Power Rangers

AI Copilot Application
Software Development Plan
Version <1.0>

Revision History

Date	Version	Description	Author
07/11/24	<1.0>	First version of Software Development Plan document	Phùng Tố Uyên

Table of Contents

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of the Software Development Plan (SDP) is to communicate to team members and stakeholders the approach to be taken when developing software on a program and how a program manager will utilize direct resources.
The following people use the *Software Development Plan*:

- The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
- **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the **AI Copilot Application** project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans. The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.

1.3 Project Overview

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

Purpose:
Creating an AI Copilot website for individuals, B2B, and B2C businesses, enhancing productivity and providing intelligent, personalized assistance in daily tasks. This website will empower users of all backgrounds—whether individuals or businesses—by simplifying access to advanced AI capabilities, driving efficiency, creativity, and improved customer interactions without requiring technical skills.
Scope:
The project includes the design and development of a secure, user-friendly web-based AI Copilot. It includes website development, AI model integration, user experience optimization, personalization features, security, content management, and interaction logging. Additionally, the scope includes chatbot training, data processing, testing, documentation, and maintenance.
Objectives:
The main goal of this project is to create a functional and responsive AI Copilot that delivers a seamless user experience, enhances productivity, and provides valuable insights. It aims to increase user engagement, provide tailored support based on user preferences, and expand the reach of AI to a wider audience. Long-term support, feature updates, and performance monitoring are also key goals to ensure continuous improvements.

2.2 Assumptions and Constraints

Assumptions:

- **Clear objectives:** Clearly defined requirements for the AI Copilot, including core functions, user interface, and customization options.
- **AI models and tools:** Use advanced AI models and tools, such as GPT and NLP systems, to enable high-quality responses and interactions.

- **User profiles and permissions:** Provide user roles and permissions for different access levels, suitable for individual, B2B, and B2C use cases.
- **Customizable templates:** Offer templates tailored to tasks like content generation, data analysis, customer service, and more.
- I**ntegration capabilities:** Ability to integrate with popular platforms (email services, databases) for a more seamless user experience.

Constraints:

- **Budget:** Zero – budget so we need to optimize AI model costs, cloud resources, and development tools to meet financial limitations.
- **Staffing:** Availability and expertise of team members, including project managers, UI/UX designers, developers, and testers, may be limited. Project only has 5 people, there will be no more people added during the project.
- **Equipment and tools:** Access to necessary development software and testing tools may be limited by budget constraints or technical requirements.
- **Schedule:** Adherence to a strict project timeline (12 weeks), especially if aiming for a specific launch date, could limit flexibility in refining or expanding features as needed.

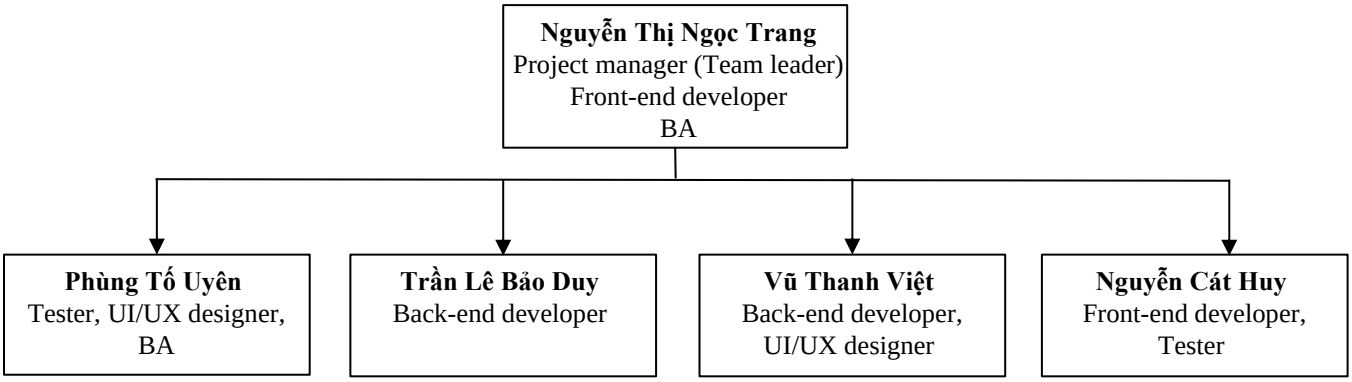
2.3 Project Deliverables

The final product is the most complete version which satisfies the user's needs. The following deliverables will be produced during the project:

- User interface prototype
- Supplementary specifications
- Business use cases
- Creative design briefs
- Navigation map
- Data model
- Design model
- Database design
- Use case survey
- Software architecture document
- Implementation subsystem
- Test package
- Change requests
- Test summary

3. Project Organization

3.1 Organizational Structure



3.2 Roles and Responsibilities

Role	Responsibilities
Project manager (Team leader)	Being responsible for projects from initiation to close, making sure the work gets done efficiently and satisfactorily: • Plan and develop the project idea • Monitor project progress and set deadlines • Solve thorny issues that arise throughout the developing process • Evaluate project performance
UI/UX designer	• Gathering and evaluating user requirements, in collaboration with product managers and engineers • Illustrating design ideas using storyboards, process flows and sitemaps • Designing graphic user interface elements, like menus, tabs and widgets
Back-end developer	• Creating, maintaining, testing, and debugging the entire back-end of the website • This includes the core application logic, databases, data and application integration, API, and other processes taking place behind the scenes
Front-end developer	• Implementing visual and interactive elements that users engage with • Creating user interfaces, test websites' usability, troubleshoot coding issues and change interfaces
Tester	• Testing websites for usability and reporting bugs or usability issues to developers • Reporting on website performance and user experience • Reporting any security risks or website errors to the appropriate department for resolution

Person	Roles

Nguyễn Thị Ngọc Trang	Project manager (Team leader), Front-end developer
Phùng Tố Uyên	Tester, UI/UX designer
Nguyễn Cát Huy	Front-end developer, Tester
Trần Lê Bảo Duy	Back-end developer
Vũ Thanh Việt	Back-end developer, UI/UX designer

4. Management Process

4.1 Project Estimates

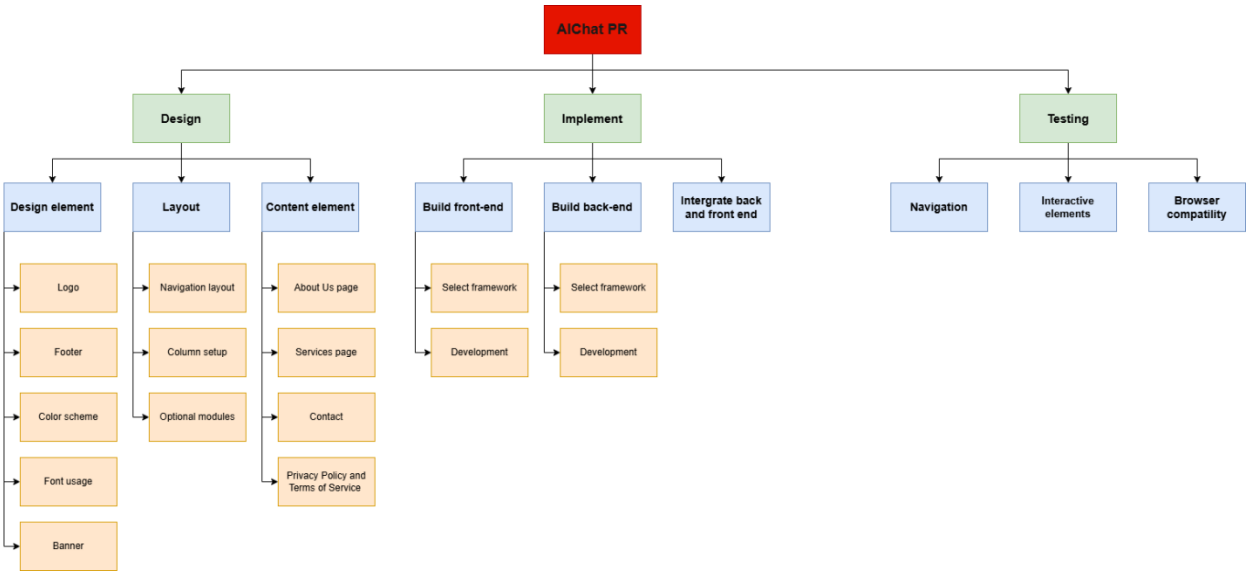
The project will take 12 weeks to carry out.

- **Week 4 (24/10 - 02/11):** Group registration, PA0 - Preparing project proposal and tools setup.
- **Week 5 (03/11 – 10/11):** PA1 - Writing project plan, vision document.
- **Week 6 (11/11 - 14/11):** PA1 - Writing project plan, vision document.
- **Week 7 (15/11 - 21/11):** PA2 - Revised project plan and detailed vision document, Use-case model and specification.
- **Week 8(22/11 - 28/11):** PA3 - Revising the use-case model and specifications from TA's feedback, Defining software architecture, class diagrams.
- **Week 9 (29/11 - 01/12):** PA4 - Revising software architecture that was submitted in PA3 and drawing a deployment diagram(s) using UML, Designing user-interface prototype.
- **Week 10 - 12 (02/12 - 19/12):** PA5 - Preparing test plan, designing and executing test cases, summarizing and reporting test results and submitting project presentation.

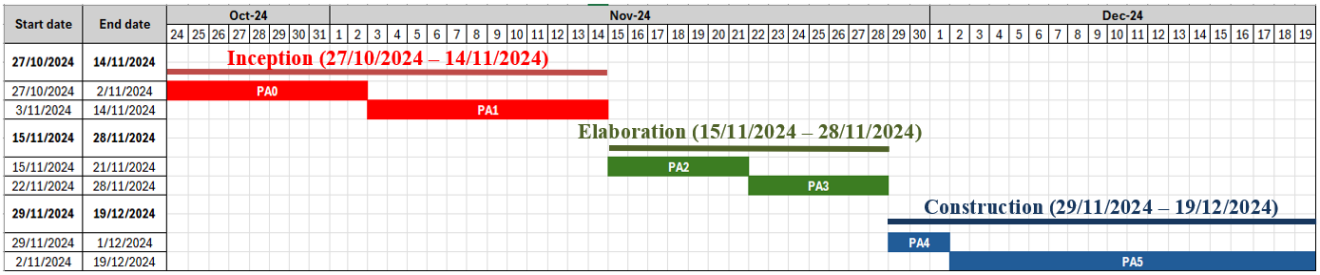
4.2 Project Plan

4.2.1 Phase and Iteration Plan

– Work Breakdown Structure:



– The allocation of time to the project phases:



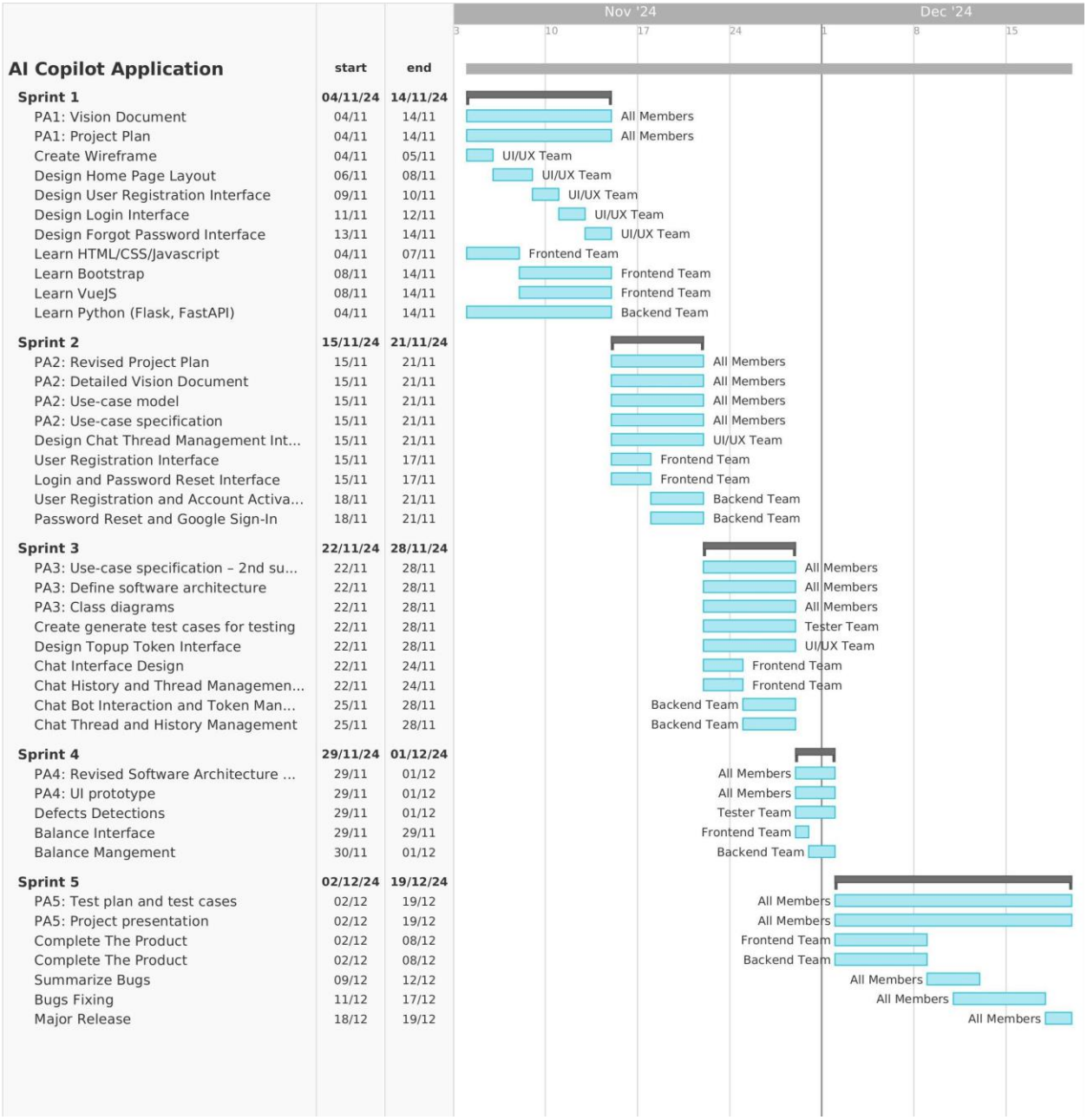
– Major milestones: Milestones are established at the end of each phase to define the particular goals to be met as well as the project's progress.

Milestone (At the end of the phase)	Achievement criteria
Inception	- Reach an agreement on project concepts. - Outline the fundamental product requirements and implementation plan.
Elaboration	- Complete the analysis and design of use-case diagrams.
Construction	- Implement the product successfully with all proposed features.

4.2.2 Releases

There will be two releases. The first, a beta version, will be completed at the end of sprint 4. Any issues detected will be fixed so that the full release may be completed before the conclusion of sprint 5.

4.2.3 Project Schedule



4.3 Project Monitoring and Control

4.3.1 Reporting

- **Weekly Meetings**: We hold meetings via Google Meets to discuss what we have done and updates, challenges, and goals for the coming week. These sessions will keep everyone aligned on project progress and priorities.
- **Weekly Status Reports**: Share a structured report summarizing key accomplishments, ongoing tasks, and any blockers encountered. This document provides a clear snapshot of project health and next steps.

- **Informal Chats**: Encourage casual check-ins between team members to address immediate issues, provide quick feedback, or share insights. This helps to maintain open communication and resolve small issues promptly.

4.3.2 Risk Management

Risk ID	Risk Description	Probability	Impact	Risk Exposure (Probability * Impact)	Priority	Mitigation Strategy or Contingency Plan
1	Team member leaving mid-project, resulting in slowed development and knowledge gaps	High	High	High	High	- Prepare a backup plan for coverage - Cross-train team members to share knowledge - Consider temporary hiring or contracting to bridge gaps
2	Design modifications based on the instructor feedback lead to increased workload and delays	Medium	High	Medium	High	- Schedule regular feedback sessions to capture changes early - Prioritize design adjustments based on core functionality
3	Software tools incompatibility, causing integration issues and potential feature adjustments	Medium	Medium	Medium	Medium	- Prepare alternate software tool options - Isolate dependencies to minimize integration impact
4	System performance lower than expected, impacting response times during peak usage	Medium	High	Medium	High	- Test system load early and continuously monitor performance - Consider scaling resources based on peak demand
5	Inaccurate time estimates for various development tasks	Medium	High	Medium	Medium	- Track progress regularly to adjust timelines - Implement a flexible project schedule to accommodate minor delays

6	Delays in bug fixing resulting in extended development times and potential missed deadlines	High	Medium	High	High	- Allocate additional time for bug fixing - Use debugging tools and code review processes to catch issues early
7	Underestimated system requirements leading to an incomplete or scaled-down feature set	Medium	Medium	Medium	Medium	- Identify core features that can be prioritized if time runs short - Plan an iterative approach for releasing additional features in later phases
8	Possible inefficiency in code generated by automated tools	Low	Low	Low	Low	- Test performance regularly - Optimize code manually if inefficiencies impact core functions
9	Project budget constraints, potentially limiting access to advanced tools or team expansion	Medium	Medium	Medium	Medium	- Optimize existing resources - Prioritize critical features and allocate resources accordingly
10	Dependence on third-party AI APIs may lead to service disruptions or integration issues	Medium	Medium	Medium	Medium	- Ensure API redundancy and consider fallback systems - Regularly monitor API updates and integration stability
11	Potential delays in finalizing premium feature set, impacting product launch timing	Medium	High	Medium	High	- Regularly track premium feature development progress - Launch with a phased approach, rolling out premium features gradually if needed
12	Requirement misalignment, where features do not meet actual user expectations	Medium	High	Medium	High	- Conduct frequent user feedback sessions - Create early-stage prototypes for user testing

4.3.3 Configuration Management

To ensure smooth collaboration and accessibility, select reliable tools for storing, sharing, and managing project files and code.

- **Google Drive** for organizing and sharing PA documents, project assets, and other non-code files.
- **Github** for efficient version control, tracking changes, and managing source code and related files across team members.

These tools will streamline the process of file management and version control, enabling better collaboration and real-time updates.