<Group 05>

<SoulNote>
Use-Case Specification

Version <1.1>

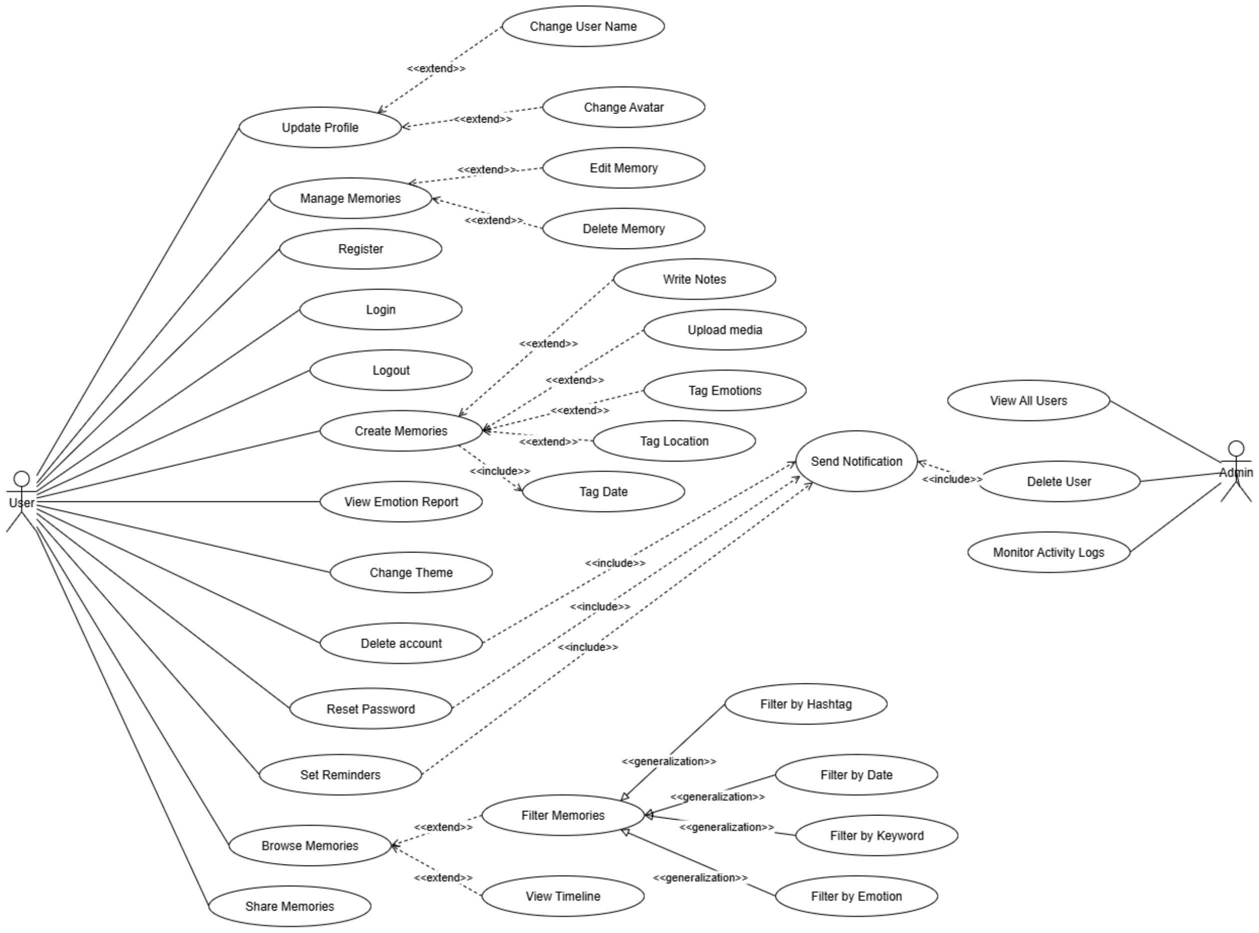
Revision History

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
17/06/25	1.1	Update Use-case model and adjust listed use cases	Ly Quoc Thanh
18/06/25	1.1	Create contents that are suitable with the new use cases (2.1 - 2.7)	Nguyen Le Quang
18/06/25	1.1	Create contents that are suitable with the new use cases (2.8 - 2.13)	Ly Quoc Thanh

Table of Contents

<SoulNote>	Version: <1.0>
Use-Case Specification	Date: <10/06/25>
<document identifier>	

1. Use-case Model



2. Use-case Specifications

2.1 Use-case: Update Profile

Use case Name	Update Profile
Brief description	This use case describes how a user updates their personal profile information such as display name, avatar, bio, or email.
Actors	User

Basic Flow	1. User logs into the SoulNote system. 2. User navigates to the “Profile” or “Account Settings” page. 3. System displays current profile details (e.g., avatar, name, bio, email). 4. User clicks the “Edit Profile” button. 5. System enters edit mode and enables input fields. 6. User modifies desired fields. 7. User clicks “Save Changes.” 8. System validates the inputs (e.g., email format, non-empty name). 9. If validation passes, system updates the profile in the database. 10. System displays a success message: “Profile updated successfully.” 11. System reflects updated information on the screen.
Alternative Flows	**Alternative flow 1: User cancels profile update** 1. From step #5 ,User clicks “Cancel.” 2. System exits edit mode and returns to the profile view with no changes saved. **Alternative flow 2: Invalid input during update** 1. From step #7, system detects invalid input (e.g., empty name or invalid email). 2. System highlights invalid fields and displays message: “Please correct the highlighted errors.” 3. User updates inputs and retries saving. **Alternative flow 3: Session expired** 1. After step #6, system detects expired session. 2. System redirects user to login page with message: “Session expired. Please log in again.” 3. After login, user must repeat the update process.
Pre-conditions	● User is authenticated and logged in.
Post-conditions	● The user's profile is updated and changes are saved to the system.

2.2 Use-case: Manage Memories

Use case Name	Manage Memories
Brief description	This use case allows users to view, create, edit, and delete their personal memories within the SoulNote system.
Actors	User
Basic Flow	1. User logs into the system. 2. User navigates to the "My Memories" or "Memory Timeline" section. 3. System displays a list/grid of saved memories. 4. User can perform the following actions: a. Create New Memory i. User clicks "+ New Memory." ii. System displays memory creation form (text, image, tags, date, location, emotion). iii. User fills in the details and clicks "Save." iv. System validates and stores the memory. v. System confirms: "Memory saved successfully." b. View Existing Memory i. User clicks on a memory card. ii. System shows full memory details. c. Edit Memory i. User clicks "Edit." ii. System enters edit mode. iii. User makes changes and clicks "Save." iv. System validates and updates memory. v. System confirms: "Memory updated successfully." d. Delete Memory i. User clicks "Delete." ii. System prompts for confirmation. iii. User confirms. iv. System deletes memory and confirms: "Memory deleted."

Alternative Flows	**Alternative flow 1: No memories exist** 1. From step #3, system finds no stored memories. 2. System displays: "No memories found. Click '+ New Memory' to start recording." **Alternative flow 2: User cancels memory creation/editing** 1. During memory form input, user clicks "Cancel". **2.** System discards changes and returns to memory timeline view. **Alternative flow 3: Invalid memory input** 1. From creation/edit step, system detects invalid fields (e.g., empty content, invalid image). 2. System highlights errors and prompts user to correct them. **3.** User corrects inputs and clicks "Save" again. **Alternative flow 4: Deletion cancelled by user** 1. From delete confirmation step, user clicks "Cancel". **2.** System aborts deletion and returns to memory view. **Alternative flow 5: Memory save fails** 1. After clicking "Save", system encounters error (e.g., network failure). 2. System displays message: "Unable to save memory. Please try again later."
Pre-conditions	● User is authenticated and has access to memory management features.
Post-conditions	● Memories are created, viewed, updated, or deleted as requested by the user.

2.3 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. The system displays the registration form. 2. User enters required information (e.g., full name, email, password, confirm password). The system performs real-time validation and highlights any input format errors (e.g., invalid email, empty fields). 3. User clicks the “Register” or “Sign Up” button. The system submits the registration data for validation. 4. The system validates the input data. It checks that all required fields are filled, verifies that the email is not already registered, and ensures the password is strong and matches the confirmation. If any validation fails, the system highlights the errors and displays appropriate messages (see Alternative Flows). 5. The system creates a new user account in the database. The user’s information is securely stored. 6. The system redirects the user to the login page or logs them in automatically. A welcome message or onboarding tutorial is displayed.
Alternative Flows	Alternative Flow 1 – Email Already Registered 1. From step 4, the system detects that the provided email is already in use. The system displays an error message: "This email is already registered." 2. The system highlights the email field for correction. The user remains on the registration form. 3. Return to step 2. The user provides a new email address and resubmits the form. **Alternative Flow 2 – Invalid Input (e.g., weak password, empty fields, mismatched confirmation)** 1. From step 4, the system detects one or more invalid inputs. The system highlights all fields that are invalid and displays

	context-specific error messages (e.g., “Password too weak”, “Passwords do not match”). 2. The user reviews and corrects the input data. The system allows the user to stay on the form. 3. Return to step 2. The user resubmits the corrected information. **Alternative Flow 3 – System Error (e.g., database failure, server error)** 1. From step 5, the system fails to create the user account due to internal error (e.g., database connection issue). The system displays a message: "Something went wrong. Please try again later." 2. The system logs the error internally for debugging. The user is returned to the registration page without losing their previously entered data. 3. Return to step 3. The user may attempt to submit the form again later.
Pre-conditions	● The user is not logged in and has not registered before with the same email.
Post-conditions	● A new user account is successfully created and stored in the system. ● The user is either automatically logged in or redirected to the login page. ● A welcome message or tutorial prompt is displayed to onboard the user.

2.4 Use-case: Login

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. The user navigates to the SoulNote login page. The system displays the login form. 2. The user enters their registered email and password. The system performs basic format validation on the input fields. 3. The user clicks the “Log In” button. The system submits the credentials for authentication. 4. The system validates the email and password. It checks whether the provided credentials match a verified account in the system. 5. If the credentials are valid, the system authenticates the user. The system redirects the user to their personal dashboard. 6. The system displays a welcome message or a summary of recent activity. The user can begin interacting with their memories.
Alternative Flows	**Alternative Flow 1 – Invalid Credentials** 1. From step 4, the system detects that the email or password is incorrect. The system displays an error message: “Invalid email or password.” 2. The user remains on the login page. The system clears or highlights the incorrect fields. 3. Return to step 2. The user re-enters their credentials and tries again. **Alternative Flow 2 – Account Not Verified** 1. From step 4, the system detects that the user’s email address has not been verified. The system displays a message: “Please verify your email to continue.” 2. The system provides a button or link to resend the verification email. The user may choose to check their inbox or request a new verification email. 3. Return to step 1. After verification, the user can attempt login again. **Alternative Flow 3 – Too Many Failed Attempts** 1. After several consecutive failed login attempts, the system triggers a

	security mechanism. The system either temporarily locks the account or presents a CAPTCHA challenge. 2. The user is informed of the lockout or verification step required. If locked, the system may indicate a wait time before retrying; if CAPTCHA is shown, the user must solve it. 3. Return to step 2. After passing the CAPTCHA or waiting, the user can attempt to log in again.
Pre-conditions	● The user has already registered an account. ● The account's email address has been verified.
Post-conditions	● The user is successfully authenticated. ● The user gains access to their personal dashboard on SoulNote.

2.5 Use-case: Logout

Use case Name	Log out
Brief description	This use case describes how the user logs out of the system.
Actors	User
Basic Flow	1. The user clicks on the "Profile" icon or menu. The system displays a dropdown or navigation menu with user options. 2. The user selects the "Logout" option. The system prepares to log the user out and may prompt for confirmation depending on the platform. 3. The system asks the user to confirm the logout action (if applicable). A dialog box appears with options to confirm or cancel. 4. The user confirms the logout request. The system proceeds with the logout process. 5. The system logs the user out and invalidates their session. The user is redirected to the login screen or homepage. 6. The system displays a confirmation message.

	The message indicates 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 chooses to cancel the logout action. The system closes the confirmation dialog without logging the user out. 2. The user remains on the current page. No changes are made to the session. 3. The flow ends. **Alternative Flow 2 – Session Timeout Auto-Logout** 1. After a predefined period of inactivity, the system automatically logs the user out. The system invalidates the session for security reasons. 2. The system redirects the user to the login page. A message is displayed stating: “Your session has timed out due to inactivity.” 3. The user must log in again to continue using the system.
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

2.6 Use-case: Create Memories

Use case Name	Create Memories
Brief description	This use case allows the user to create a new memory by adding content such as text, images, emotions, hashtags, or location.
Actors	User

Basic Flow	1. User logs into the SoulNote system. 2. User navigates to “+ New Memory” section. 3. System displays the memory creation form. 4. User enters content (e.g., text, emotion, media, tags, location). 5. User clicks “Save”. 6. System validates all inputs. 7. System saves the memory to the database. 8. System displays message: “Memory saved successfully.” 9. System redirects user to the timeline or memory view.
Alternative Flows	**Alternative flow 1: User cancels memory creation** 1. From step #3, user clicks “Cancel”. 2. System discards all inputs and returns to the previous screen. **Alternative flow 2: Invalid or incomplete input** 1. From step #6, system detects invalid fields (e.g., empty content, unsupported file). 2. System highlights errors and shows message: “Please fix the highlighted fields.” 3. User corrects and returns to step #5. **Alternative flow 3: System fails to save memory** 1. From step #7, system encounters an error (e.g., no connection). 2. System shows error: “Could not save memory. Please try again later.” 3. User retries or cancels.
Pre-conditions	● User is logged in and has access to memory creation.
Post-conditions	● The new memory is saved or the creation process is canceled.

2.7 Use-case: View Emotion Report

Use case Name	View Emotion Report
Brief description	This use case allows the user to view graphical reports of their emotional trends over a specific period.

Actors	User
Basic Flow	1. User logs into SoulNote. 2. User navigates to “Emotion Trends” or “Emotion Report”. 3. System displays time range selection (e.g., past week, month, custom). 4. User selects a time range. 5. System fetches all emotion-tagged memories within that range. 6. System renders a visual chart (e.g., line or bar graph). 7. User analyzes emotion trends via the chart.
Alternative Flows	**Alternative Flow 1: No data in selected period** 1. From step #5, system finds no matching memories. 2. System displays message: “No emotion data available for this time range.” 3. User selects a different time range. **Alternative Flow 2: Graph rendering failed** 1. From step #6, system fails to load the chart (e.g., rendering or connection issue). 2. System shows message: “Error loading chart. Please retry later.” 3. User may reload or return to home. **Alternative Flow 3: User cancels viewing report** 1. From step #3, user exits the page. 2. System returns to the dashboard without loading the report.
Pre-conditions	● User is logged in with at least one emotion-tagged memory.
Post-conditions	● The user views or attempts to view emotion trends.

2.8 Use-case: Change Theme

Use case Name	Change Theme
Brief description	This use case allows the user to switch between dark and light modes of the application interface.

Actors	User
Basic Flow	1. User logs into SoulNote. 2. User navigates to the settings or clicks the theme toggle icon. 3. System previews the alternate theme. 4. User confirms the selection. 5. System applies the new theme immediately. 6. System saves the theme preference for future sessions. 7. System displays message: “Theme updated successfully.”
Alternative Flows	**Alternative Flow 1: User cancels theme change** 1. From step #3, user exits the page or clicks “Cancel”. 2. System retains the current theme setting. **Alternative Flow 2: System fails to save theme** 1. From step #6, system encounters an error saving the preference. 2. System displays warning: “Theme will reset on next login.” 3. Theme remains applied for the current session. **Alternative Flow 3: Theme toggle not supported** 1. At step #2, system detects the user’s browser or device does not support theme toggling. 2. System displays message: “Theme switching not supported on this device.”
Pre-conditions	● User is logged in and the system supports theme toggling.
Post-conditions	● The theme is changed or the action is canceled.

2.9 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	1. The user logs into the system. The system verifies the login and displays the user's dashboard. 2. The user navigates to the "Account Settings" page. The system loads and displays the account management options. 3. The user selects the "Delete Account" option. The system opens a confirmation dialog or a dedicated deletion page. 4. The system displays a warning message about the permanent deletion of the account and associated data. The warning explains that all user memories, profile data, and content will be erased. 5. The user confirms the deletion request. This may include clicking a "Delete" button and/or re-entering the account password for confirmation. 6. The system validates the confirmation and password if required. If valid, the deletion process proceeds. If invalid, see Alternative Flow 2. 7. The system deletes the user's account and all associated data from the database. This includes memories, emotional tags, reminders, profile info, and shared links. 8. The system displays the message: "Your account has been deleted." The user is redirected to the homepage or a goodbye page, and the session is terminated.
Alternative Flows	**Alternative Flow 1 – User Cancels Deletion** 1. From step 4, the user decides to cancel the deletion operation. The system closes the confirmation dialog or exits the deletion page. 2. The user is returned to the "Account Settings" page. No changes are made to their account. **Alternative Flow 2 – Invalid Password Confirmation** 1. From step 6, the system detects that the entered password is incorrect. The system displays an error message: "Incorrect password." 2. The user remains on the deletion page. They can retry the password input or cancel the operation.

	3. Return to step 5. **Alternative Flow 3 – Session Expired During Process** 1. Before step 6, the user's session expires due to inactivity. The system redirects the user to the login page and displays the message: “Session expired. Please log in again.” 2. After logging in, the user is taken back to the homepage. They must restart the account deletion process from step 2.
Pre-conditions	● The user is logged into the system. ● A valid session is active.
Post-conditions	● The user’s account and all associated data are permanently deleted. ● The user is logged out and cannot log in again with the deleted account.

2.10 Use-case: Reset Password

Use case Name	Recover password
Brief description	This use case describes how the user recovers their password if they forget it.
Actors	User
Basic Flow	1. The user goes to the SoulNote login page. The system displays the login interface with a “Forgot Password?” option. 2. The user clicks on the “Forgot Password” link. The system navigates to the “Recover Password” page. 3. The system displays the “Recover Password” form. The user is prompted to enter their registered email address. 4. The user enters their registered email address. The system receives the input and prepares to validate it. 5. The system validates the entered email address. If the email exists in the system, it proceeds to the next step. 6. The system sends a password recovery email with a secure reset link to the provided email address.

	The system also displays a confirmation message: “If this email is registered, a reset link has been sent.” 7. The user receives the email and clicks on the reset link. The system verifies the validity of the reset link. 8. The system displays the “Reset Password” page. The user is prompted to enter a new password and confirm it. 9. The user enters a new password and confirmation. The system checks that both fields match and meet password strength requirements. 10. The system updates the user’s password in the database. A success message is displayed: “Your password has been updated.” 11. The user can now log in with the new password. The system redirects the user to the login page.
Alternative Flows	**Alternative Flow 1 – Invalid Email Entered** 1. From step 5 of the basic flow, the system detects that the entered email is not registered. The system displays an error message: “Email not found.” 2. The user remains on the recovery form. The user can re-enter a different email address and retry. 3. Return to step 4. **Alternative Flow 2 – User Does Not Receive Email** 1. After step 6, the user does not receive the recovery email (e.g., blocked by spam filters). The user checks their inbox and requests a resend. 2. The user clicks the “Resend Recovery Email” option. The system re-sends the password recovery email. 3. Return to step 7. **Alternative Flow 3 – Reset Link Expired** 1. The user clicks on a reset link that has expired or has already been used. The system detects the invalid token. 2. The system displays an error message: “Reset link has expired. Please request a new one.”

	The user is prevented from continuing with the reset process. 3. The system redirects the user to the “Forgot Password” page. Return to step 2.
Pre-conditions	● The user has previously registered an account. ● The account is associated with a valid email address.
Post-conditions	● The user's password is successfully updated. ● The user is able to log in using the new password.

2.11 Use-case: Set Reminders

Use case Name	Set Reminders
Brief description	This use case allows the user to set up reminders to upload new memories periodically.
Actors	User
Basic Flow	1. User logs into the SoulNote system. 2. User navigates to the “Settings” or “Reminder Preferences” section. 3. System displays current reminder configuration. 4. User selects reminder frequency (e.g., every 7 days, 14 days). 5. User selects delivery method (e.g., in-app notification, email). 6. User clicks “Save Settings”. 7. System validates and saves the settings. 8. System confirms with message: “Reminders updated successfully.”

Alternative Flows	**Alternative Flow 1: User cancels reminder setup** 1. From step #4, user clicks “Cancel”. 2. System discards changes and returns to previous screen. **Alternative Flow 2: Invalid reminder configuration** 1. From step #6, system detects invalid data (e.g., unsupported email format). 2. System highlights the issue and requests user correction. 3. User corrects input and returns to step #6. **Alternative Flow 3: System fails to save settings** 1. From step #7, a system error occurs. 2. System displays message: “Could not save reminder. Please try again later.” 3. User retries or exits.
Pre-conditions	● User is logged in and has access to reminder preferences.
Post-conditions	● Reminder settings are saved or the setup process is canceled.

2.12 Use-case: Browse Memories

Use case Name	Browse Memories
Brief description	This use case allows users to search and browse their saved memories using one or more filters like date, emotion, or hashtag.
Actors	User

Basic Flow	1. User logs into the SoulNote system. 2. User navigates to the “Browse Memories” section. 3. System presents two options: a. View memories as a chronological timeline. b. Apply filters to refine memory results. 4. User chooses either “View Timeline” or “Filter Memories”. If user selects “View Timeline”: 5. System displays a scrollable list of memories sorted by date. 6. User selects a memory to view details. If user selects “Filter Memories”: 5. System displays available filters: Date, Hashtag, Keyword, Emotion. 6. User selects one or more filters. 7. System validates and processes filter query. 8. System displays matching memories. 9. User selects a memory to view details.
Alternative Flows	**Alternative Flow 1: No memory matches selected filters** 1. From step 8, system finds no results. 2. System displays: “No memories found. Please try different filters.” 3. User returns to step 6. **Alternative Flow 2: Filter error or failed load** 1. From step 7, system encounters an error (e.g., network/database). 2. System displays: “Unable to load filtered results. Please try again later.” 3. User may retry or cancel. **Alternative Flow 3: User cancels browsing** 1. At any step, user navigates away or clicks “Back”. 2. System returns to dashboard with no action saved.
Pre-conditions	● User is logged in and has at least one memory saved.

Post-conditions	● User either views the timeline or filtered list of memories.

2.13 Use-case: Share Memories

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. The user logs into the system and navigates to the memory timeline or memory details page. The system displays the selected memory along with available actions. 2. The user selects a specific memory to share. The system loads the detailed view of the selected memory. 3. The user clicks on the "Share" button. The system initiates the sharing process. 4. The system checks the memory's privacy settings. If allowed, it generates a secure and unique shareable link for that memory. 5. The system displays sharing options such as copy link, share via email, or integrated messaging/social platforms. The user is presented with buttons or icons to choose a method. 6. The user selects a sharing method and sends the link to the desired recipients. The system triggers the selected action (e.g., opens email client, copies link to clipboard). 7. The system logs the sharing action for optional history or analytics purposes. This may include timestamps and platforms used. 8. The system displays a success message: "Memory shared successfully." The user remains on the memory view page.

Alternative Flows	**Alternative Flow 1 – Memory is Private or Restricted** 1. From step 4, the system detects that the memory has privacy settings preventing sharing. The system displays an error message: “This memory is private and cannot be shared.” 2. The user may choose to update the memory’s visibility settings. Alternatively, the user may cancel the sharing action. 3. Return to step 2 if user updates settings. **Alternative Flow 2 – User Cancels Sharing** 1. From step 6, the user decides to cancel the sharing operation. The system aborts the action and does not generate or send any link. 2. The user is returned to the memory view. No changes or sharing logs are recorded.
Pre-conditions	● The user is logged into the system. ● The selected memory is eligible for sharing (i.e., 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.