
<Group 05>

<SoulNote>
Software Development Plan
Version **<1.0>**

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

Revision History

| Date | Version | Description | Author |
|------------|---------|---|-------------------|
| 31/05/2025 | 1.0 | This version includes detailed requirements and initial design. | All group members |
| | | | |
| | | | |
| | | | |

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

Table of Contents

- 1. Introduction.....4**
- 2. Project Overview.....4**
 - 2.1 Project Purpose, Scope, and Objectives..... 4
 - 2.2 Assumptions and Constraints..... 5
 - 2.3 Project Deliverables.....5
- 3. Project Organization.....6**
 - 3.1 Organizational Structure..... 6
 - 3.2 Roles and Responsibilities.....6
- 4. Management Process.....7**
 - 4.1 Project Estimates.....7
 - 4.2 Project Plan..... 7
 - 4.2.1 Phase and Iteration Plan.....7
 - 4.2.2 Releases..... 8
 - 4.2.3 Project Schedule..... 13
 - 4.3 Project Monitoring and Control..... 14
 - 4.3.1 Reporting..... 14
 - 4.3.2 Risk Management..... 14
 - 4.3.3 Configuration Management.....14

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

Software Development Plan

1. Introduction

This Software Development Plan outlines the overall approach for developing the software system. It defines the purpose and scope of the project, introduces relevant terms and references, and provides a brief overview of the plan's structure. The document serves as a guide for managing and coordinating all development activities throughout the software lifecycle.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

Purpose:

The SoulNote project is designed to meet the growing need for a personal, emotionally enriched journaling platform that allows users to preserve and reflect on meaningful life experiences. The reasons for undertaking this project include:

- **Promoting Emotional Well-being:** Helping users process and revisit memories tied to emotions, fostering self-awareness and emotional resilience.
- **Preserving Personal History:** Enabling individuals to document important life moments through text, images, voice recordings, and emotional tags.
- **Encouraging Reflection and Growth:** Providing tools such as emotion-based filtering and timeline navigation to allow users to track emotional journeys over time.
- **Supporting Organization and Categorization:** Offering custom tags and themes for intuitive memory organization and retrieval.

Scope:

The scope of the SoulNote project includes the design, development, and deployment of a full-featured web-based memory journaling application. Key functionalities include:

- User account registration and authentication
- Creating and managing memory entries with text, photos, voice recordings, moods, and tags
- Viewing memories via calendar views, emotional filters, and visual timelines
- Tag management and personalized memory organization

Objectives:

The objectives of the SoulNote project are to:

- Develop a user-friendly web application that supports multimedia journaling, including text, images, and voice recordings.
- Implement emotion tagging and filtering features to help users reflect on their emotional journeys.
- Enable secure user authentication and personalized account management.
- Design intuitive interfaces for creating, organizing, and retrieving memories through tags, moods, and timeline navigation.
- Ensure responsive, accessible, and cross-platform compatibility across modern web browsers.
- Support data privacy and user control over their personal content.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

2.2 Assumptions and Constraints

Assumptions:

- The project team will have continuous access to necessary development tools and resources.
- All team members are proficient in the required technologies and methodologies.
- Stakeholders will provide timely feedback and approvals throughout the project lifecycle.
- The platform will be accessible through major web browsers and mobile devices.
- Adequate documentation and guidelines will be provided for all third-party tools and libraries used.
- End users will be willing to engage with the platform and provide valuable feedback for improvements.

Constraints:

- The project must be completed within a fixed timeline of 12 weeks.
- This is a zero-budget project with no external funding or paid services.
- The team consists of 5 members, and no additional personnel will be added during the project.
- Only free or open-source development tools and platforms (e.g., Firebase, GitHub) may be used.
- All development and testing must be conducted within the academic semester timeframe.

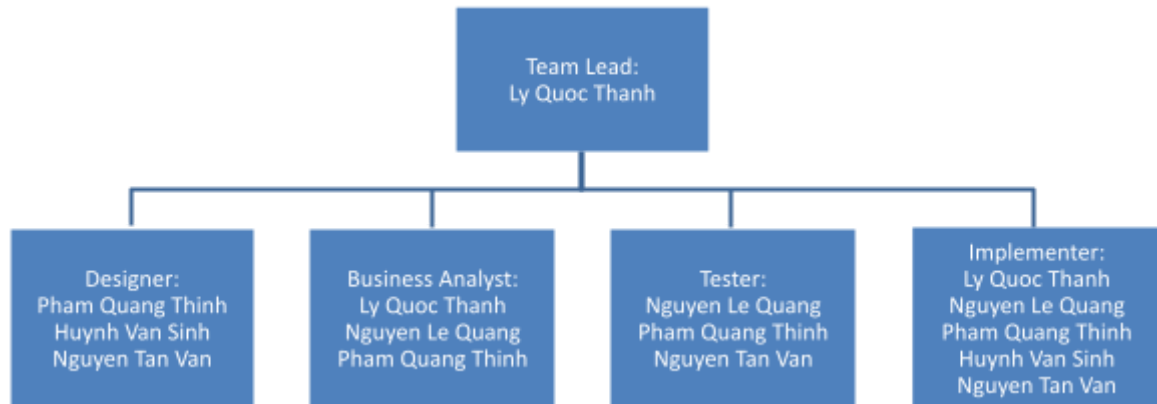
2.3 Project Deliverables

Deliverables for each project phase are identified in the Development Case. Deliverables are delivered towards the end of the iteration, as specified in section [4.2.3 Project Schedule](#).

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

3. Project Organization

3.1 Organizational Structure



3.2 Roles and Responsibilities

| Role | Person | Responsibilities |
|------------------|--|--|
| Team Lead | Ly Quoc Thanh | <ul style="list-style-type: none"> - Write the project plan (including task assignments) - Monitor and report project status weekly - Coordinate meetings - Main contact point for the project |
| Business Analyst | Ly Quoc Thanh, Nguyen Le Quang, Pham Quang Thinh | <ul style="list-style-type: none"> - Document requirements (e.g., interview customers) - Communicate requirements with team members - Coordinate to review requirements |
| Designer | Pham Quang Thinh, Huynh Van Sinh, Nguyen Tan Van | <ul style="list-style-type: none"> - Design the system (components, classes, databases, etc.) - Design the user interface - Create the Software Architecture Document |
| Tester | Nguyen Le Quang, Pham Quang Thinh, Nguyen Tan Van | <ul style="list-style-type: none"> - Write test plans and test cases - Perform system testing |
| Implementer | Ly Quoc Thanh, Nguyen Le Quang, Pham Quang Thinh, Huynh Van Sinh, Nguyen Tan Van | <ul style="list-style-type: none"> - Write source code - Perform unit tests - Review source code |

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

4. Management Process

4.1 Project Estimates

4.2 Project Plan

4.2.1 Phase and Iteration Plan

- Phase 1: Inception (contains 1 sprint)

Duration: 2 weeks

Starting date: 19/05/2025

Ending date: 01/06/2025

| Sprint | Objectives |
|--------------------------------|--|
| 1 (19/05/2025 - 01/06/2025) | <ul style="list-style-type: none"> - Group registration - Project Proposal - Tools creation - Done planing most of the project - Vision document - 2 weekly reports on group works |

- Phase 2: Elaboration (contains 2 sprints)

Duration: 4 weeks

Starting date: 02/05/2025

Ending date: 29/06/2025

| Sprint | Objectives |
|--------------------------------|--|
| 2 (02/06/2025 - 15/06/2025) | <ul style="list-style-type: none"> - Revise and improve project documents - Schedule Sprint 2 backlog - Create use-case models - Write detailed use-case specifications - 2 weekly reports on group works |
| 3 (16/06/2025 - 29/06/2025) | <ul style="list-style-type: none"> - Revise and extend use-case specifications - Define software architecture - Model class diagrams - Design database - 2 weekly reports on group works |

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

- Phase 3: Construction (contains 2 sprints)

Duration: 4 weeks

Starting date: 14/07/2025

Ending date: 10/08/2025

| Sprint | Objectives |
|--------------------------------|---|
| 4 (14/07/2025 - 27/07/2025) | <ul style="list-style-type: none"> - Revise Software Architecture Document - Design UI prototype - Deliver working software - Prepare test plan and test cases - 2 weekly reports on group works |
| 5 (28/07/2025 - 10/08/2025) | <ul style="list-style-type: none"> - Implementation - Manual Testing - Automated Testing - 2 weekly reports on group works |

4.2.2 Releases

Sprint 1 – PA0 & PA1: Group Registration, Tool Setup & Project Proposal:

Objectives:

- Form project teams and complete group registration.
- Propose and define a software product idea.
- Set up essential collaboration and development tools: Slack, Trello, GitHub, JIRA, Moodle.

Deliverables:

- Completed Group Registration Form (Google Sheet).
- **Project Proposal** including:
 - Product description and motivation.
 - Target users and platforms.
 - Key features.
 - Evidence of interviews with at least 5 potential users (e.g., recordings, notes, profiles).
- **Tooling Setup:** Team Slack channel, JIRA board, GitHub repository with correct folder structure.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

Outcomes:

- The team was successfully registered with all required member information.
- The project idea was clearly defined and documented.
- All required tools were fully set up and actively used.

Sprint 2 – PA2: Requirement Specification and Design Foundation:

Objectives:

- Refine the project plan and vision document based on feedback.
- Specify detailed requirements and use-case modeling.
- Schedule Sprint 2 tasks using JIRA with proper time estimation and task assignment.

Deliverables:

- **Revised Project Plan & Vision Document**
 - Incorporated TA feedback and added more detailed functional/non-functional requirements.
 - Added analysis of competitors, alternatives, and user environments.
 - Revision history and changes highlighted in blue.
- **Sprint 2 Backlog**
 - Backlog items clearly defined and estimated.
 - All team members assigned at least one task.
- **Use-Case Model**
 - Diagrams created using tools like StarUML or Visio.
 - Covered all major user interactions based on functional requirements.
- **Use-Case Specifications**
 - Detailed scenarios for each use-case including basic and alternative flows, pre/post conditions.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

- **Weekly Reports**

- Updated team progress for each week during the sprint.

Outcomes:

- Requirements were clearly documented and validated through use-case modeling.
- Initial design preparation established a solid foundation for later stages.
- All team members were actively involved in planning and requirement analysis.

Sprint 3 – PA3: Architecture and Design Foundation

Objectives:

- Finalize the use-case model and specifications.
- Define the system's software architecture.
- Create class and database models based on current requirements.

Deliverables:

- **Use-Case Specification – 2nd Submission**

- Refined and expanded use-cases based on TA feedback and new requirement updates.
- Revision history updated; changes highlighted for clarity.

- **Software Architecture Document**

- Architecture follows the selected style (e.g., MVC, n-tier, microservices).
- Identified key components, their interactions, and technologies used (e.g., languages, frameworks).
- Answered how components are connected and deployed.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

- **Class Diagrams**

- Modeled core classes based on architectural components.
- Included attributes, methods, and relationships among classes.
- Updated Section 4.x in the architecture document.

- **Database Design**

- Designed the ER model with entities, attributes, and relationships.
- Integrated into the DB section of the architecture document.

- **Weekly Reports**

- Progress logged from the end of PA2 to submission time using the provided template.

Outcomes:

- The team established a clear architectural foundation.
- System structure and components were well-defined for the next development phase.
- Database and class-level designs aligned with system requirements.

Sprint 4 – PA4: UI Design, Deployment & Testing

Objectives:

- Finalize system design and structure.
- Implement core functionality.
- Build UI mockups for key scenarios.
- Plan, execute, and report on functional testing.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

Deliverables:

- **Revised Software Architecture Document (SAD)**
 - Updated Sections 1–4 based on TA feedback and new project insights.
 - Added Section 5: UML deployment diagram describing system nodes and network connections (e.g., client-server or Android-only setups).
 - Added Section 6: Code folder structures for components such as Web server, client, etc.
- **UI Prototype**
 - Designed the main screen and at least 3 key scenario screens using tools like Figma, Moqups, or Visily.
 - Documented all screens and usage descriptions.
- **Working Software**
 - Developed and delivered a working version of at least one main use-case.
 - Demonstrated functionality aligned with previous specifications and requirements.
- **Test Plan & Test Cases**
 - Completed a comprehensive test plan including all required sections.
 - Selected 3 use-cases and created 15+ functional test cases (minimum 5 per use-case).
 - Executed test cases and recorded outcomes in a test report:
- **Weekly Reports**
 - Weekly progress from the end of PA3 to submission time documented using the provided template.

Outcomes:

- System design is now complete with clear deployment and folder structure.
- Main use-case implementation verified through hands-on testing.
- UI design closely reflects use-case flows, improving usability and planning for the next sprint.
- Software quality ensured through testing and documentation.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

4.2.3 Project Schedule

| Phase | Iteration | Start date - Due date | Task | Result | Releases |
|--------------|-----------|---|---|--|---|
| Inception | Sprint 1 | 19/05/2025 - 01/06/2025 (2 weeks) | - Group registration - Write project proposal - Set up working tools - Plan initial project timeline and tasks - Create initial vision document - Write 2 weekly group reports | - Project proposal - Vision document - Project plan - Weekly report 1 - Weekly report 2 - PA0 and PA1 | - Project proposal - Vision document |
| | | | | | |
| Elaboration | Sprint 2 | 02/06/2025 - 15/06/2025 (2 weeks) | - Revise and improve project documents - Create and schedule Sprint 2 backlog - Create use-case models - Write detailed use-case specifications - Learning and training technology, tools - Write 2 weekly group reports | - Updated project proposal - Revised project plan - Revised vision document - Use-case diagram - Use-case specifications - Weekly report 3 - Weekly report 4 - PA2 | - Use-case diagram - Use-case specification document |
| | | | | | |
| Construction | Sprint 3 | 16/06/2025 - 29/06/2025 (2 weeks) | - Revise and extend use-case specifications - Define software architecture - Model class diagrams - Learning technology and implement code - Design project database - Write 2 weekly group reports | - Extended use-case specs - Software architecture diagram - Class diagram - Initial database design - Weekly report 5 - Weekly report 6 - PA3 | - Class diagram - Software architecture design - Database design document |
| | | | | | |
| Construction | Sprint 4 | 14/07/2025 - 27/07/2025 (2 weeks) | - Revise software architecture document - Design UI prototype - Deliver first working version of software - Prepare test plan and test cases - Write 2 weekly reports | - Updated software architecture document - UI prototype - Working software prototype - Test plan and test cases - Weekly report 7 - Weekly report 8 - PA4 | - Software prototype v1 - UI prototype - Test plan document |
| | | | | | |
| Construction | Sprint 5 | 28/07/2025 - | - Implement core features - Perform manual testing | - Implemented features - Test reports (manual & | - Working software version |
| | | | | | |

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

| | | | | | |
|--|--|-------------------------|---|---|-----------------------|
| | | 10/08/2025 (2 weeks) | - Perform automated testing - Prepare for final submission - Write 2 weekly reports | automated) - Weekly report 9 - Weekly report 10 - Final submission - PA5 | - Test result reports |
|--|--|-------------------------|---|---|-----------------------|

4.3 Project Monitoring and Control

4.3.1 Reporting

- **Weekly Meeting:** Our team holds one main meeting in the middle of each week after all members have read and understood the assignment requirements. During this meeting, we assign tasks and clarify responsibilities. Additionally, we conduct one or two meetings at the end of the week to review the progress, check the completed work, and make necessary adjustments based on team discussions.
- **Weekly Status Report:** At the end of our weekend meeting, we upload all completed tasks to GitHub. These tasks are reviewed by the team leader and the TA before being officially used. After that, we prepare a weekly status report, which includes: achievements, challenges and next week's goals.
- **Informal Chats:** We use Messenger group chat and Discord for quick updates, questions, and informal discussions between team members. This helps us stay connected and resolve minor issues without waiting for official meetings.

4.3.2 Risk Management

| <i>Risk ID</i> | <i>Risk Description</i> | <i>Probability</i> | <i>Impact</i> | <i>Risk Exposure</i> | <i>Priority</i> | <i>Mitigation Strategy or Contingency Plan</i> |
|-----------------------|--------------------------------|---------------------------|----------------------|-----------------------------|------------------------|---|
| | | | | =Probability * Impact | | |
| | | | | | | |
| | | | | | | |

4.3.3 Configuration Management

- **Google Drive:** Used for storing and sharing project documents and files collaboratively.
- **Google Docs:** Used for real-time editing and collaboration on documents such as reports and plans.
- **Git:** Used for version control of source code to track changes and manage code history.
- **GitHub:** Used as the remote repository to host the Git codebase, facilitate collaboration, and conduct code reviews.

| | |
|---------------------------|------------------|
| SoulNote | Version: <1.0> |
| Software Development Plan | Date: 31/05/2025 |
| <document identifier> | |

- **Trello:** Used to manage tasks, organize backlog items, and track team progress.
- **Google Meet / Messenger:** Used for communication, online meetings, and quick discussions among team members.