<Group 03>

<Memoir>
Software Development Plan
Version <1.4>

Revision History

Date	Version	Description	Author
30/05/2025	1.0	1. Introduction 2. Project Overview (and its subsection)	Phạm Nam Huyên
31/05/2025	1.1	3. Project Organization	Nguyễn Gia Huy
31/05/2025	1.2	4. Management Process	Bùi Hải Long
01/06/2025	1.3	Fixing issues	Bùi Hải Long Nguyễn Thị Trà My
02/06/2025	1.4	Adding missing team role	Bùi Hải Long Nguyễn Gia Huy

Memoir	Version: <1.4>
Software Development Plan	Date: <2-June-2025>
<document identifier>	

Table of Contents

Software Development Plan

1.   Introduction

- Software Development Plan (SDP), describes all processes, step by step, that will be required to bring the project to life, covering aspects like planning, ideation, development, documentation, deployment, and sometimes maintenance.

- The main idea behind this document is to gather as much information as possible before actual work begins in order to exercise maximum control over the project. It describes the approach the development of the software will undertake and should be viewed as the primary plan to direct the development effort.

2.   Project Overview

2.1   Project Purpose, Scope, and Objectives

2.1.1 Purpose:

The Memoir project was built to help users **remember and manage personal relationships effectively**. In the digital age, forgetting information such as names, birthdays, or memories with friends, relatives, or colleagues is a common thing. Memoir will help users store and organize this important information, thereby maintaining and strengthening social connections. In addition, the application is also suitable for fields such as education, human resources, or team management - where remembering information about people is essential.

2.1.2 Scope:

Within the scope of the project:
- Develop a mobile application (Android or cross-platform).
- Personal information storage feature: name, photo, date of birth, meeting place, notes, relationship type, etc.
- Automatic reminder system for important events.
- User-friendly interface, easy to use and personal customization.
- Classify contacts by tags and display by relationship chart.
- Synchronize data between devices and cloud backup.
- Public note for users to share notes to everyone.
- Administrative functions for admin: account management, user statistics, feedback processing.

Out of the scope of the project: *(not implemented in this phase)*
- Do not develop a web version for uses
- Do not integrate with social networks (Facebook, Zalo, ...)
- Do not use advanced artificial intelligence (AI) in the MVP phase

2.1.3 Objectives:

- Build a minimum viable product (MVP) within **1 semester**, including core features:
  - Detailed contact information management.
  - Event calendar and reminder system.
  - Contact classification by tags.
  - Data synchronization and cloud backup.

- ○ Note sharing
- ○ Relationship visualization by charts.
- ● Ensure:
  - ○ The application is user-friendly and easy to use on mobile devices.
  - ○ At least 90% of test users are satisfied with the main features.
  - ○ The system operates stably with at least 50 test users.
- ● Complete deliverables:
  - ○ Complete mobile application.
  - ○ User manual.
  - ○ Report detailing the project development process.

2.2 Assumptions and Constraints
Assumptions:

1. This app will mainly be used for people with a wide network, people who don't have a good memory, or anyone else who wants to organize their professional/intimate relationships and keep the memo.
2. Users will have to authentify their accounts via email or social network accounts.

Constraints:

1. **Budget:** limited to the cost of CH Play store, hire API keys for some features, hire cloud hosting to maintain database
2. Staff: 5 students in group 3
3. Equipment: CH Play store will be used to promote this app.
4. Schedule: This app aims to launch by a specific target date. (will be determined based on the project schedule).
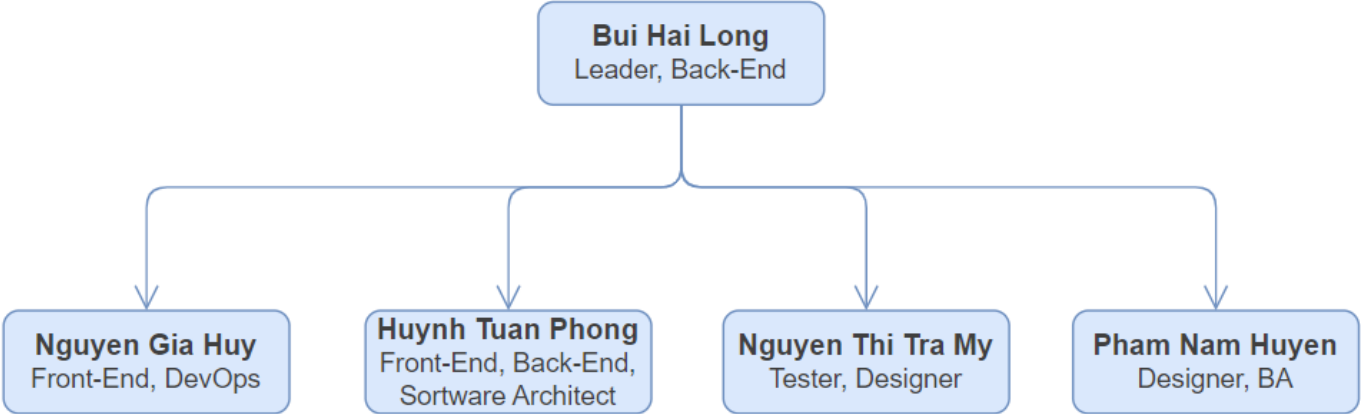
2.3 Project Deliverables
    - This app will deliver a comprehensive taking-note app on a mobile platform, including functionalities such as a user account system, account configuration, and admin management tools. The timeline and specific delivery dates are outlined in the 4.2 section below.

3. Project Organization

3.1 Organizational Structure

3.2 Roles and Responsibilities

Person	Role
Bui Hai Long	**Disciplines:** Leader, Back-End developer **Workflow details:** Overall project management, including task planning and responsibility assignment. Monitoring progress and ensuring product quality. Responsible for the development of back-end architecture, server-side logic, APIs, and database schemas. Evaluates and selects a suitable backend server or hosting platform to support the application's infrastructure needs. **Support processes:** Coordinates with all team members to ensure steady project progress and high-quality deliverables. Collaborates closely with front-end developers on API contracts and integration points. Works with DevOps (or handles DevOps-related tasks when needed) for deployment, infrastructure setup, and selecting the appropriate server or backend-as-a-service platform for reliable and scalable backend services.
Huynh Tuan Phong	**Disciplines:** Software Architecture, Front-End & Back-End **Workflow details:** Oversees and contributes to the development of both front-end and back-end components, ensuring scalability, performance, and security. Makes key technical decisions, provides technical guidance to the development team, and ensures cross-functional collaboration between front-end and back-end efforts. **Support processes:** Creates and maintains architectural documentation, conducts technical feasibility studies, mentors developers on architectural best practices
Nguyen Gia Huy	**Disciplines:** Front-End, DevOps **Workflow details:** Develop the user interface based on design specifications. Ensure interactivity, performance, and cross-platform compatibility. Designs, implements, and manages Continuous Integration/Continuous Deployment (CI/CD) pipelines. Monitors application and infrastructure health and performance **Support processes:** Work with designers to implement accurate UI and coordinate with testers to resolve UI-related issues. Troubleshoots build and deployment issues, and implements infrastructure as code practices

Nguyen Thi Tra My	**Disciplines:** Tester, Designer
	Writes detailed test plans and test cases, executes system and post-fix testing, and reports results. Provides feedback on usability and requirement clarity, ensuring adherence to quality standards throughout the development lifecycle. Additionally contributes to UI design efforts, specifically focusing on the admin interface, helping align design with functional and user experience goals.
	Support processes: Coordinates with developers to track, report, and verify bug fixes. Maintains test environments and actively contributes to improving testing processes. Help in design, review and refine the admin-side user interface for better usability and consistency.
Pham Nam Huyen	**Disciplines:** Designer, BA
	Workflow details: Gathers, analyzes, and documents user requirements through interviews. Design user interface with a focus on aesthetics and user experience. Propose layout, color schemes, and graphical elements, conducts design reviews, and ensures designs align with user needs
	Support processes: Collaborate with front-end developers to ensure accurate implementation of the UI design. Maintains requirements traceability and supports usability testing by providing design context and gathering feedback

4. Management Process

4.1 Project Estimates

- Labor Cost

● Total Hours: Each member works 15 hours per week.
● Weekly Cost per Member: 15 hours * 200.000 VND/hour = 3.000.000 VNĐ
● Total Labor Cost for teams (10 weeks): 3.000.000 VND * 5 members * 10 weeks = 150.000.000 VNĐ

- Database and upload:

● Database: 300.000 VND/ month
● Upload on CHplay store: 25 USD ≈ 650.000 VNĐ

4.2 Project Plan

4.2.1 Phase and Iteration Plan

The project is planned over a duration of 10 weeks and structured into **three main phases**: Inception, Elaboration, and Construction. The timeline is further divided into **five sprints**, each lasting **2 weeks**, with clearly defined objectives and milestones.

Phase Plan:

Phase	Timeline	Main Objective
Inception phase	● **Start Date:**: 19/05/2025 ● **End Date**: 01/06/2025	Finalize project proposal and gather initial requirements
Elaboration phase	**Sprint 2:** ● **Start Date:** 02/06/2025 ● **End Date:** 15/06/2025 **Sprint 3:** ● **Start Date:** 16/06/2025 ● **End Date:** 29/06/2025	● **Sprint 2:** Refine detailed requirements and develop the initial system design. ● **Sprint 3:** Finalize system architecture, complete detailed design, and create the test plan.
Construction phase	**Sprint 4 & 5:** ● **Start Date:** 30/06/2025 ● **End Date:** 10/08/2025	System implementation, integration, and testing

Iteration Plan

Sprint 1: Inception

- **Start Date:** 19/05/2025
- **End Date:** 01/06/2025
- **Primary Objective:** PA1 – Project Proposal and Planning
- **Deliverables:**
  - Project proposal
  - Initial high-level requirements gathering (PA0, PA1)
  - Project plan and timeline scheduling
  - Vision document

Sprint 2: Requirement Refinement & Prototyping Setup

- **Start Date:** 02/06/2025
- **End Date:** 15/06/2025
- **Primary Objective:** PA2 – Requirement Specification & Prototype Kickoff
- **Management:**
  - Revised project plan and vision document (2nd submission)
  - Sprint backlog for Sprint 2
  - Use-case model and specifications
  - Weekly report
- **Development Team:**
  - Learn Flutter framework
  - Research backend technology stack
  - Prototype: Authentication module
  - Prototype: Note-taking functionality
- **UI/UX Team:**
  - Learn Figma
  - Create initial UX design
  - Prototype: Home page and note-taking screen

Sprint 3: Architecture Design & Core Prototype Development

- **Start Date:** 16/06/2025
- **End Date:** 29/06/2025
- **Primary Objective:** PA3 – Software Architecture & Functional Prototypes
- **Management:**
  - Use-case specification (2nd submission)
  - Define software architecture and class diagrams
  - Database design
  - Weekly report
- **Prototype Development:**

- ○ Database operations: send/receive text, images, and custom file formats
  - ○ Synchronization features
  - ○ Tagging and filtering system
  - ○ Calendar and notification system
- **Backend Development:**
  - ○ Server-side storage
  - ○ Admin features: server management console
- **Test Development:**
  - ○ Write initial test plan
- **UI/UX Design:**
  - ○ Tagging, filtering, and calendar UI
  - ○ Initial integration of graph and map-based views
- **Sprint Release:**
  - ○ Core feature prototype

Sprint 4: System Development & UI Integration

- **Start Date:** 14/07/2025
- **End Date:** 27/07/2025
- **Primary Objective:** PA4 – Functional Software and UI Enhancements
- **Management:**
  - ○ Revise software architecture documentation
  - ○ Write test plan and test cases documentation
  - ○ Weekly report
- **Frontend Development:**
  - ○ Implement graph-based note visualization
  - ○ Integrate map view
  - ○ Implement note sharing functionality
- **Backend Development:**
  - ○ Enable server-side backup
  - ○ Admin feature: user account management
  - ○ Backend for note sharing
- **Test Development:**
  - ○ Write detail test case and report
- **UI/UX Design:**
  - ○ Admin interface: server controls, user management, reports, and statistics
- **Sprint Release:**
  - ○ Demo version

Sprint 5: Final Implementation & Testing

- **Start Date: 28/07/2025**
- **End Date: 10/08/2025**

- **Primary Objective: PA5 – Final Release and Presentation**
- **Management:**
  - Final implementation
  - Weekly report
  - Final presentation preparation
- **Frontend Development:**
  - Feedback and quality assurance system
- **Backend Development:**
  - Admin dashboard: reports and analytics
- **Test Development:**
  - Design and execute test cases
  - Automated testing setup
- **UI/UX Team:**
  - Prepare visuals and interfaces for final presentation
- **Sprint Release:**
  - Production-ready release

4.2.2   *Releases*

Throughout the project lifecycle, we deliver multiple releases aligned with key milestones. Each release incrementally adds functionality and refinement, progressing from early concept validation to the final production-ready version.

Release 1: **Prototype Release (End of Sprint 3)**

- **Type:** Alpha / Proof-of-Concept
- **Date:** 29/06/2025
- **Description:**

This release demonstrates core system functionality and validates foundational architecture decisions. It includes basic note-taking, database connectivity (text/image/custom file), tag/filter features, and early UI mockups for calendar and map integration. Intended for internal review and technical validation.

Release 2: **Demo Release (End of Sprint 4)**

- **Type:** Demo / Internal Beta
- **Date:** 27/07/2025
- **Description:**

A more feature-complete version showcasing major user-facing functions, including note sharing, graph and map views, calendar integration, and admin management interface. It is used for internal testing, team feedback, and mid-project assessment. Supports UI testing and begins system integration.

Release 3: **Final Release (End of Sprint 5)**

- **Type:** Production / Final Release
- **Date:** 10/08/2025

Memoir	Version: <1.4>
Software Development Plan	Date: <2-June-2025>
<document identifier>	

● **Description:**

The production-ready version of the software. Includes all planned features, finalized UI/UX, fully functional backend, admin statistics/reporting, and user feedback systems. Accompanied by a comprehensive testing report, documentation, and project presentation.

4.2.3 Project Schedule
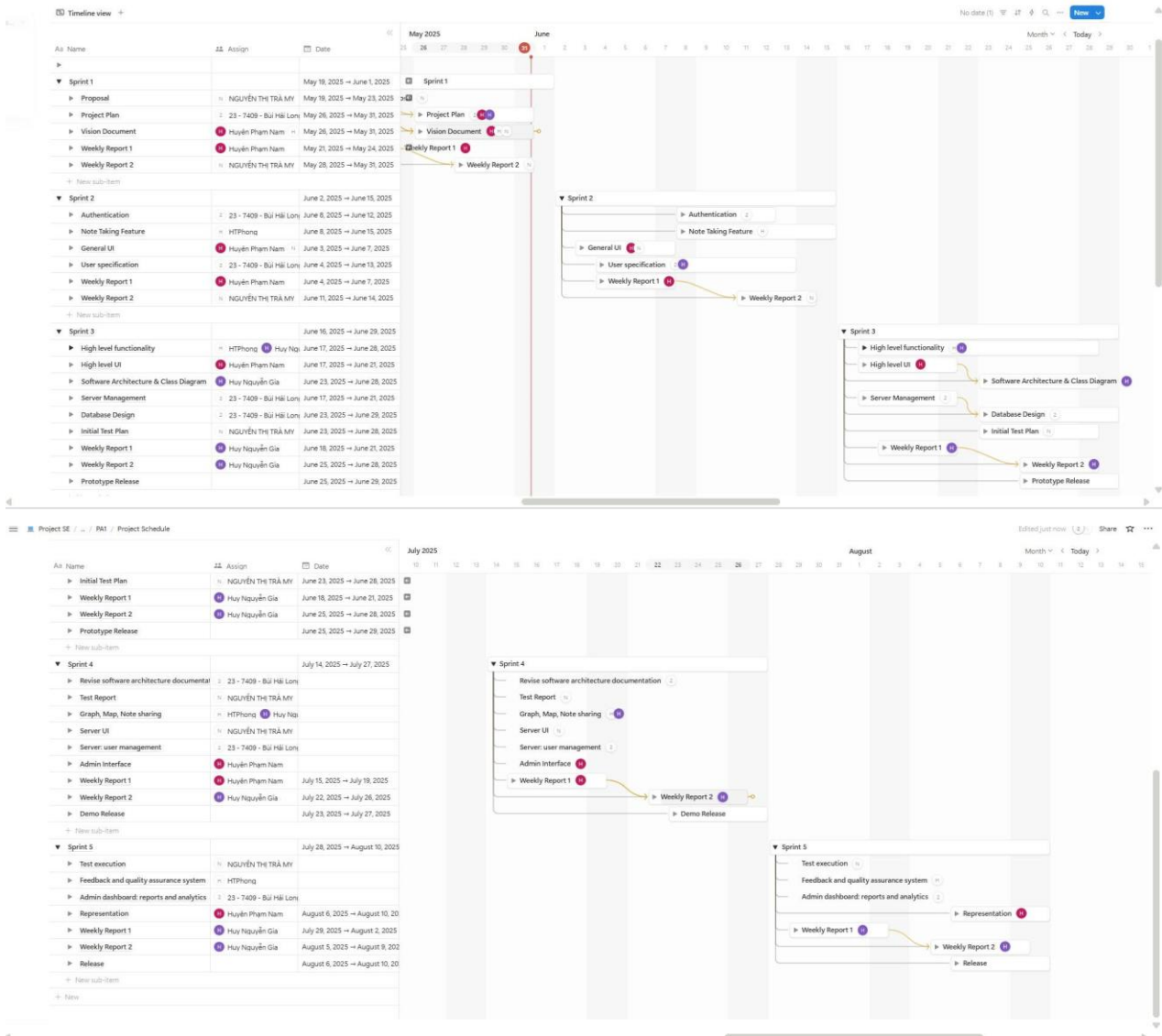
4.3 Project Monitoring and Control

4.3.1 Reporting

To ensure effective project tracking and team coordination, we adopt a combination of structured reporting, informal communication, and visual task management.

Weekly Meeting

- **Frequency:** Once per week, every Monday
- **Format:** Meet-up (1-2 hours)
- **Agenda:**
  - Discuss what've been done last week, what to do next week.
  - Discuss how to do the PA for consistent result
  - Task assign for each member
  - Updated cost and schedule estimates
- **Goal:** Promote transparency, encourage accountability, and make quick adjustments to the sprint plan if needed.

Weekly status reports

- **Prepared by:** Assigned rotating member
- **Distributed via:** Google Docs
- **Contents:** Following the format given in the template
  - Summary of tasks completed
  - Issues and Impacts
  - Next week assignment
  - Next week goal

Informal chats

- **Tools:** Messenger, Discord,
- **Use Case:** Day-to-day coordination, quick questions, sharing progress or ideas
- **Benefit:** Keeps communication fluid between formal updates

Task boards

- **Task tracking:** Trello for visualizing progress and assigning tasks
- **Task Scheduling:** Notion for visualizing overall project scheduling.

4.3.2   *Risk Management*

Risk ID	Risk Description	Probability	Impact	Risk Exposure	Priority	Mitigation Strategy or Contingency Plan

4.3.3   *Configuration Management*

- Github: Managing source code and Formal documentation
- Google Drive: For document and file sharing