DitShev

**GoaTalks
Use-case Specification**

Version 2.0

Revision History

Date	Version	Description	Author
14/11/2023	1.0	First Version	DitShev
02/12/2023	2.0	Second Version	DitShev

Table Of Content

Vision (Small Project)

1.   UML Use-case Diagram
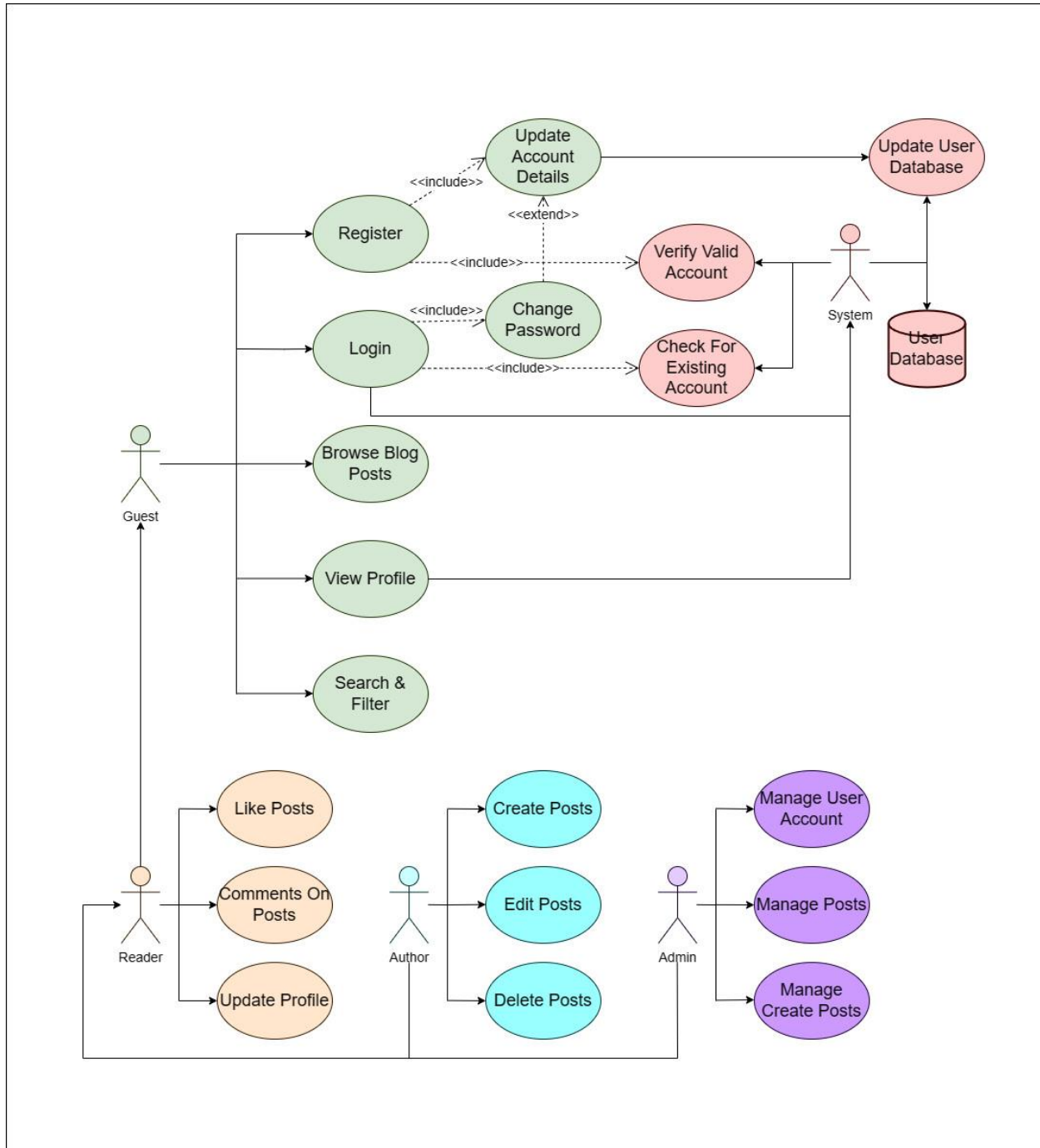
2.   Use-case Specification

2.1   Authorization

2.1.1 Register

	User Registration
Actors	- User (Author, Reader) - Admin
Description	A new user wants to register an account on the blogging website.
Flow	1. Guest navigates to the registration page. 2. Guest provides required information (username, email, password). 3. System validates the information. 4. If validation is successful, the system creates a new user account. 5. The system sends a confirmation email to the user. 6. User confirms the email address.
Alternate Flow	If the provided information is invalid, the system prompts the user to correct the errors
Exception Flow	If the email confirmation is not received within a specified time, the system prompts the user to request another confirmation email.

2.1.2 Log in

	User Login
Actors	- Registered User - Admin
Description	A registered user wants to log in to their account.
Flow	1. User navigates to the login page. 2. User enters their username and password. 3. The system validates the credentials. 4. If validation is successful, the system logs in the user.
Alternate Flow	If the entered credentials are incorrect, the system prompts the user to enter valid information.

2.1.3 Change password

	Change Password
Actors	- Logged-in User - Admin
Description	A logged-in user wants to change their password for security reasons or as part of routine maintenance.
Pre-conditions	The user is logged into their account.
Flow	1. User navigates to the "Account Settings" or "Change Password" section. 2. System prompts the user to enter their current password for verification. 3. User enters their current password. 4. System validates the current password. • If validation fails, the system notifies the user and the use-case ends. • If validation is successful, the user proceeds to the next step. 5. User enters a new password and confirms it. 6. System validates the new password against specified criteria (e.g., length, complexity). • If validation fails, the system notifies the user and the use case ends. • If validation is successful, the user's password is updated. 7. The system confirms the successful password change to the user.
Alternate Flo	If the user fails to provide the correct current password, the system notifies them and prompts them to try again.
Exception Flow	If the system encounters technical issues during the password change process, it notifies the user about the problem and advises them to try again later.
Post-conditions	The user's password is successfully changed, and they can log in using the new credentials.

2.2   Browse Blog Posts

	Browse Blog Posts
Actors	- Guest and Logged-in User - Admin
Description	A user wants to browse and read blog posts.
Flow	1. User navigates to the "Browse" or "Home" page. 2. The system displays a list of recent blog posts. 3. User clicks on a post to view its full content.
Alternate Flow	If the system cannot retrieve posts, it displays an appropriate message.

2.3   Create Posts

	Create Blog Posts
Actors	- Logged-in User
Description	A logged-in user wants to create a new blog post.
Flow	1. User navigates to the "Create Post" page. 2. User enters the title, content, and any relevant details. 3. System validates the content. 4. If validation is successful, the system creates the blog post and associates it with the user's account.
Alternate Flow	If the entered content is invalid, the system prompts the user to correct errors.

2.4   Comment On Posts

	Comment On Posts
Actors	- Logged-in User
Description	A logged-in user wants to leave a comment on a blog post.
Flow	1. User navigates to a blog post. 2. User enters a comment in the provided field. 3. The system validates the comment. 4. If validation is successful, the system adds the comment to the post.
Alternate Flow	If the entered content is invalid, the system prompts the user to correct errors.

2.5   Like A Post

	Like A Post
Actors	- Logged-in User
Description	A logged-in user wants to express appreciation for a blog post by liking it.
Pre-conditions	1. The user is logged into their account. 2. The user is viewing a specific blog post.
Flow	1. User navigates to the blog post they want to like. 2. User clicks on the "Like" button associated with the post. 3. The system records the user's like for the specific post.
Alternate Flow	If the user has already liked the post, clicking the "Like" button again may toggle the like status (like/unlike).
Exception Flow	If the system encounters technical issues during the liking process, it notifies the user about the problem and advises them to try again later.
Post-conditions	The user's like is successfully recorded for the blog post.

2.6   Edit / Delete Post

	Edit / Delete Post
Actors	- Author
Description	The author of a blog post wants to edit or delete their own post.
Flow	1. Author navigates to their blog post. 2. Author selects the option to edit or delete. 3. If editing, the system allows the author to modify the post content. 4. If deleting, the system prompts the author for confirmation. 5. After confirmation, the system either updates the post or deletes it.
Alternate Flow	If the system encounters issues with editing or deleting, it provides an appropriate error message.

2.7 Search and Filter Posts

	Search and Filter Posts
Actors	- Guest and Logged-in User
Description	A user wants to find specific blog posts or narrow down the list of displayed posts based on certain criteria.
Pre-conditions	The user is on the "Browse" or "Search" page of the blogging website.
Flow	1. User navigates to the search/filter interface on the website. 2. User enters search keywords, filters, or selects sorting options. 3. The system processes the user's input and retrieves relevant blog posts. 4. The system displays the filtered or searched blog posts according to the user's criteria.
Alternate Flow	If there are no matching results, the system informs the user and suggests modifying the search or filter criteria.
Exception Flow	If there are technical issues during the search/filter process, the system displays an error message and advises the user to try again later.
Post-conditions	The user sees a list of blog posts that match the specified search or filter criteria.

2.8   Manage User Account

	Manage User Account
Actors	- Logged-in User
Description	A logged-in user wants to manage various aspects of their user account, including updating personal information, changing account settings, and reviewing account activity.
Pre-conditions	The user is logged into their account.
Flow	1. User navigates to the "Account Settings" or "Manage Account" section.
	2. User is presented with options to manage different aspects of their account, such as:
	● Update Personal Information:
	2.1 User selects the option to update personal information.
	2.2 User modifies relevant details like name, email, or profile picture.
	2.3 System validates the updated information.
	2.4 If validation is successful, the system updates the user's account information.
	● Change Password:
	2.1 User selects the option to change the password (see the "Change Password" use case for detailed steps).
	● Notification Preferences:
	2.1 User selects the option to manage notification preferences.
	2.2 User specifies the types of notifications they want to receive.
	2.3 System updates the user's notification settings accordingly.
	● Review Account Activity:
	2.1 User selects the option to review account activity.
	2.2 System displays a log of recent account activity, such as login history.
	● Deactivate Account:
	2.1 User selects the option to deactivate their account.
	2.2 System prompts the user to confirm the account deactivation.
	2.3 If confirmed, the system deactivates the user's account.
Alternate Flow	If the user encounters any issues during the management process, the system provides appropriate error messages and guidance.
Exception Flow	If the system encounters technical issues during any account management operation, it notifies the user about the problem and advises them to try again later.

Post-conditions	The user's account information is updated, notification preferences are modified, account activity is reviewed, or the account is deactivated based on the selected option.

2.9   Guest View Profiles

Actors	Guest user
Description	The Guest View Profile use-case defines the interactions and functionalities available to users who are not logged in to the blogging website.
Pre-conditions	The user is not logged in. The user is accessing the blogging website.
Flow	
Exception Flow	If certain content is restricted to registered users, the guest user is prompted to log in or register when attempting to access that content. Some features, like saving favorite posts or creating custom feeds, are only available to registered users.
Post-conditions	The guest user can explore, read, and share blog posts without creating an account. The guest user cannot perform actions that require authentication, such as liking, commenting, or creating blog posts.

2.10   Author Manage Posts

Actors	Author
Description	The Author Manages Posts use-case outlines the interactions and functionalities available to a registered author for managing their blog posts on the website.
Pre-conditions	The author is logged in to the blogging website. The author has created one or more blog posts.
Flow	1. Dashboard: Upon logging in, the author is directed to their author dashboard. 2. View Posts: The author can view a list of all their blog posts on the dashboard. Each post is displayed with relevant information such as title, publication date, and status. 3. Edit Post: The author can click on a post to enter the editing mode. In edit mode, the author can modify the post content, update the title, or make other changes. 4. Delete Post: The author can choose to delete a post from the dashboard.

	A confirmation prompt may be displayed before permanently removing the post.
	5. Drafts:
	The author can view and manage draft posts separately.
	Draft posts may include works in progress that haven't been published yet.
	6. Publish Post:
	The author can choose to publish a draft post, making it accessible to website visitors.
	7. View Published Posts:
	The author can see a list of all their published posts.
	8. Logout: The author can log out when done managing their posts.
Exception Flow	If the author attempts to edit or delete a post that is currently being edited by another author, a notification or lock mechanism may prevent conflicts.
	The author may encounter error messages if there are issues with saving changes or publishing a post.
Post-conditions	The author can efficiently manage and update their blog posts through the author dashboard.

2.11   Admin Manage Posts

	Admin Manage Posts
Actors	Admin
Description	The Admin Manages Posts use-case outlines the interactions and functionalities available to an admin user for managing blog posts on the website.
Pre-conditions	The admin is logged in to the blogging website.
	The admin has administrative privileges.
Flow	1. Admin Dashboard: Upon logging in, the admin is directed to the admin dashboard.
	2. View All Posts: The admin can view a list of all blog posts on the website. Each post is displayed with relevant information such as title, author, publication date, and status.
	3. Edit Any Post: The admin can click on any post to enter the editing mode. In edit mode, the admin can modify the post content, update the title, or make other changes.
	4. Delete Any Post: The admin can choose to delete any post from the website. A confirmation prompt may be displayed before permanently removing the post.
	5. View Drafts: The admin can view and manage draft posts. This includes viewing, editing, publishing, or deleting drafts.
	6. Approve/Reject Posts (if moderation is required): If there is a moderation process, the admin can approve or reject posts submitted by authors.
	7. View Published Posts: The admin can see a list of all published posts on the website.
	8. View Reports (if applicable): If there are reporting features, the admin can view statistics or reports related to blog posts.

	9. Logout: The admin can log out when done managing posts.
Exception Flow	The admin may encounter error messages if there are issues with saving changes or deleting a post. If there are restrictions or special permissions on certain posts, the admin may need additional authorization.
Post-conditions	The admin has the ability to manage all blog posts on the website, ensuring content quality and adherence to policies.

2.12   Admin Manage Account

	Admin Manage Account
Actors	Admin
Description	The Admin Manages User Accounts use-case outlines the interactions and functionalities available to an admin user for managing user accounts on the website.
Pre-conditions	The admin is logged in to the blogging website. The admin has administrative privileges.
Flow	1. Admin Dashboard: Upon logging in, the admin is directed to the admin dashboard. 2. View User List: The admin can view a list of all registered users on the website. User information such as username, email, and account status is displayed. 3. Search for Users: The admin can use a search feature to find specific users based on criteria like username or email address. 4. View User Details: Clicking on a user's name or profile allows the admin to view detailed information about that user, including their activity and posts. 5. Suspend/Activate User Account: The admin can suspend or activate a user account based on policy violations, user requests, or other reasons. Suspended users may be restricted from logging in or performing certain actions. 6. Reset Password: The admin can initiate a password reset for a user, generating a temporary password for the user to regain access to their account. 7. Edit User Profile (if necessary): The admin may have the ability to edit certain details of a user's profile, such as their display name or email address. 8. View Reports (if applicable): If there are reporting features, the admin can view statistics or reports related to user activities.

	9. Logout:
	The admin can log out when done managing user accounts.
Exception Flow	The admin may encounter error messages if there are issues with suspending, activating, or editing a user account.
	Additional security measures, such as two-factor authentication for admin actions, may be implemented for sensitive operations.
Post-conditions	The admin has the ability to manage user accounts, ensuring compliance with policies and maintaining a secure user environment.

2.13 System Verify/Valid Account

	System Verify/ Valid Account
Actors	System
Description	The System Verify/Validate Account use-case outlines the interactions and functionalities performed by the system to verify and validate user accounts on the website.
Pre-conditions	A user has completed the account registration process.
	The user has received a confirmation email or taken necessary steps to verify their account.
Flow	1. User Registration: A user completes the registration process by providing necessary information.
	2. Email Verification: The system sends a verification email to the email address provided during registration. The email contains a unique verification link.
	3. User Receives Email: The user receives the verification email in their registered email inbox.
	4. Clicks Verification Link: The user clicks the verification link within the email.
	5. System Verifies Email Link: The system verifies the authenticity of the verification link. If valid, the system marks the user account as verified.
	6. Account Activation: The system activates the user account, allowing the user to log in.
Exception Flow	If the user does not receive the verification email, they may request a resend or check their spam folder.
	If the user clicks an expired verification link, the system may prompt them to request a new verification email.
Post-conditions	The user's account is verified and activated, allowing them to log in and access the website.

2.14 System Check For Existing Accounts

	System Check For Existing Accounts
Actors	System
Description	The Check for Existing Account use-case outlines the interactions and functionalities performed by the system to check if a user account already exists during the registration or login process.
Pre-conditions	The user is in the process of registering on the website
Flow	1. User Provides Registration Information:

	The user enters their registration information, including email address and/or username.
	2. System Checks for Existing Account: The system queries the user database to check if an account with the provided email address or username already exists.
	3. Existing Account Found: If an account with the provided details already exists, the system informs the user.
	4. User Adjusts Registration Details: The user is prompted to adjust their registration details, such as choosing a different username or using a different email address.
	5. Registration Continues: The user continues with the registration process after making the necessary adjustments.
Exception Flow	If there are issues with the system's database or connectivity, the check for existing accounts may be temporarily unavailable. The system should handle errors gracefully and inform the user of any issues with the registration process.
Post-conditions	The system ensures that the user does not create duplicate accounts during the registration process.

2.15 System Update User Database

	System Update User Database
Actors	System
Description	The System Update User Database use-case outlines the interactions and functionalities performed by the system to update user information in the database.
Pre-conditions	The system receives a request to update user information. The user's request complies with any relevant system constraints.
Flow	1. Update User Information Request: The system receives a request to update user information, which may include changes to the username, email, password, or other profile details.
	2. Validation of Updated Information: The system validates the updated information to ensure it meets any relevant criteria, such as length requirements, format constraints, or security policies.
	3. Check for Existing Account (Optional): If the update involves changes to the username or email, the system may check for the existence of another account with the updated information to avoid conflicts.
	4. Update Database: If the updated information is valid and there are no conflicts, the system proceeds to update the user's information in the database.
	5. Database Update Confirmation:

	The system provides a confirmation message indicating that the user's information has been successfully updated. adjustments.
Exception Flow	1. Validation Error: If there are issues with the validation process, the system rejects the update request and provides an error message to the user, specifying the reason for rejection (e.g., invalid password format). 2. Conflicts Detected: If conflicts arise during the check for existing accounts (e.g., attempting to update to an existing username or email), the system prompts the user to choose a different identifier to avoid duplication.
Post-conditions	The user's information in the database is updated according to the changes requested.