<Company Name>

Pro Chicken Fitness
Software Development Plan (Small Project)
Version <1.0>

Revision History

Date	Version	Description	Author
28/10/20223	1.0	Build-up the idea	ProChicken team

Table of Contents

Software Development Plan (Small Project)

1. Introduction

1.1 Purpose

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.

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 Pro Chicken Fitness 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 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

This project aims to develop a product with nutritional care tips and provide exercise instructors, built on two platforms: mobile application and web.

The goal of the project is to perform at least basic functions, develop and complete products on schedule, ensuring compliance with the needs and tastes of the current market.

2.2 Assumptions and Constraints

Budget: estimated $1000 Human resources: 5 students

Equipment:

- Personal computer (for projects), projector (for presentations)
- Jira (for project management), Git & GitHub (for saving code), MS Word (for writing documents), MS PowerPoint (for presentations)

Schedule: estimated 2 months

2.3 Project Deliverables

Plan and assign

Product descriptions and instructions

Modules for finished products and products

3. Project Organization

3.1 Organizational Structure

1. Project Owner/Manager:

- Responsible for the overall management of the project.
- Defines project objectives, sets goals, and ensures that the project aligns with the organization's strategic vision.
- Manages the budget, timeline, and resources.

2. Development Team:

- **Front-End Developers (Web and Mobile)**: Responsible for developing the user interfaces and user experience for both the web and mobile versions of the application. They work on implementing the design created by UI/UX designers.
- **Back-End Developers**: Build the server-side components, including the API, database, and application logic. They are responsible for the core functionality of the application, including user authentication and data management. Manage the database where user data, meal plans, workout routines, and other application data are stored.
- **Testers**: Responsible for testing the application to identify and report bugs and issues.

3. UI/UX Team:

- **UI/UX Designers**: Collaborate with the development team to create visually appealing and user-friendly interfaces for the application. They also focus on user experience design to ensure the app is easy to navigate and engaging.

3.2 Roles and Responsibilities

1. Đạt, Lộc: Research and Design

- **Role**: Researcher and UI/UX Designer
- **Responsibilities**:
  - Conduct research to understand the features and components typically found in fitness applications.
  - Create user interface (UI) and user experience (UX) designs using tools like Figma.
  - Develop wireframes, prototypes, and mockups for the fitness application based on research findings.
  - Collaborate with other team members to ensure the design aligns with the project's objectives.

2. Hoà, Đức: Frontend Development

- **Role**: Frontend Developers (React, React Native)
- **Responsibilities**:
  - Learn and become proficient in frontend technologies such as React and React Native.
  - Convert the UI/UX designs created by Đạt, Lộc into functional web and mobile user interfaces.
  - Implement the front-end logic, user interactions, and navigation based on the design specifications.
  - Ensure the user interface is responsive and provides an engaging user experience.

3. Đức Anh: Backend Development

- **Role**: Backend Developer (Spring Boot)
- **Responsibilities**:
  - Utilize Spring Boot as the backend technology to develop the API for the fitness application.
  - Create and manage the server, databases, and application logic.
  - Implement user authentication, data storage, and server-side functionality.
  - Ensure the backend system is scalable, secure, and efficient.

4. Management Process

4.1 Project Estimates

Labor Costs:

- Development Team (Front-End, Back-End, UI/UX, QA): Salaries, hourly rates, or contract fees.

Technology and Infrastructure Costs:

- Software development tools and licenses.
- Cloud hosting and server costs.
- Third-party API costs (e.g., nutritional databases, payment gateways).
- UI/UX design tools and software.

Testing and Quality Assurance Costs:

- Testing tools and software.
- QA tester salaries or fees.

4.2 Project Plan

4.2.1 Phase Plan

Phase 1: Project Initiation

WBS:

- Define Project Scope
- Form Project Team
- Establish Project Objectives
- Secure Budget and Resources

Timeline:

- Start Date:
- Target Completion Date:

Major Milestone:

- Project Scope Defined
- Team Assembled
- Project Objectives Established
- Budget Secured

Phase 2: Requirements and Design

WBS:

- User Requirements Gathering
- UI/UX Design
- Database Design
- Feature Prioritization

Timeline:

- Start Date: [Date]
- Target Completion Date: [Date]

Major Milestone:

- User Requirements Collected
- UI/UX Design Completed
- Database Design Finalized
- Feature Prioritization

Phase 3: Development

WBS:

- Front-End Development (Web and Mobile)
- Back-End Development
- User Authentication and Registration
- User Profile Creation

Timeline:

- Start Date: [Date]
- Target Completion Date: [Date]

Major Milestone:

- Front-End Development Complete
- Back-End Development Completed
- User Authentication Implemented
- User Profile Creation

Phase 4: Testing and Quality Assurance

WBS:

- Unit Testing
- Integration Testing
- User Testing
- Bug Fixing

Timeline:

- Start Date: [Date]
- Target Completion Date: [Date]

Major Milestone:

- Unit Testing Phase
- Integration Testing Phase
- User Testing and Feedback
- Bug Fixing Completed

Phase 5: Deployment

WBS:

- Staging Environment Setup
- Production Deployment

Timeline:

- Start Date: [Date]
- Target Completion Date: [Date]

Major Milestone:

- Staging Environment Ready
- Production Deployment

4.2.2 Iteration Objectives

Iteration 1: Requirements Gathering and Planning

- Define the project scope, goals, and constraints.
- Identify user stories and feature requirements.
- Create the project plan, including the schedule and budget.
- Assemble the project team and assign roles and responsibilities.

- Develop a comprehensive list of required features and prioritize them.

Iteration 2: User Authentication and Authorization

- Implement user registration and login functionality.
- Develop the user profile creation process.
- Ensure that user data is securely stored and encrypted.
- Perform initial testing to verify the effectiveness of authentication and registration.

Iteration 3: User Interface Design

- Collaborate with UI/UX designers to create wireframes and mockups.
- Design the visual elements of the application, including color schemes and layouts.
- Establish a consistent design language for both web and mobile interfaces.
- Gather user feedback on the design and make necessary adjustments.

Iteration 4: Front-End Development

- Implement the user interface based on the approved design.
- Develop responsive web interfaces for different devices and screen sizes.
- Create mobile app interfaces using React Native for iOS and Android.
- Begin integration with the back-end for user registration and authentication.

Iteration 5: Back-End Development - User Profiles

- Develop the back-end system to manage user profiles and preferences.
- Implement the database structure to store user data securely.
- Establish user profile customization features, such as favorite ingredients.
- Ensure efficient API endpoints for retrieving and updating user profiles.

Iteration 6: Meal Planning Feature

- Design the meal planning feature, including user preferences.
- Create an algorithm for suggesting personalized meal plans.
- Integrate nutritional databases or APIs for ingredient information.
- Implement the feature to allow coaches to request ingredients for users.
- Begin initial testing of the meal planning system.

Iteration 7: Workout Planning Feature

- Design the workout planning feature, including user fitness goals and coach recommendations.
- Develop the system for generating workout plans based on user profiles.
- Enable coaches to create customized workout plans for their clients.
- Ensure seamless synchronization between meal and workout planning features.
- Initiate testing for workout plan generation and coach-client interaction.

Iteration 8: Testing and Quality Assurance

- Perform thorough testing of the entire application, focusing on user registration, meal planning, workout planning, and data security.
- Identify and address any bugs, issues, or performance bottlenecks.
- Conduct user testing to gather feedback and make necessary adjustments.

Iteration 9: Deployment

- Prepare the application for deployment to staging and production environments.
- Deploy the application to production servers.
- Ensure a smooth transition to the live environment.

4.2.3 Releases

4.2.4 Project Schedule

Phase/Milestones	Start Date	Demo Date	Completion Date
Requirements Gathering			
Authentication and Authorization			
User interface			
Frontend components			
Meal Planning Feature			
Workout Planning Feature			
Testing			
Deployment			

4.2.5 Project Resourcing

1. Project Team Members

Project Manager: 1

- **Skills/Experience**: Project management, leadership, and communication skills.

Front-End Developers (Web and Mobile): 2

- **Skills/Experience**: Proficiency in HTML, CSS, JavaScript, React, React Native.
- **Training**: Stay updated with the latest front-end development trends and tools.

Back-End Developers: 1

- **Skills/Experience**: Proficiency in server-side technologies, e.g., Spring Boot, database management.
- **Training**: Continuous learning about back-end development best practices.

UI/UX Designers: 1

- **Skills/Experience**: Strong design skills, experience in UI/UX design tools like Figma.
- **Training**: Stay updated with UI/UX design principles and trends.

QA Testers: 2

- **Skills/Experience**: Experience in software testing, bug tracking, and quality assurance processes.
- **Training**: Continuous training in software testing methodologies.

2. Approach for Finding and Acquiring Staff

- **Internal Team**: Recruit internal team members from your organization with the required skills, such as developers, designers, and project managers.

- **External Hires**: If internal resources are insufficient, consider hiring external talent through job postings, recruitment agencies, or freelancers for specialized roles.

3. Special training project

4.3 Project Monitoring and Control

4.3.1 Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

- Use a centralized requirements management system to document, track, and manage product requirements.
- Implement a change control board (CCB) to evaluate and approve changes to requirements.
- Clearly document requirements, including functional, non-functional, and design requirements.

4.3.2 Reporting and Measurement

Updated cost and schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics, will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

In addition, overall costs will be monitored against the project budget.

4.3.3 Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Risk Ranking (High, Medium, Low)	**Risk Description and Impact**	**Mitigation Strategy and/or Contingency Plan**
Medium	Key team members may leave the project or have gaps in required skills.	Cross-train team members to reduce knowledge gap. Document critical knowledge and processes. Maintain clear communication with the team.
High	Potential integration challenges with third-party systems. Lead to project delays, increased development costs.	Conduct a thorough assessment of third-party integrations before implementation
High	Scalability issues due to rapid user growth. Lead to negative user experience	Implement load testing to evaluate scalability

	and users leaving the app.	
Medium	Resource constraints, including budget limitations or difficulties in securing necessary personnel. Lead to delays in development, reduced feature scope, or increased project costs.	Carefully manage the budget, explore alternative resource options, and prioritize critical features.
High	Potential data breaches, unauthorized access, or other security vulnerabilities. Lead to loss of user trust, legal consequences, and compromised user data.	Implement strong encryption, regular security audits, and user data protection measures.
Low	Uncontrolled expansion of project scope as new requirements or features are added. Lead to project delay	Implement agile method to continually evaluate the progress of project

4.3.4 Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

Process:

- Implement a change request process for submitting and reviewing problems and changes.
- Use version control systems for managing project artifacts, source code, and documentation.

Artifact Naming, Marking, and Numbering:

- Establish naming conventions for project artifacts to ensure consistency and traceability.
- Mark and number hardware, software, and documentation artifacts clearly.

Retention Policies:

- Define retention policies for different types of project artifacts, considering legal and regulatory requirements.

Back-up, Disaster, and Recovery Plans:

- Implement regular backup procedures for critical project data and artifacts.
- Develop disaster recovery plans to ensure the project's continuity in the event of unforeseen issues.

Media Retention:

- Specify where and how project media is retained, including online and offline storage, media type, and format.