<Group 05>

<SoulNote>
Use-Case Specification

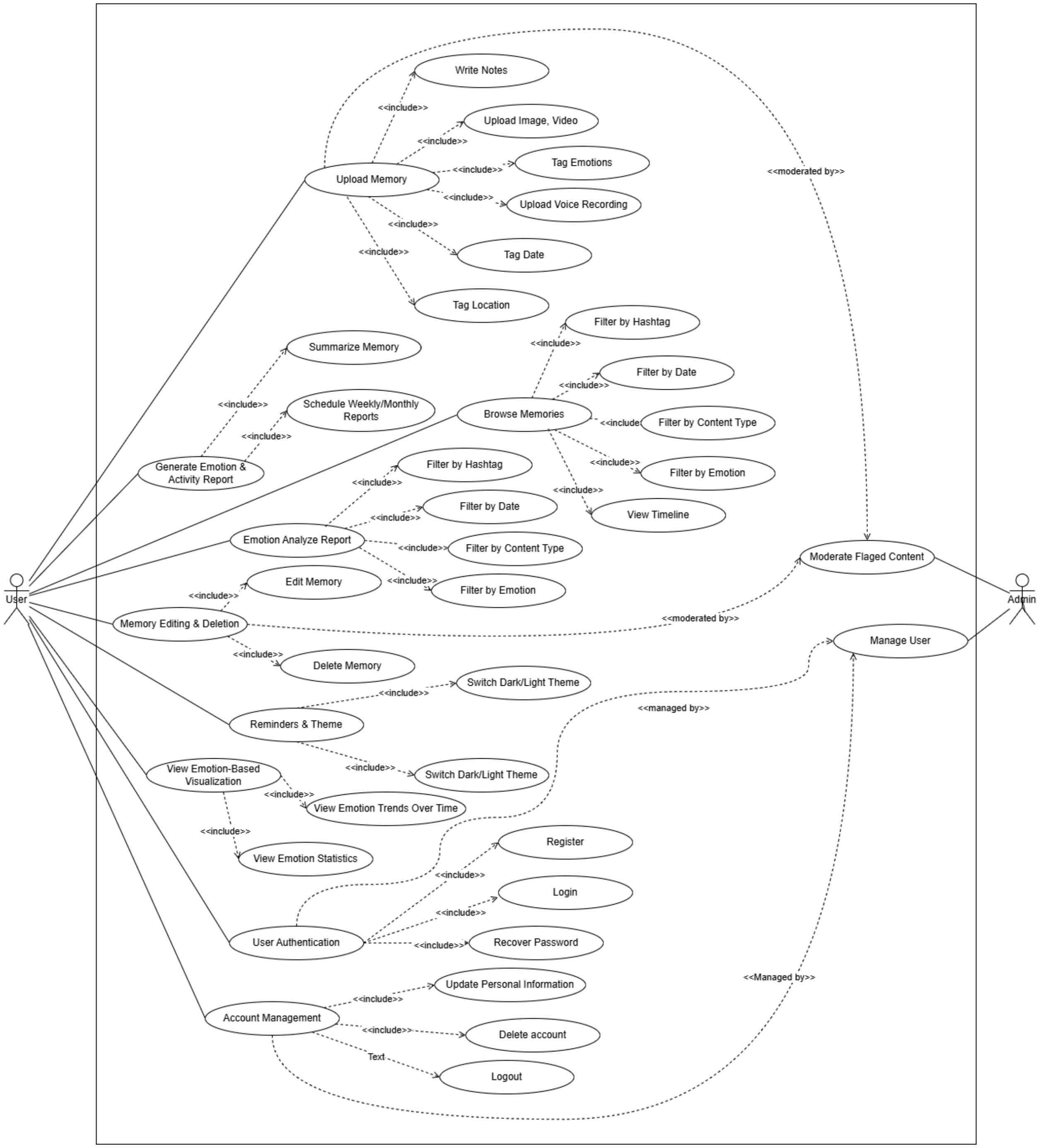
Version <1.0>

Revision History

Date	Version	Description	Author
02/06/25	1.0	Use-case Model	Ly Quoc Thanh
03/06/25	1.0	Use-case: Upload memories Use-case: Summarize memory Use-case: Schedule weekly/monthly reports Use-case: Browse memory	Nguyen Le Quang
03/06/25	1.0	Use-case: Browse memories using multiple filters Use-case: Edit memory Use-case: Delete memory Use-case: Remind uploading threads in a period	Huynh Van Sinh
03/06/25	1.0	Use-case: Switch dark/light theme Use-case: View emotion trends over time Use-case: Delete emotion statistic	Nguyen Tan Van
03/06/25	1.0	Use-case: Recover password Use-case: Update personal information Use-case: Delete account	Pham Quang Thinh
04/06/25	1.0	Use-case: Register Use-case: Log in Use-case: Log out	Nguyen Tan Van
04/06/25	1.0	Use-case: Share memory Use-case: Location Integration	Nguyen Tan Van

Table of Contents

1. Use-case Model



2. Use-case Specifications

2.1 Use-case: Upload memories

Use case Name	Upload Memory
Brief description	This use case describes how a user uploads a memory to the system with text, images, audio, and emotional tags.
Actors	User
Basic Flow	1. The user logs in and navigates to the memory creation page. 2. The user writes text (a note) for the memory. 3. The user selects files to upload images or videos. 4. The user records or uploads an audio file. 5. The user selects one or more emotional tags (e.g., happy, nostalgic, anxious). 6. The user assigns a date, location, and custom hashtags to the memory. 7. The user clicks the "Upload" button to save the memory. 8. The system confirms that the memory was successfully saved.
Alternative Flows	**Alternative Flow 1: Missing Required Information** 1. At step #2 or #5, if the user leaves out required text or emotion tags, 2. The system shows an error message requesting all required fields be completed. **Alternative Flow 2: File Upload Error** 1. At step #3 or #4, if the uploaded file is invalid or there is a connection issue, 2. The system displays an error and asks the user to try again.
Pre-conditions	The user is logged in.
Post-conditions	A new memory is successfully saved in the user's account.

2.2 Use-case: Summarize memory

Use case Name	Summarize Memory
Brief description	This use case describes how the user receives a summary of their memories by emotional trend to better understand their emotional journey.
Actors	User
Basic Flow	1. The user navigates to the emotion summary/report page. 2. The system analyzes all stored memories based on emotional tags. 3. The system displays statistical charts: emotional proportions (e.g., 40% happy, 20% anxious...). 4. The system shows a timeline of emotional trends over time. 5. The user can filter by hashtag, time period, or memory type for more specific insights. 6. The system updates and displays filtered data accordingly.

Alternative Flows	**Alternative Flow 1: No Data Available** 1. If there are no memories matching the filter criteria, the system shows a "No data found" message.
Pre-conditions	The user has saved at least one memory with emotion tags
Post-conditions	The user views an overview of their emotional journey and trends.

2.3 Use-case: Schedule weekly/monthly reports

Use case Name	Schedule Weekly/Monthly Reports
Brief description	This use case describes how a user schedules to receive weekly or monthly emotional and activity reports.
Actors	User
Basic Flow	1. The user accesses the recurring report settings page. 2. The user selects the report type: weekly or monthly. 3. The user chooses the desired day and time to receive the report. 4. The user selects the delivery method: email or in-app notification. 5. The system saves the report settings and confirms success. 6. The system sends the report on schedule with a summary of emotional trends and recent activities.
Alternative Flows	**Alternative Flow 1: Incomplete Configuration** 1. At steps #2–#4, if the user leaves required information blank, **2.** The system prompts the user to complete the configuration before saving. **Alternative Flow 2: Report Delivery Failure** 1. If the system fails to deliver the email or notification (e.g., due to connectivity issues), 2. The system logs the error and notifies the user the next time they log in.
Pre-conditions	The user has logged in and saved at least one memory.
Post-conditions	The system automatically delivers scheduled reports as configured.

2.4 Use-case: Browse memory

Use case Name	Browse Memory
Brief description	This use-case allows users to browse individual memories using specific filters such as date, hashtag, emotion, or content type.
Actors	User
Basic Flow	1. The user navigates to the “Browse Memories” section. 2. The user selects one filter type (e.g., date, emotion, hashtag, or content type). 3. The system retrieves and displays the matching memory entries.

Alternative Flows	**Alternative flow 1: No memory matches the selected filter** 1. The system informs the user that no memory matches the selected criteria. 2. The user is prompted to choose a different filter.
Pre-conditions	The user has at least one memory stored in the system.
Post-conditions	The user views memory entries filtered by the selected criteria.

2.5 Use-case: Browse memories using multiple filters

Use case Name	Browse Memories Using Multiple Filters
Brief description	This use-case allows users to search and browse memories using multiple filters simultaneously (e.g., filter by both emotion and date).
Actors	User
Basic Flow	1. The user navigates to the "Advanced Browse" or "Search with Filters" section. 2. The user selects multiple filters (e.g., emotion + date + hashtag). 3. The system processes the query and retrieves relevant memory entries. 4. The system displays the filtered memories.
Alternative Flows	**Alternative flow 1: No memory matches all selected filters** 1. The system notifies the user that no result matches the combined filters. 2. The user is prompted to remove or change some filters.
Pre-conditions	The user has at least one memory stored and has permission to use filter features.
Post-conditions	The user sees a refined list of memories matching the combination of selected filters.

2.6 Use-case: Edit memory

Use case Name	Edit Memory
Brief description	This use-case allows users to update the contents or metadata of a previously saved memory (e.g., text, image, emotion tag, date).
Actors	User
Basic Flow	1. The user navigates to the memory they want to edit. 2. The user selects the "Edit" option. 3. The user updates the desired fields (e.g., note content, emotion tags, date, media). 4. The user confirms the changes. 5. The system saves and reflects the updates.
Alternative Flows	**Alternative flow 1: The user cancels the edit** 1. The user exits the edit screen without saving changes. 2. The system discards the edits and retains the original memory.
Pre-conditions	The user is authenticated and owns the memory.

Post-conditions	The memory is updated with new information and saved to the system.

2.7 Use-case: Delete memory

Use case Name	Delete Memory
Brief description	This use-case allows users to permanently remove a memory from their collection.
Actors	User
Basic Flow	1. The user navigates to the memory they want to delete. 2. The user selects the "Delete" option. 3. The system prompts for confirmation. 4. The user confirms the deletion. 5. The system removes the memory from the database.
Alternative Flows	**Alternative flow 1: The user cancels the deletion** 1. The user dismisses the confirmation dialog. 2. The memory is retained.
Pre-conditions	The user is logged in and owns the memory.
Post-conditions	The memory is permanently removed and no longer accessible.

2.8 Use-case: Remind uploading threads in a period

Use case Name	Remind Uploading Threads in a Period
Brief description	This use case describes how the system reminds users to upload new memories (threads) periodically if they haven't done so in a set time frame.
Actors	System, User
Basic Flow	1. User logs into the SoulNote system. 2. System checks the timestamp of the user's last uploaded memory. 3. System compares it with the defined reminder interval (e.g., 7 days). 4. If the time exceeds the interval, the system generates a reminder. 5. System displays a notification or sends an email reminding the user to upload a new thread. 6. User views the reminder and may choose to upload a new memory.
Alternative Flows	**Alternative flow 1: User recently uploaded a memory** 1. From step #3, the system detects that the user has uploaded a memory within the period 2. No reminder is sent **Alternative flow 2: User disables reminder notifications** 1. From step #1, the system detects reminder notifications are disabled in user settings 2. Skip all remaining steps
Pre-conditions	User is registered and has notification permissions enabled (if applicable).

	System has access to user’s last upload timestamp.
Post-conditions	Users are reminded to upload a new memory if applicable; no action is taken if not.

2.9 Use-case: Switch dark/light theme

Use case Name	Switch Dark/Light Theme
Brief description	This use case allows the user to toggle between dark mode and light mode for the application interface.
Actors	User
Basic Flow	1. User logs into the SoulNote system. 2. User navigates to the settings or theme toggle area (e.g., top bar or settings page). 3. User selects the current theme switch (e.g., clicks “Dark Mode” or “Light Mode”). 4. System applies the selected theme immediately. 5. System saves the user’s theme preference for future sessions.
Alternative Flows	**Alternative flow 1: System fails to save preference** 1. From step #5, the system fails to persist the theme selection (e.g., due to network or database error) 2. System displays a warning 3. Theme remains changed for the current session, but reverts in the next login **Alternative flow 2: User cancels the action** 1. User decides not to switch theme after step #2 (e.g., navigates away) 2. No change is made, flow ends
Pre-conditions	User is logged in. System supports multiple UI themes.
Post-conditions	User interface is updated to the selected theme, and the preference is saved for future logins.

2.10 Use-case: View emotion trends over time

Use case Name	View Emotion Trends Over Time
Brief description	This use case allows the user to visualize their emotional patterns based on tagged memories over a selected time range.
Actors	User
Basic Flow	1. User logs into the SoulNote system. 2. User navigates to the “Emotion Trends” or “Insights” section. 3. User selects a time range (e.g., past week, month, or custom). 4. System retrieves memories and associated emotion tags within that time frame. 5. System generates a visual report (e.g., chart or graph) showing the distribution and trend of emotions over time. 6. User views and analyzes the displayed emotional trends.

Alternative Flows	**Alternative flow 1: No memories in the selected period** 1. From step #4, system finds no data in the selected time range 2. System displays a message: "No memories found in this period" 3. User can choose a different time range **Alternative flow2: System fails to load the chart** 1. From step #5, system encounters an error rendering the chart 2. System displays an error message and suggests retrying later
Pre-conditions	User is logged in. System supports multiple UI themes.
Post-conditions	The user views a graphical summary of their emotional trends over time, if applicable.

2.11 Use-case: Delete emotion statistic

Use case Name	Delete Emotion Statistic
Brief description	This use case allows the user to remove previously generated or stored emotion statistics from their profile or dashboard.
Actors	User
Basic Flow	1. User logs into the SoulNote system. 2. User navigates to the "Emotion Trends" or "Statistics" section. 3. User selects the specific emotion statistic or data set to delete. 4. System prompts the user for confirmation. 5. User confirms the deletion. 6. System removes the selected statistic from the database. 7. System updates the dashboard to reflect the change.
Alternative Flows	**Alternative flow 1: User cancels deletion** 1. From step #4, user chooses to cancel the action 2. System closes the dialog 3. No change is made **Alternative flow 2: System fails to delete statistic** 1. From step #6, a system error occurs (e.g., network or database failure) 2. System displays an error message 3. User can retry or cancel
Pre-conditions	User is logged in and has at least one emotion statistic generated.
Post-conditions	The selected emotion statistic is permanently deleted from the system and no longer appears in reports.

2.12 Use-case: Register

Use case Name	Register
Brief description	This use case allows a new user to create an account on the SoulNote platform to begin storing and managing personal memories.
Actors	New User

Basic Flow	1. New user navigates to the SoulNote registration page. 2. User enters required information (e.g., full name, email, password, confirm password). 3. User clicks the “Register” or “Sign Up” button. 4. System validates the input data. 5. System creates a new user account in the database. 6. System redirects the user to the login page or logs them in automatically. System displays a welcome message or tutorial prompt.
Alternative Flows	**Alternative flow 1: Email already registered** 1. From step #4, system detects that the email is already in use 2. System displays an error message 3. Return to step #2 **Alternative flow 2: Invalid input (e.g., weak password or mismatched confirmation)** 1. From step #4, system identifies invalid or missing fields 2. System highlights the errors 3. Return to step #2
Pre-conditions	User is not logged in and has not registered before with the same email.
Post-conditions	A new user account is created and ready for use.

2.13 Use-case: Log in

Use case Name	Log in
Brief description	This use case describes how a registered user accesses their SoulNote account using valid login credentials.
Actors	User
Basic Flow	1. User navigates to the SoulNote login page. 2. User enters their registered email and password. 3. User clicks the “Log In” button. 4. System validates the email and password. 5. If valid, the system authenticates the user. System redirects the user to their personal dashboard. 6. System displays a welcome message or recent activity.
Alternative Flows	**Alternative flow 1: Invalid credentials** 1. From step #4, system detects incorrect email or password 2. System displays error message: “Invalid email or password” 3. User returns to step #2 to re-enter credentials **Alternative flow 2: Account not verified** 1. From step #4, system detects that the user's email has not been verified 2. System displays message: “Please verify your email to continue” 3. System provides option to resend verification email **Alternative flow 3: Too many failed attempts** 1. If multiple login attempts fail consecutively 2. System temporarily locks the account or prompts CAPTCHA 3. User must wait or verify identity

Pre-conditions	User has previously registered an account with verified email and password.
Post-conditions	User is successfully authenticated and gains access to their SoulNote account.

2.14 Use-case: Log out

Use case Name	Log out
Brief description	This use case describes how the user recovers their password if they forget it.
Actors	User
Basic Flow	1. User goes to the Login page. 2. User clicks on the 'Forgot Password' link. 3. System displays the 'Recover Password' page. 4. User enters their registered email address. 5. System validates the email address. 6. If valid, the system sends a password recovery email with a reset link. 7. User receives the email and clicks on the reset link. 8. System displays the 'Reset Password' page. 9. User enters a new password and confirms it. 10. System updates the user's password and displays a success message. 11. User can now log in with the new password
Alternative Flows	**Alternative flow 1: Invalid email entered** 1. From step #5 of the basic flow, if the email is not registered, the system displays an error message *"Email not found."* 2. User can re-enter email and continue at step #4. **Alternative flow 2: User does not receive email** 1. After step #6, the user does not receive the recovery email (e.g. due to spam filters). 2. Users can request to resend the email. 3. System re-sends the recovery email. **Alternative flow 3: Reset link expired** 1. User clicks on a reset link that has expired. 2. System displays message *"Reset link has expired. Please request a new one."* 3. User is redirected to step #2 of the basic flow.
Pre-conditions	User has previously registered an account and linked a valid email address
Post-conditions	User's password is successfully updated and user can log in with the new password

2.15 Use-case: Recover password

Use case Name	Recover password
Brief description	This use case describes how the user logs out of the system.
Actors	User
Basic Flow	1. User clicks on the 'Profile' icon or menu. 2. User selects the 'Logout' option.

	3. System asks for confirmation (if applicable). 4. User confirms logout (if applicable). 5. System logs the user out and redirects to the login or homepage. 6. System displays a confirmation message that the user has been logged out successfully.
Alternative Flows	**Alternative flow 1: User cancels logout confirmation** 1. From step #3 of the basic flow, the user cancels the logout operation. 2. System returns the user to the current page without logging out. **Alternative flow 2: Session timeout auto-logout** 1. After a period of inactivity, the system automatically logs the user out. 2. System redirects to the login page and displays a session timeout message.
Pre-conditions	User is logged into the system
Post-conditions	User is successfully logged out of the system and redirected to the login or homepage. Session is invalidated

2.16 Use-case: Update personal information

Use case Name	Update personal information
Brief description	This use case describes how the user updates their personal information in their profile.
Actors	User
Basic Flow	1. User logs into the system. 2. User navigates to the 'Profile' or 'Account Settings' page. 3. System displays current personal information. 4. User clicks on the 'Edit Profile' button. 5. User modifies the desired fields (e.g. display name, avatar, bio, etc.). 6. User clicks the 'Save' button. 7. System validates the new information. 8. System updates the information in the database. 9. System displays a confirmation message *"Profile updated successfully."*
Alternative Flows	**Alternative flow 1: User cancels the update** 1. At step #5, the user decides not to save changes and clicks the 'Cancel' button. 2. System discards the unsaved changes and returns to the Profile page. **Alternative flow 2: Invalid input** 1. At step #7, the system detects invalid input (e.g. empty required field, invalid image format, etc.). 2. System displays appropriate error messages. 3. User corrects the input and continues at step #6. **Alternative flow 3: Session expired during update**

	1. If user's session expires during editing (before step #8), the system redirects the user to the Login page with *"Session expired. Please log in again."* 2. After login, the user may return to the Profile page to retry.
Pre-conditions	User is logged into the system.
Post-conditions	User's personal information is successfully updated in the system and reflected in their profile.

2.17 Use-case: Delete account

Use case Name	Delete account
Brief description	This use case describes how the user deletes their account and all associated data from the system.
Actors	User
Basic Flow	4. User logs into the system. 5. User navigates to the 'Account Settings' page. 6. User selects the 'Delete Account' option. 7. System displays a warning message about permanent deletion of account and data. 8. User confirms the deletion (e.g. via the 'Delete' button or re-entering password). 9. System validates the confirmation (and password if required). 10. System deletes the user's account and all associated data (memories, profile, etc.). 11. System displays a message *"Your account has been deleted."* and redirects to the homepage or goodbye page.
Alternative Flows	**Alternative flow 1: User cancels deletion** 3. At step #5, the user cancels the operation. 4. System aborts the deletion and returns to the Account Settings page. **Alternative flow 2: Invalid password confirmation** 1. At step #6, if password confirmation is incorrect, the system displays an error message *"Incorrect password."* 2. Users may retry or cancel the deletion. **Alternative flow 3: Session expired during process** 1. If session expires before step #7, the system redirects the user to the Login page with *"Session expired. Please log in again."* 2. After login, the user must repeat the deletion process.
Pre-conditions	User is logged into the system.
Post-conditions	User's account and all associated data are permanently removed from the system. User is logged out and cannot log in again.

2.18 Use-case: Share Memory

Use case Name	Share Memory
Brief description	This use case describes how the user shares a memory with others via a generated shareable link or through integrated platforms (e.g., email, messaging apps).
Actors	User
Basic Flow	1. User logs into the system. User navigates to the memory timeline or memory details page. 2. User selects a specific memory to share. 3. User clicks on the 'Share' button. 4. System generates a secure, unique shareable link for that memory. 5. System displays sharing options (e.g., copy link, share via email, or social apps). 6. User selects a sharing method and sends the link to recipients. 7. System logs the sharing action (for optional history or analytics). 8. System displays a success message: "Memory shared successfully."
Alternative Flows	**Alternative flow 1: Memory is private or restricted** 1. At step #4, if the memory has privacy restrictions, system shows an error: "This memory is private and cannot be shared." 2. User may update memory visibility or cancel the action. **Alternative flow 2: User cancels sharing** 1. At step #6, user chooses to cancel the operation. 2. System returns to the memory view without sharing.
Pre-conditions	User is logged into the system. The memory to be shared is not restricted by privacy settings.
Post-conditions	A unique shareable link is created and optionally shared via selected platforms. Memory may now be viewable (read-only) by anyone with the link, depending on privacy settings.

2.19 Use-case: Location Integrations

Use case Name	Location Integration
Brief description	This use case describes how the user adds a location to a memory and views memories on an interactive map based on their tagged locations.
Actors	User
Basic Flow	1. User logs into the system. 2. User creates or edits a memory. 3. User selects the location field and searches or chooses a location (e.g., via map pin or search bar). 4. System saves the selected location as metadata for the memory. 5. User navigates to the 'Map View' or 'Memories by Location' section. System displays an interactive map with markers showing memory locations. 6. User clicks on a marker to view memory details associated with that location.
Alternative Flows	**Alternative flow 1: User skips location input** 1. At step #3, user chooses not to tag a location. 2. System continues saving the memory without location metadata. **Alternative flow 2: Invalid or unrecognized location** 1. At step #3, if the entered location is invalid or unrecognized, system displays: "Location not found. Please try again." 2. User may retry or cancel location tagging. **Alternative flow 3: Map loading failure** 1. At step #6, if the map fails to load (e.g., due to poor network), system displays an error: "Unable to load map. Please check your connection."
Pre-conditions	User is logged into the system. Device or browser supports geolocation or map rendering.
Post-conditions	Memory is tagged with a valid location. User can view all location-tagged memories on the interactive map.