Shelf-Smart

Software Development Plan

Version <1.0>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| <21/02/2024> | <1.0> | Introduction - Project Overview | Ansuman Sharma,  Yadhunath Tharakeswaran,  Shayaan Mohammed,  Kundana Dongala |
| <24/02/2024> | <1.0> | Project Organization | Ansuman Sharma,  Yadhunath Tharakeswaran,  Shayaan Mohammed,  Kundana Dongala |
| <25/02/2024> | <1.0> | Management Process - Annexes | Ansuman Sharma,  Yadhunath Tharakeswaran,  Shayaan Mohammed,  Kundana Dongala |
|  |  |  |  |

Table of Contents

[**1.**](#_heading=h.30j0zll) **Introduction 4**

[*1.1*](#_heading=h.1fob9te) *Purpose 4*

[*1.2*](#_heading=h.3znysh7) *Scope 4*

[*1.3*](#_heading=h.2et92p0) *Definitions, Acronyms, and Abbreviations 4*

[*1.4*](#_heading=h.tyjcwt) *References 4*

[*1.5*](#_heading=h.3dy6vkm) *Overview 5*

[**2.**](#_heading=h.1t3h5sf) **Project Overview 5**

[*2.1*](#_heading=h.4d34og8) *Project Purpose, Scope, and Objectives 5*

[*2.2*](#_heading=h.2s8eyo1) *Assumptions and Constraints 5*

[*2.3*](#_heading=h.17dp8vu) *Project Deliverables 5*

[*2.4*](#_heading=h.3rdcrjn) *Evolution of the Software Development Plan 5*

[**3.**](#_heading=h.26in1rg) **Project Organization 5**

[*3.1*](#_heading=h.lnxbz9) *Organizational Structure 5*

[*3.2*](#_heading=h.35nkun2) *External Interfaces 6*

[*3.3*](#_heading=h.1ksv4uv) *Roles and Responsibilities 6*

[**4.**](#_heading=h.44sinio) **Management Process 6**

[*4.1*](#_heading=h.2jxsxqh) *Project Estimates 6*

[*4.2*](#_heading=h.z337ya) *Project Plan 6*

[*4.3*](#_heading=h.3whwml4) *Project Monitoring and Control 7*

[*4.4*](#_heading=h.qsh70q) *Requirements Management 7*

[*4.5*](#_heading=h.49x2ik5) *Quality Control 7*

[*4.6*](#_heading=h.147n2zr) *Reporting and Measurement 7*

[*4.7*](#_heading=h.ihv636) *Risk Management 8*

[*4.8*](#_heading=h.1hmsyys) *Configuration Management 8*

[**5.**](#_heading=h.2grqrue) **Annexes 8**

Software Development Plan

# Introduction

This document contains details/specifications of the process adopted by our team to make a web application called Shelf-Smart which lets the user add and keep track of their pantry supplies and also recommends the user recipes which can be prepared using available pantry ingredients. This document also outlines a plan which is efficient, low-risk, and provides equal opportunities/responsibilities to each team member in the development of our app.

## Purpose

This document will delve into the scope, definitions, acronyms, and abbreviations pertinent to Shelf-Smart. It will also provide detailed insights into the logical view, describing packages, modules, and subsystems, along with an interface description. quality aspects, such as how the architecture contributes to system capabilities, will be addressed, highlighting the significance of optimized performance, quality attributes, and risk management in achieving the project’s goals. Shelf-Smart’s architecture will be a key driver in ensuring organizational standards, scalability, maintainability, and flexibility for users in managing their culinary resources effectively.

The following people use the *Software Development Plan*:

* The Team Lead and the Scrum Master use it to plan the project schedule and resource needs, and to track progress against the schedule.
* All 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.

## Scope

This *Software Development Plan* describes the overall plan to be used by Shelf-Smart, 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*.

## Definitions, Acronyms, and Abbreviations

* Ingredient: A commodity which can be added to the pantry by the user.
* Spice (child class of Ingredient): Resembles an ingredient except that it doesn’t require an expiry date
* Recipe: Only includes the name of the dish and 1 or 2 line description.
* User: Any one who uses the app to keep track of their pantry.

See the Project Glossary.

## References

* Iteration Plans: Plan to implement pantry tracking, plan to implement recipe recommendations
* Vision: to develop an innovative and user-friendly platform revolutionizing meal planning and pantry management in C++
* Glossary
* Development Case

## 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.

Management Process  — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines — provide an overview of the software development process, including methods, tools and techniques to be followed.

# Project Overview

## Project Purpose, Scope, and Objectives

## The purpose of the Shelf-Smart is to develop an innovative and user-friendly platform that revolutionized meal planning and pantry management using C++. The primary objective is to design a system that takes input on food items, quantities, and expiry dates, ultimately generating personalized dish recommendations based on the available pantry ingredients. The project will integrate essential features such as inventory display, ingredient classification, and the ability to recommend dishes with user input.

As a part of the software Development Process, Shelf-Smart will adhere to a well-structured plan, including the creation of a detailed project plan, a comprehensive requirements document, a design document aligned with the specified requirements, a set of test cases, and the delivery of a fully developed product. Each iteration will be conducted timely, ensuring alignment with functional requirements defined for the project.

## Assumptions and Constraints

The website will be initially developed using C++, with the front-end crafted using ReactJS and the back-end built using CppCMS. Our team, consisting of four members, will evenly distribute tasks to collaborate on both the front-end and back-end development. As we are relatively new to C++, we plan to frequently consult external sources to augment our understanding of the language and improve our website development skills. We have scheduled bi-weekly team meetings, taking into consideration the availability of teammates, to discuss progress, address challenges, and allocate tasks for the upcoming week. With this collaborative approach, we are confident that by the end of the semester, we will successfully create a basic website with fully functional features.

## Project Deliverables

* Requirements document: Comprehensive document outlining the functional and non-functional requirements of shelf-smart.
* Design Specifications: Detailed specifications defining the architecture, modules, and interfaces of shelf smart
* Test Cases: A set of test cases ensuring thorough testing of shelf-Smart’s functionalities to meet quality standards
* Code: The actual implementation of shelf-smart, including both front end and back end (C++ with CppCMS)

## Evolution of the Software Development Plan

* Version 1.0: Create a basic draft of the website ideas, and document various resources to work over it.
* Version 2.0: Complete code of the functioning website in the backend side.
* Version 3.0: Complete front-end development of the website.

# Project Organization

## Organizational Structure:

Our dynamic team comprises four highly skilled members, each dedicated to their specialized roles, ensuring a well-rounded and efficient collaboration. The Project Lead orchestrates the overall vision. The Assistant Project Lead provides invaluable support and coordination. The Backend Developer deals technical insights, and ensures the robustness of our project. The Scrum Master navigates the team through agile methodologies, and fosters a streamlined workflow.

To maintain transparency and synergy, our team engages in bi-weekly meetings. These sessions serve as a platform to delve into the current project status, address any existing challenges, exchange diverse perspectives, and closely monitor individual progress. These sessions, focused on testing and validation, not only ensure the quality of our deliverables but also provide an opportunity for collective learning and enhancement.

.

## External Interfaces

In order for the ShelfSmart initiative to be implemented and accepted by users, communication with other organizations is essential. This is how the project communicates with external entities:

* *End Users:* 
  + Internal contact: Ansuman Sharma(Team lead) will work towards getting input from users, comprehending their needs, and working with the development team to put the improvements into practice.
  + External contact: users who will be utilizing the ShelfSmart platform will offer their input on the general user experience, functionality, usability, and accordingly accept the finished product depending on the performance satisfaction.
* *Software Testing/QA Teams***:**
  + Internal contact: Shayaan Mohammed (Quality Assurance/Backend Dev) will make sure the program satisfies quality requirements, observe and record any errors or problems, and work with the development team to fix them.
  + External contact: Dedicated QA testers or testing teams test the product thoroughly, reporting problems and bugs, and confirm that the program satisfies requirements before it is released.
* *Deployment Team:*
  + Internal contact: Yadunath Tharakeswaran (Assistant Team Lead) looks after organizing the deployment procedure, working with the deployment team to guarantee a seamless software installation and configuration, and provide the required assistance during deployment.
  + External contact: IT or deployment team responsible for deploying software will put the deployment plan into action, setting up the program in the production environment, and making sure end users can utilize it.
* *Stakeholders/Management:*
  + Internal contact: Kundana Dongala( Scrum Master) Maintains the line of communication between project objectives and stakeholder expectations, soliciting feedback from stakeholders, and communicating project progress.
  + External contact: Project stakeholders, management, or sponsors provide suggestions for the project's course, examining project deliverables and milestones, and, in the end, accept the finished product provided it satisfies corporate requirements and objectives.

## Roles and Responsibilities

* **Project Lead/Technical Leader:** Ansuman Sharma
  + **Contact:** [ansuman@ku.edu](mailto:ansuman@ku.edu); 785-371-7647
  + **Responsibilities:**

1. Overall coordination and management of the project.
2. Organizing the agenda for each meeting.
3. Presiding over team meetings.
4. Project testing.

* **Assistant Project Lead:** Yadunath Tharakeswaran
  + **Contact:** [yadhu2003@ku.edu](mailto:yadhu2003@ku.edu); 913-375-3250
  + **Responsibilities:**

1. Assisting with organization and team planning.
2. Project testing.
3. Presiding over meetings in the absence of Project Lead.
4. Actively contribute to feature engineering.

* **Quality Checker/Backend Developer:** Shayaan Mohammed
  + **Contact:** [s362m330@ku.edu](mailto:s362m330@ku.edu); 913-548-5807
  + **Responsibilities:**

1. Keep track of risks and quality of code being used.
2. Decide the inclusion/exclusion of any part of code in the program.
3. Actively contribute to handling web page rendering in a browser.
4. Keep Project Lead informed of any required changes in code.

* **Scrum Master:** Kundana Dongala
  + **Contact:** [k628d416@ku.edu](mailto:k628d416@ku.edu); 913-762-2065
  + **Responsibilities:**

1. Maintain logs of team meetings and progress in plan.
2. Make required changes to deadlines/process flow as required.
3. Inform the team of required topics to be discussed.
4. Actively contribute to building a database feature for the project.

# Management Process

## Project Estimates

* Cost:
* The estimated cost for the Shelf-Smart project is currently set at Zero(nothing)
* Schedule:
  + The estimated schedule for the shelf-Smart project is based on the semester timeline and the planned iterations. The project is scheduled to be completed by th end of the semester, aligning with academic deadlines

## Project Plan

The project plan for Shelf-Smart spans the entire semester, from [Start Date] to [End Date], with bi-weekly iterations covering tasks in front-end and back-end development, testing, and documentation. Major milestones include the completion of the Requirements Document, Design Specifications, Test Cases, and subsequent code implementations leading to internal demos, beta releases, and final user acceptance testing. The team comprises a Project Manager, Scrum Master, Front-end Developer, Back-end Developer, QA Engineer, and Documentation Specialist, each with well-defined roles and responsibilities. The chosen technology stack involves ReactJS for front-end, C++ with CppCMS for back-end, and GitHub for version control. Iteration plans detail specific tasks, goals, and risk mitigation strategies, while regular team and stakeholder communication ensures ongoing progress updates and feedback sessions. Continuous documentation updates capture changes, challenges, and solutions, making the plan a comprehensive roadmap for the team's organized and efficient development process.

### Iteration Objectives

The iteration objectives for Shelf-Smart are designed to guide the structured development process over eight iterations. Beginning with the creation of a detailed Requirements Document and initial project setup, subsequent iterations focus on finalizing design specifications, implementing front-end and back-end components, and conducting comprehensive testing. Additional features are integrated based on user feedback, leading to a beta release for internal testing and subsequent adjustments. The final iterations involve thorough testing of all functionalities, user documentation finalization, and preparation for the official launch. Refer to the related Iteration Plan Documents for more detailed information on specific tasks and timelines.

### Releases

Shelf-Smart will undergo an internal demo release for team feedback, followed by a beta release for thorough internal testing and refinement. The final user release marks the official launch, providing users access to the fully developed Shelf-Smart platform for pantry tracking and recipe recommendations.

### Project Schedule

The project schedule includes key milestones:

* Requirements Documentation
* Design Specifications Finalization
* Test Case Development
* Initial Code Implementation
* Beta Release for Internal Testing
* Final Code Implementation
* Release Candidate for UAT
* Final Release

Each of them would be completed by corresponding target dates. These milestones represent major phases, ensuring a structured progression from initial planning to the official launch of Shelf-Smart.

### Project Resourcing

Based on the roles, the team's each member brings a specific set of skills and responsibilities essential for the success of the project.

Here's a breakdown of any special training they may require:

* Project Lead/Technical Leader: It could be helpful to receive training in software testing processes, C++ programming, and project management methodologies.
* Assistant Project Lead: Training in project management, team leadership, and C++ programming would help make their role as assistant project lead much stronger.
* Quality Checker/Backend Developer: Training in web development technologies, C++ programming, code review procedures, and quality assurance practices is required for the position
* Scrum Master: It could be helpful to receive training in database administration, C++ programming, Agile techniques, and Scrum methodology.

To guarantee that team members have the abilities and information required to carry out their responsibilities successfully, training should preferably be finished prior to the project's commencement or at the start of each project phase or iteration.

## Project Monitoring and Control

* Requirements Management:
  + We will adhere to the principles of artifact elicitation to comprehensively gather essential ingredients that a majority of users would utilize on a daily basis, which would be fruitful to include the different recipes to our website based on it. Additionally, we will delve into some reliable sources to study C++ and stack systems that would be beneficial towards the back-end design of Shelf-Smart.
* Quality Control:
  + Our team is committed to a rigorous code review process, meticulously examining each other's code before merging into the master repository on GitHub. A well-defined test code criteria will be established for every section of the code, ensuring the quality of all specified requirements. Overall, our team would be committed to a rigorous debugging process, implement robust exception handling, and real-life scenario tests before finalizing the outcome for each iteration and version of the code.
* Risk Management:
  + Each team member will monitor potential risks such as memory leaks, complex errors, and database corruption on a daily basis. We will establish systematic meetings to proactively discuss potential and actual risk factors, taking preventive measures promptly.
* Configuration Management:
  + GitHub will serve as our primary platform for version control, and we will integrate aspects of the Scrum Agile methodology that would make sure that development will progress continuously, with testing occurring simultaneously. For instance, during the testing phase of each version, a dedicated team of two will focus on testing while the rest continue developing subsequent versions. In case of a major error in version 1.0 requiring the entire team's attention, the team can address the issue without compromising progress on version 2.0 due to the parallel development approach. Crucially, each part of the code will be named (e.g., version 1.0) to facilitate version tracking.

## 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.

## Quality Control

Defects will be recorded and tracked as Change Requests, and defect metrics will be gathered (see Reporting and Measurement below).

All deliverables are required to go through the appropriate review process, as described in the Development Case. The review is required to ensure that each deliverable is of acceptable quality, using guidelines and checklists.

Any defects found during review which are not corrected prior to releasing for integration must be captured as Change Requests so that they are not forgotten.

## Reporting and Measurement

Updated 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.

## Risk Management

Risks will be identified in the 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. Some risks we may face is the inability to be able to render the backend into a webpage, unexpected misbehavior of the API calls, and loss of quality due to failure to meet deadlines.

## 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.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

# Annexes

The project will follow the UPEDU process.

Other applicable process plans are listed in the references section, including Programming Guidelines.