented with a list of manga titles based on the applied filters.
Alternative Flows:	1. Reset Filters: a. User has the option to reset all filters to their default values
Exceptions:	1. No Matching Manga: a. If no manga titles match the specified filters, the system informs the user and suggests adjusting the filters.
Requirements:	1. Advanced Search Interface: Implement a user-friendly interface allowing easy selection and modification of filters. 2. Filter Persistence: Ensure that user-selected filters persist during the browsing session. 3. Real-time Filtering: Provide real-time updates to the manga list as the user adjusts filters. 4. Responsive Design: Ensure the system is responsive and works well across different devices.

3.5. Language Selection and Content Localization

Name of Use Case:	Language Selection and Content Localization		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	20/1/2024
Description:	This use case involves enabling users to select their preferred language and ensuring content is displayed in the chosen language, supporting Internationalization (i18n) and Localization (l10n).		
Actors:	User		
Preconditions:	Users access the platform.		
Postconditions:	User interfaces and content are displayed in the selected language.		
Flow:	1. Users access the platform. 2. User navigates to the settings or profile section. 3. User initiates the language selection process.		

	a. View Available Languages: User views a list of available languages. b. Select Preferred Language: User selects their preferred language from the list. 4. System saves the selected language preference in the user's profile. 5. User interfaces and content are displayed in the selected language. a. Dynamic Content Localization: All dynamic content, including menus, buttons, and labels, is displayed in the chosen language. b. Static Content Localization: Static content, such as help sections and informational text, is presented in the selected language.
Alternative Flows:	1. Fallback to Default Language: a. If the selected language translation is unavailable for certain content, the system falls back to the default language.
Exceptions:	1. Unavailable Language: a. If the selected language is not available, the system informs the user and suggests an alternative.
Requirements:	5. Multilingual Support: Provide support for a range of languages to accommodate diverse user preferences. 6. Language Preference Persistence: Ensure that the user's language preference is saved for future visits. 7. Content Localization Resources: Establish a system for translating and managing content in different languages. 8. Fallback Mechanism: Implement a mechanism to gracefully handle situations where translations are not available.

3.6. User Leaving Comment

Name of Use Case:	User Leaving Comment		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	20/1/2024
Description:	Describes the process of a user leaving a comment on a chapter of the manga.		
Actors:	User		

Preconditions:	User is logged in and viewing a chapter of the manga.
Postconditions:	The user's comment is successfully added to the system.
Flow:	1. User navigates to a manga or chapter. 2. User decides to leave a comment and enters the content. 3. User submits the comment. 4. The application associates the comment with the user, manga, and chapter.
Alternative Flows:	1. Users can reply to existing comments, creating a threaded discussion.
Exceptions:	1. If the comment content is empty, the system prompts the user to enter valid content.
Requirements:	1. The system should allow users to engage in threaded discussions by replying to existing comments.

3.7. User Giving Rating

Name of Use Case:	User Giving Rating		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	21/1/2024
Description:	Describes the process of a user providing a rating and optional review comment for a manga.		
Actors:	User		
Preconditions:	User is logged in and viewing a manga.		
Postconditions:	The user's rating is successfully added to the system.		
Flow:	1. User navigates to the manga they want to rate. 2. User provides a numerical rating (1 to 5) and an optional review comment. 3. User submits the rating. 4. The application associates the rating with the user and manga.		

Alternative Flows:	1. If the user chooses not to leave a review comment, they can skip that part of the process.
Exceptions:	1. If the rating input is invalid, the system prompts the user to provide a valid rating.
Requirements:	1. The system must validate the numerical rating provided by the user.

3.8. Reporting Inappropriate Content

Name of Use Case:	Reporting Inappropriate Content		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	21/1/2024
Description:	Describes the process of a user reporting inappropriate content or spam.		
Actors:	User, System		
Preconditions:	User identifies inappropriate content while using the application.		
Postconditions:	A web view report is submitted to the system.		
Flow:	1. User identifies inappropriate content. 2. User initiates the report, providing details such as the type of content and a description. 3. User submits the report. 4. The application associates the report with the user, manga, and chapter.		
Alternative Flows:	1. Users can select the type of inappropriate content from predefined categories.		
Exceptions:	1. If the report details are incomplete, the system prompts the user to provide all required information.		
Requirements:	1. Inappropriate content reports should include predefined categories for ease of classification.		

3.9. User Following an Author

Name of Use Case:	User Following an Author		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	21/1/2024
Description:	Describes the process of a user following an author to receive updates.		
Actors:	User		
Preconditions:	User is logged in and viewing an author's profile.		
Postconditions:	The user successfully follows the author, and the follow action is recorded.		
Flow:	1. User navigates to an author's profile. 2. User clicks the 'Follow' button. 3. The application records the follow action, including a timestamp.		
Alternative Flows:	1. Users can unfollow an author by clicking an 'Unfollow' button.		
Exceptions:	1. If the user is not logged in, the system prompts them to log in before following an author.		
Requirements:	1. The follow mechanism must be robust, allowing users to unfollow authors if desired.		

3.10. User Bookmarking Manga

Name of Use Case:	User Following an Author		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	21/1/2024
Description:	Describes the process of a user bookmarking a manga to save it for future reading.		
Actors:	User		
Preconditions:	User is logged in and browsing manga titles.		

Postconditions:	The selected manga is successfully bookmarked in the user's profile for future reference.
Flow:	1. User navigates to the list of available manga titles. 2. Users click on a specific manga title they want to bookmark. 3. User selects the option to "Bookmark" the manga. 4. The application adds the manga to the user's list of bookmarked titles.
Alternative Flows:	1. If the user accidentally clicks on the "Bookmark" option, they can undo the action by selecting "Remove Bookmark."
Exceptions:	1. If there is an error in bookmarking the manga, the system prompts the user to try again.
Requirements:	1. The system must provide a user-friendly interface for managing bookmarked manga titles. 2. Users should be able to quickly access their bookmarked titles from their profile.

3.11. User Notification Inbox

Name of Use Case:	User Notification Inbox		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	20/1/2024	**Last Revision Date:**	21/1/2024
Description:	This use case involves providing users with a dedicated notification inbox within the web comic/manga application.		
Actors:	User,System		
Preconditions:	User is logged into their account.		
Postconditions:	Users can conveniently review and manage notifications within the dedicated inbox.		
Flow:	1. Users access the notification inbox from their profile. 2. The system displays a chronological list of notifications. 3. Users can mark notifications as read, unread, or delete them.		

Alternative Flows:	1. If the user misses a notification, the system retains it in the inbox for a certain period. 2. If the user clicks on a notification, the system directs them to the relevant content (e.g., manga, chapter).
Exceptions:	1. If there is an issue retrieving notifications, the system displays an error message and prompts the user to try again. 2. If a user marks a notification as unread, but the action fails, the system logs the error for further investigation. 3. If the user attempts to delete a notification that is critical for system updates, the system prompts for confirmation to avoid accidental deletion.
Requirements:	1. The system must provide real-time updates for time-sensitive notifications. 2. Notification inbox must support filtering and search capabilities. 3. Notifications should be retained for a reasonable period, even if the user is not actively checking the inbox. 4. The system should offer a seamless transition from the notification inbox to the associated content.

3.12. Author Registration and Onboarding

Name of Use Case:	Author Registration and Onboarding		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the registration and onboarding process for new authors.		
Actors:	Admin, Author		
Preconditions:	Admin is logged into the admin dashboard.		
Postconditions:	New authors are successfully registered and added to the platform.		
Flow:	1. Admin accesses the Author Management section. 2. Admin initiates the author registration process. 3. Author completes the registration form, providing necessary details. 4. The application validates the information and creates a new author profile.		
Alternative Flows:	1. If the registration form is incomplete, the system prompts the admin to ensure all required fields are filled.		

Exceptions:	1. If the registration process encounters technical errors, the system logs the issue for further investigation.
Requirements:	1. The system must enforce unique usernames and email addresses. 2. Secure transmission and storage of registration data.

3.13. Author Profile Management

Name of Use Case:	Author Profile Management		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case covers the management of author profiles, including updating information.		
Actors:	Admin, Author		
Preconditions:	Author is logged into their account.		
Postconditions:	Author profile details are successfully updated.		
Flow:	1. Author accesses their profile from the dashboard. 2. Author edits profile information such as pen name, biography, and social media links. 3. The application validates and updates the author's profile.		
Alternative Flows:	1. If the profile update includes invalid data, the system prompts the author to correct the errors.		
Exceptions:	1. If there are issues with updating the profile picture, the system allows the author to retry or select a new image.		
Requirements:	1. The system must support the upload and storage of profile pictures. 2. Real-time validation of social media links.		

3.14. Manga Creation and Management

Name of Use Case:	Manga Creation and Management

<table>
<tr><td><b>Created By:</b></td><td>M. Oskhar Mub</td><td><b>Last Updated By:</b></td><td>M. Oskhar Mub</td></tr>
<tr><td><b>Date Created:</b></td><td>28/1/2024</td><td><b>Last Revision Date:</b></td><td>28/1/2024</td></tr>
<tr><td><b>Description:</b></td><td colspan="3">This use case involves the creation and management of manga titles by authors, with an added focus on genre classification and tagging.</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">Author</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">The author is logged into their account.</td></tr>
<tr><td><b>Postconditions:</b></td><td colspan="3">New manga titles are successfully created and associated with the author's profile, including genre classification and tagging.</td></tr>
<tr><td><b>Flow:</b></td><td colspan="3">1. Author navigates to the dashboard's manga management section.
2. Author initiates the process of adding a new manga.
3. Author provides details such as title, description, genre, and cover image.
a. Support for multiple genres: Author can select multiple genres for each manga.
4. The application associates the new manga with the author's profile.
a. Validation of unique titles: The system ensures that each title is unique for the author.</td></tr>
<tr><td><b>Alternative Flows:</b></td><td colspan="3">1. If the manga details are incomplete:
i. The system prompts the author to fill in the required fields.
2. If there is an issue with uploading the cover image:
i. The system allows the author to retry or select a different image.</td></tr>
<tr><td><b>Exceptions:</b></td><td colspan="3">1. Incomplete Manga Details:
i. If the manga details are incomplete, the system prompts the author to fill in the required fields.
2. Issue with Cover Image:
i. If there is an issue with uploading the cover image, the system allows the author to retry or select a different image.</td></tr>
<tr><td><b>Requirements:</b></td><td colspan="3">1. Support for multiple genres for each manga.
2. Validation of unique titles for each author.</td></tr>
</table>

3.15. Chapter Management

Name of Use Case:	Chapter Management		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the creation and management of chapters within a manga.		
Actors:	Author, System		
Preconditions:	The author is logged into their account.		
Postconditions:	New chapters are successfully added to the selected manga.		
Flow:	1. Author selects a manga from their dashboard. 2. Author initiates the process of adding a new chapter. 3. Author provides details such as title, chapter number, and content. 4. The application associates the new chapter with the selected manga.		
Alternative Flows:	1. If the chapter details are incomplete, the system prompts the author to fill in the required fields.		
Exceptions:	1. If there is an issue with uploading chapter content, the system allows the author to retry or choose a different content file.		
Requirements:	1. Support for various content formats (images, text) for chapters. 2. Automatic chapter numbering.		

3.16. Manga Trend Statistics for Authors

Name of Use Case:	Manga Trend Statistics for Authors		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case enables authors to view trend statistics derived from user interactions across all manga titles, providing a comprehensive overview of global user engagement.		
Actors:	Author, System		

Preconditions:	Author has logged into their account on the manga platform.
Postconditions:	Author has access to trend statistics reflecting user interactions across all manga titles.
Flow:	1. Author navigates to the dashboard or analytics section. 2. Author initiates the "Manga Trend Statistics" feature. a. Select Global View: Author selects the option for a global view of trend statistics. 3. System generates and displays trend statistics. a. Aggregate User Interactions: The system aggregates data on user interactions, including views, likes, comments, and bookmarks, across all manga titles. b. Time-Based Analysis: Author can choose different time intervals (e.g., daily, weekly, monthly) for trend analysis. c. Popular Genres: System highlights popular genres based on user engagement. d. User Demographics: Optionally, the system may provide insights into user demographics engaging with the manga content. 4. Author reviews and interprets trend statistics. a. Identify Top-performing Manga: Author identifies manga titles with the highest user engagement. b. Understand User Preferences: Author gains insights into user preferences, allowing for better decision-making in future content creation. c. Adapt Strategies: Authors may adapt their content creation or marketing strategies based on observed trends.
Alternative Flows:	1. Filter by Specific Manga: a. Author has the option to view trend statistics for a specific manga title.
Exceptions:	1. Insufficient Data: a. If there is not enough data available for meaningful trend analysis, the system informs the author and suggests reviewing a longer time period.

Requirements:	1. Comprehensive User Interaction Tracking: Ensure that the system accurately tracks user interactions across all manga titles. 2. Data Visualization: Implement clear and visually appealing charts or graphs to represent trend statistics. 3. Custom Time Range Selection: Allow authors to choose custom time ranges for trend analysis. 4. Privacy Considerations: Ensure that user demographics are presented in an aggregated and anonymized manner to protect user privacy.

3.17. User Interaction Tracking

Name of Use Case:	User Interaction Tracking		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves tracking user interactions with an author's content.		
Actors:	Author, System		
Preconditions:	The author is logged into their account.		
Postconditions:	Author gains insights into user engagement.		
Flow:	1. Author accesses the dashboard's analytics section. 2. Author reviews metrics such as views, comments, and ratings for each manga and chapter. 3. The application displays user interaction data in a comprehensible format.		
Alternative Flows:	1. If there is a sudden spike in user engagement, the system provides real-time notifications to the author.		
Exceptions:	1. If the analytics data retrieval fails, the system logs the issue for investigation.		
Requirements:	1. Real-time updates for user engagement metrics. 2. Historical data tracking for analytics.		

3.18. Notification Management

Name of Use Case:	Notification Management

Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case covers the management of notifications related to user interactions.		
Actors:	Author, System		
Preconditions:	Author can efficiently manage and respond to notifications.		
Postconditions:	Author gains insights into user engagement.		
Flow:	1. Author accesses the dashboard's notification center. 2. Author reviews new comments, ratings, or follow notifications. 3. The application allows the author to respond or take relevant actions directly from the dashboard.		
Alternative Flows:	1. If the author receives a large number of notifications, the system provides sorting and filtering options.		
Exceptions:	1. If the notification retrieval fails, the system retries or provides an error message.		
Requirements:	1. Support for various notification types (comments, ratings, follows). 2. Bulk actions for efficient notification management.		

4. System Use Case

4.1. Admin Managing Users

Name of Use Case:	Admin Managing Users		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the admin managing user accounts and roles within the system.		
Actors:	Admin		
Preconditions:	Admin is logged into the admin dashboard.		

Postconditions:	User accounts and roles are successfully managed.
Flow:	1. Admin navigates to the user management section. 2. Admin views a list of all users and their details. 3. Admin has the ability to search for specific users based on username or email. 4. Admin can view and edit user details, including username, email, and role. 5. Admin can deactivate or delete user accounts as needed.
Alternative Flows:	1. If a user account is deactivated, the system retains the user data but restricts login access.
Exceptions:	1. If there are errors in editing user details, the system prompts the admin to correct them.
Requirements:	1. The admin must have the authority to view, edit, deactivate, or delete user accounts. 2. Deactivated user accounts should not be able to log in.

4.2. Admin Managing Authors

Name of Use Case:	Admin Managing Users		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the admin managing author profiles and associated manga titles.		
Actors:	Admin		
Preconditions:	Admin is logged into the admin dashboard.		
Postconditions:	Author profiles and manga titles are successfully managed.		
Flow:	1. Admin navigates to the author management section. 2. Admin views a list of all authors and their profiles. 3. Admin has the ability to search for specific authors based on pen name or email. 4. Admin can view and edit author details, including pen name and biography. 5. Admin can deactivate or delete author profiles.		

	6. Admin can view a list of manga titles associated with each author. 7. Admin has the ability to edit or delete individual manga titles.
Alternative Flows:	1. If an author profile is deactivated, all associated manga titles are also deactivated.
Exceptions:	1. If there are errors in editing author details or manga titles, the system prompts the admin to correct them.
Requirements:	1. The admin must have the authority to view, edit, deactivate, or delete author profiles and associated manga titles.

4.3. Admin Managing Manga

Name of Use Case:	Admin Managing Manga		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the admin managing manga details, chapters, and user interactions.		
Actors:	Admin		
Preconditions:	Admin is logged into the admin dashboard.		
Postconditions:	Manga details, chapters, and user interactions are successfully managed.		
Flow:	1. Admin navigates to the manga management section. 2. Admin views a list of all manga titles and their details. 3. Admin has the ability to search for specific manga titles based on title or genre. 4. Admin can view and edit manga details, including title, description, and cover image. 5. Admin can deactivate or delete manga titles. 6. Admin views a list of chapters for each manga and has the ability to edit or delete chapters. 7. Admin can view user comments and ratings for each manga.		

	8. Admin has the authority to delete inappropriate comments or ban users if necessary.
Alternative Flows:	1. If a manga title is deactivated, all associated chapters, comments, and ratings are also deactivated.
Exceptions:	1. If there are errors in editing manga details or chapters, the system prompts the admin to correct them.
Requirements:	1. The admin must have the authority to view, edit, deactivate, or delete manga titles, chapters, and user interactions.

4.4. Admin Managing Web View Reports

Name of Use Case:	Admin Managing Web View Reports		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the admin managing web view reports submitted by users.		
Actors:	Admin, User		
Preconditions:	Admin is logged into the admin dashboard.		
Postconditions:	Web view reports are successfully managed.		
Flow:	1. Admin navigates to the web view report management section. 2. Admin views a list of all web view reports with details such as report type and description. 3. Admin can search for specific reports based on manga title or user. 4. Admin can view the details of each report, including the reported content and status. 5. Admin has the ability to mark reports as processed or rejected based on the investigation.		
Alternative Flows:	1. If a report is marked as processed, the admin may need to take further actions such as content removal or user warning.		

Exceptions:	1. If there are errors in processing web view reports, the system prompts the admin to correct them.
Requirements:	1. The admin must have the authority to view, process, and manage web view reports.

4.5. Admin Managing Follows

Name of Use Case:	Admin Managing Follows		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	This use case involves the admin managing user follows on authors.		
Actors:	Admin, System		
Preconditions:	Admin is logged into the admin dashboard.		
Postconditions:	Follow relationships are successfully managed.		
Flow:	1. Admin navigates to the follow management section. 2. Admin views a list of all user follows with details such as follower and following authors. 3. Admin can search for specific follow relationships based on user or author. 4. Admin has the ability to view details of each follow relationship, including timestamps. 5. Admin can delete or modify existing follow relationships if necessary.		
Alternative Flows:	1. If a following relationship is modified, the timestamp is updated to reflect the last interaction.		
Exceptions:	1. If there are errors in managing follow relationships, the system prompts the admin to correct them.		
Requirements:	1. The admin must have the authority to view, modify, or delete user follow relationships.		

4.6. Admin Giving Warning to Author

Name of Use Case:	Admin Giving Warning to Author		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	Describes the process of an admin issuing a warning to an author for content violations or policy breaches.		
Actors:	Admin, Author		
Preconditions:	Admin is logged into the admin dashboard, and the author has violated content policies.		
Postconditions:	The author receives a warning notification, and the admin's action is recorded.		
Flow:	1. Admin identifies a specific author who has violated content policies or breached platform rules. 2. Admin selects the option to issue a warning to the author. 3. Admin provides a clear and detailed explanation of the violation or breach. 4. The application sends a notification to the author about the warning.		
Alternative Flows:	1. If the violation is severe, the admin may choose to suspend the author's account temporarily.		
Exceptions:	1. If there are errors in issuing the warning, the system prompts the admin to correct them.		
Requirements:	1. The warning system should be transparent, providing authors with a clear understanding of the violation and potential consequences.		

4.7. Admin Sending Message to Author

Name of Use Case:	Admin Sending Message to Author		
Created By:	M. Oskhar Mub	**Last Updated By:**	M. Oskhar Mub
Date Created:	28/1/2024	**Last Revision Date:**	28/1/2024
Description:	Describes the process of an admin sending a direct message to an author for communication purposes.		
Actors:	Admin, Author		

Preconditions:	Admin is logged into the admin dashboard, and there is a need for direct communication with the author.
Postconditions:	The author receives the admin's message, and both parties can engage in a private conversation.
Flow:	1. Admin navigates to the author management section and selects the specific author. 2. Admin chooses the option to send a direct message to the author. 3. Admin composes the message, providing necessary details or instructions. 4. The application sends the message to the author's inbox, notifying them of the new communication.
Alternative Flows:	1. Authors can respond to admin messages, creating a threaded communication history.
Exceptions:	1. If there are errors in sending the message, the system prompts the admin to correct them.
Requirements:	1. The messaging system should ensure privacy, and messages should be stored securely for reference purposes.

5. Sponsor Acceptance

Approved by the Project Sponsor:

________________________ Date:________________________

<Project Sponsor>
<Project Sponsor Title>