5CLCsCan

Gogle
Software Development Plan
Version 2.1

Revision History

Date	Version	Description	Author
08/6/2024	1.0	Report the project planning	Mẫn, Duy, Kiệt, Anh
06/7/2024	2.0	Update the project planning	Cao Huu Khuong Duy
07/7/2024	2.1	Review and modifying	Vo Nguyen Phuong Quynh

Table of Contents

Software Development Plan

1. Introduction

This Software Development Plan outlines the roadmap for the Gogle software project, detailing three key sections: the Project Overview, Project Organization, and Project Plan.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

Gogle aims to provide a web platform that helps users create trip plans easily, offering personalized recommendations for destinations, accommodations, activities, and dining options. The objectives are to develop a user-friendly interface, implement a recommendation system for personalized travel suggestions, provide accurate travel information, enable trips management, and create a community through user reviews and ratings.

2.2 Assumptions and Constraints

- Project has to finish in 12 weeks.
- The development team is fixed with 5 people, there will not be any more or less during the project.
- The final application only supports English.
- The team catches up about 3 times a week to review the project progress.
- Submit the Project Assignment to lecturers
- Complete weekly report for revision
- The project does not require any additional cost.
- Only one graphic designer for designing required screens
- User feedback is limited to our own team members, and some friends.
- Scalable design to support future growth plans

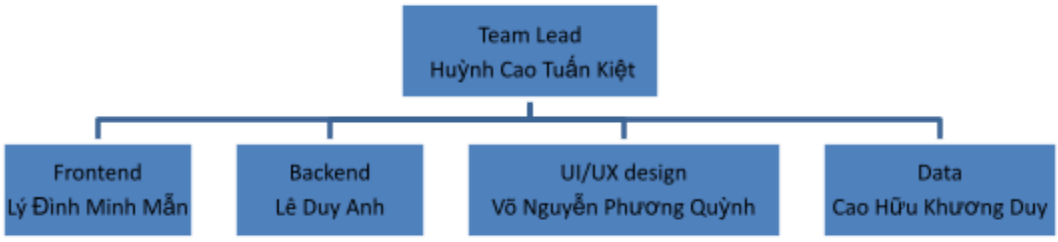
2.3 Project Deliverables

- User Interface Design - Wireframes and mockups of the website interface. (Target Delivery Date: Week 2)
- Backend Architecture - Design and development of the backend system, including database setup. (Target Delivery Date: Week 2)
- Recommendation System - Development and integration of the recommendation algorithms. (Target Delivery Date: Week 8)
- Front-end Development - Implementation of the user interface and integration with the backend. (Target Delivery Date: Week 9)
- Back-end Development – Design and development of the backend system, including database setup, API development, and integration with the frontend. (Target Delivery Date: Week 9)
- Testing Plan - A comprehensive testing plan, including unit tests, integration tests, and user acceptance tests. (Target Delivery Date: Week 10)
- Beta Version Release - Initial release for beta testing with selected users. (Target Delivery Date: Week 10)
- User Feedback Analysis - Collection and analysis of beta user feedback. (Target Delivery Date: Week 11)

- Final Version Release - Final release of the platform incorporating feedback from beta testing. (Target Delivery Date: Week 12)

3. Project Organization

3.1 Organizational Structure



3.2 Roles and Responsibilities

Person	Responsibilities
Huỳnh Cao Tuấn Kiệt	Role: Team leader, Backend developer. - API development - Implement the recommendation system - Manage the project progress
Lê Duy Anh	Role: Backend developer - API development - Performance Optimization
Võ Nguyễn Phương Quỳnh	Role: UI/UX designer - Design UI/UX - Support Frontend development
Lý Đình Minh Mẫn	Role: Frontend developer - Implementation of the user interface. - Integrate with the server side
Cao Hữu Khương Duy	Role: Data engineering - Collecting data - Setup database - Implementation search engine

4. Management Process

4.1 Project Estimates

Project Planning and Requirements Gathering (1 week):

- Define the project scope, objectives, and requirements.
- Gather user preferences and criteria for recommending places.

Design and Architecture (1-2 weeks):

- Design the overall system architecture.
- Define database schema (MongoDB) and data models.
- Plan the integration with the search engine for efficient place recommendations.

Frontend Development (3-4 weeks):

- Develop the frontend using Next.js.
- Implement user interface components for capturing preferences and displaying recommendations.
- Ensure responsiveness and usability across different devices.

Backend Development (3-4 weeks):

- Set up MongoDB database.
- Implement backend logic for processing user preferences and querying the database.
- Integrate with the search engine for efficient and relevant place recommendations.

Testing and Quality Assurance (1-2 weeks):

- Conduct unit testing for both frontend and backend components.
- Perform integration testing to ensure the system works as expected.
- Validate recommendations against different user scenarios and preferences.

Deployment and Optimization (1 week):

- Deploy the application on a suitable hosting platform.
- Optimize frontend and backend performance.
- Set up monitoring and error logging for ongoing maintenance.

4.2 Project Plan

4.2.1 Phase and Iteration Plan

Our project has three phases: Inception, Elaboration, and Construction.

Phase 1: Inception

Objective: Establish project scope, requirements, and initial design.

- **Iteration 1: Requirements Gathering and Analysis**
  - **Start Date:** May 15, 2024
  - **End Date:** May 28, 2024
  - **Objectives:**
    - Identify project stakeholders.
    - Gather and document requirements.
    - Define project scope.
    - Create high-level use cases.
    - Develop initial project plan and schedule.
    - **Milestone**: Approval of requirements and project scope.

Phase 2: Elaboration

Objective: Detailed system design, architecture, and initial development.

- **Iteration 2: System Architecture and Design**
  - **Start Date:** May 29, 2024
  - **End Date:** June 18, 2024
  - **Objectives:**
    - Develop system architecture.
    - Create detailed design documents.
    - Define database schema.
    - Create wireframes and prototypes.
    - Conduct feasibility analysis for recommendation system.
    - **Milestone**: Approval of system design and architecture.

Phase 3: Construction

Objective: Full system development, testing, and deployment.

- **Iteration 3: Initial Development and Integration**
  - **Start Date:** June 19, 2024
  - **End Date:** July 16, 2024
  - **Objectives:**
    - Set up a development environment.
    - Develop core functionalities (user authentication, basic trip planning).
    - Integrate AI module for place recommendations.
    - Develop and test API endpoints.
    - Conduct initial testing and debugging.
    - **Milestone**: Completion of initial development.
- **Iteration 4: Full Development, Testing, and Final Adjustments**
  - **Start Date:** July 17, 2024
  - **End Date:** August 5, 2024
  - **Objectives:**
    - Complete development of all core features (user interface, advanced trip planning, recommendation system).
    - Perform unit testing, integration testing, and system testing.
    - Conduct user acceptance testing (UAT).
    - Fix bugs and optimize performance.
    - Prepare documentation and user guides.
    - Prepare deployment environment.
    - **Milestone**: Final release and project completion.

Work Breakdown Structure (WBS)

1. **Project Initialization**
   - Stakeholder Identification
   - Requirements Gathering
   - Scope Definition
   - Project Plan Development
2. **System Design**
   - System Architecture Development
   - Detailed Design Documentation
   - Database Schema Design
   - Wireframe and Prototype Creation
3. **Initial Development**
   - Development Environment Setup

- Core Functionality Development
- API Development
- Initial Testing and Debugging

4. **Full Development and Testing**
    - Feature Completion
    - Unit Testing
    - Integration Testing
    - System Testing
    - User Acceptance Testing (UAT)
    - Bug Fixing and Performance Optimization
    - Documentation and User Guide Preparation
    - Deployment Preparation

Major Milestones

1. **Requirements and Scope Approval** (May 28, 2024)
2. **System Design Approval** (June 18, 2024)
3. **Initial Development Completion** (July 16, 2024)
4. **Final Release and Project Completion** (August 5, 2024)

Release Points and Demos

- **Initial Prototype Demo:** End of Iteration 2 (June 18, 2024)
- **Beta Release:** End of Iteration 3 (July 16, 2024)
- **Final Release and Presentation:** End of Iteration 4 (August 5, 2024)

4.2.2 Releases

Version 1.0: Demo release

- Description: The purpose of this demo is to demonstrate the initial progress, an overview of the user interface and gather feedback from stakeholders.
- Release date: June 18, 2024

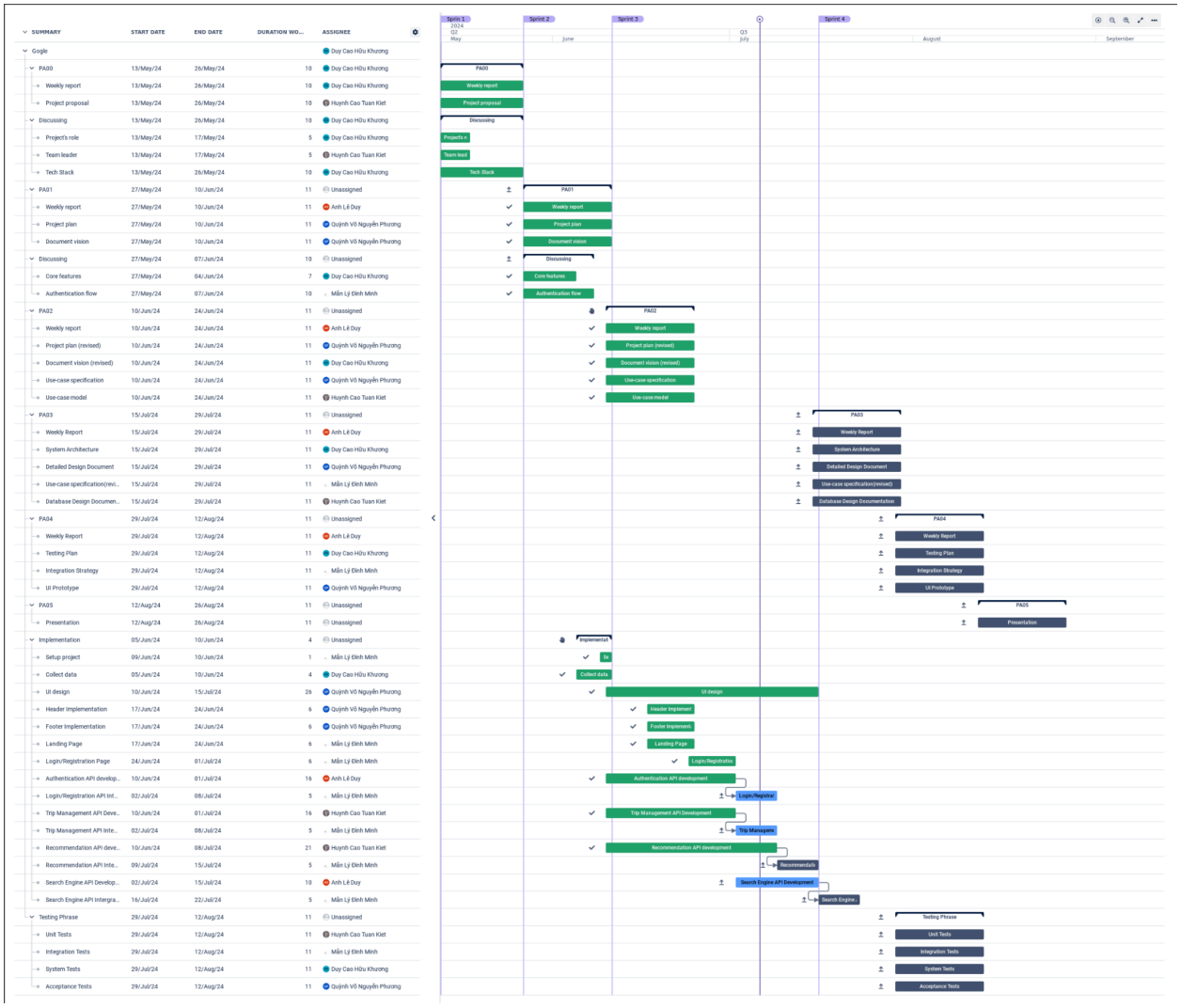
Version 2.0: Beta release:

- Description: This release will feature a fully functional user interface, user login, registration, core backend services and the initial version of recommendation system.
- Release date: July 16, 2024

Version 3.0: Final release

- Description: The final release will include all planned features, enhanced version of recommendation system, complete scheduled management and community features like user review and ratings. This version will be fully tested and ready for release.
- Target Date: August 5, 2024.

4.2.3 Project Schedule



4.3 Project Monitoring and Control

4.3.1 Reporting

Weekly meeting

We have at least 1 meeting a week to report the task's progress.

- Week 1:
  - Date: 16/5
    - Method: offline
    - Content: Discuss work processes and select a team leader.
- Week 2:
  - Date: 20/5
    - Method: offline
    - Content: Brainstorm the project idea, discussing the core features.
- Week 3:
  - Date: 27/5
    - Method: offline

        - Content: Set up project repository, database, choose the tech stack.
    - Date: 30/5
        - Method: offline
        - Content: Divide the project's roles, assign the work to team members.
    - Date: 31/5
        - Method: online
        - Content: Discuss the recommendation system, review the tasks.
- Week 4
    - Date: 2/6
        - Method: online
        - Content: Assign people to do PA01.
    - Date: 4/6
        - Method: offline
        - Content: Review work progress.
    - Date: 7/6
        - Method: online
        - Content: Provide feedback on the authentication flow, deploy the Alpha version.
- Week 5
    - Date: 11/6
        - Method: online
        - Content: Define the database schema, research the search engine algorithm.
    - Date: 14/6
        - Method: online
        - Content: Add Swagger into the project, adjust some designed screens.
    - Date: 16/6
        - Method: online
        - Content: Review work progress.
- Week 6
    - Date: 19/6
        - Method: online
        - Content: Assign people to do PA02.
- Week 7
    - Date: 27/6
        - Method: online
        - Content: Discuss the search engine problem, provide feedback on PA02.
- Week 8
    - Date: 1/7
        - Method: online
        - Content: Upgrade the recommendation system to handle users' budgets and deploy the search engine.

Weekly status report

	Progress	Issues/Risks	Next Steps
Week 1	Research the tools which support our work processes.	None at this state.	Standardize work tools and start brainstorming ideas.
Week 2	Generated a list of potential project ideas and discussed key features.	None at this stage.	Finalize the project idea and start initial planning.
Week 3	Initialized the project tools, assigned member's roles.	Learning curve with new tech stack	Begin development, continue tasks.
Week 4	Begin development and provide feedback on member's tasks.	Minor issues identified with authentication flow.	Resolve the authentication problem, continue development.
Week 5	Resolve some technical problems.	The code should be refactored to incorporate the latest changes.	Plan to start PA02.
Week 6	Work on PA02.	Potential delays in PA02 assignments due to team availability.	Complete PA02 and continue with the implementation.
Week 7	Final check PA02, implement search engine functionality.	Search engine issues should be addressed carefully to ensure effective results.	Ensure effective performance of the recommendation system.
Week 8	Discuss how to upgrade the recommendation system.	Ensure effective performance of the recommendation system.	Complete the recommendation system.

4.3.2 Risk Management

Risk ID	Risk Description	Probability	Impact	Priority	Mitigation Strategy or Contingency Plan
Technical Integration Issues	Difficulties in integrating various technologies and components (e.g., Next.js, MongoDB, and the search engine) may cause	Medium	Project delays, increased costs, reduced functionality	High	Conduct thorough testing during integration phases. Plan for incremental integration to identify and resolve issues early. Maintain open communication channels

	delays or functional issues				among team members for quick resolution
Data Security Breach	Unauthorized access to or leakage of sensitive user data, leading to privacy violations and potential legal consequences	Low	Legal consequences, reputational damage, financial loss	Medium	Implement strong encryption methods for sensitive data. Regularly update security protocols and conduct vulnerability assessments. Educate team members about data security best practices.
Performance Bottlenecks	The application may face performance issues, such as slow response times or crashes, under high user load.	High	User frustration, negative reviews, resource intensive	High	Design and implement scalable architecture. Conduct performance testing under load conditions. Optimize code and database queries for efficiency. Monitor system performance post-deployment and implement necessary optimizations.
Resource Constraints	Insufficient budget, time, or team resources to complete the project, potentially causing delays or scope reduction.	Medium	Project delays, scope reduction, quality compromise	High	Develop a detailed resource plan at the project outset. Allocate resources efficiently and prioritize critical tasks. Consider phased development or MVP (Minimum Viable Product) approach to deliver essential functionalities within constraints.
Team Member Availability	Team members may have conflicting commitments or become unavailable, affecting project progress.	Low	Project delays, knowledge gaps, increased workload	Low	Maintain transparent communication regarding individual schedules and commitments. Plan for contingencies by cross-training team members on critical tasks. Implement collaborative tools and regular progress updates to ensure continuity.

4.3.3 Configuration Management

- **Document Storage and Sharing:** Use Google Drive, Google Docs to store and share all project documents and files. Ensure these files are well-organized and access-controlled.
- **Process Management:** Utilize Trello to manage project workflows and tasks. Integrate Trello with Slack for real-time updates and notifications.
- **Source Code Management:** Store and manage source code on GitHub. Implement best practices for branching, code reviews, and CI/CD.
- **Communication:** Use Slack as the central hub for team communication. Integrate with other tools to centralize updates and notifications.