**My Pantry**

## Software Development Plan and Team Guide

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change/Comments** |
| **0.1** | **2/3** | **Rachel** | **Collecting and filling all the team members’ information and team responsibilities.** |
| **0.2** | **2/10** | **Rachel** | **Entered some of the CIs after discuss with all the team members.** |
| **0.3** | **2/12** | **Rachel** | **Modified some Cis and start up working on risk management.** |
| **0.4** | **2/13** | **Rachel** | **Discussed the potential risks of the project and finish up the risk matrix after discuss with the team.** |
| **1.0** | **2/14** | **Rachel** | **Finished all the discussed entries and ready to turn in.** |
| **1.1** | **2/14** | **Yang** | **Modify some part of the documents after received Prof. C’s comments.** |

**Scope and Purpose**

The purpose of this document is to provide a plan for the development of the My Pantry. The plan outlines the structure of the team, risks of the project, the configuration management, and software test plan.

# Table of Contents

Introduction…………………………………………………………………………………. 4

Key Terms………………………………………………………………………………….. 4 Team Responsibilities and Structure……………………………………………………. 5 Team Structure…...………………………………………………………………… 5

Team Members…………………………………………………………………….. 5

Team Responsibilities…………………………………………………………….. 5

Additional Responsibilities……………………………………………………….. 5

Risk Management…………………………………………………………………………. 6

Purpose……………………………………………………………………………... 6

Matrix………………………………………………………………………………... 6

Note…………………………………………………………………………………. 7

Configuration Management……………………………………………………………….. 7 Software Test Plan………………………………………………………………………… 9

**Introduction**

Our project is an application that aids consumers in keeping track of their inventory and shopping lists with little hassle. It involves a webpage from which the user will be able to add and remove items from an inventory as well as a shopping list. The shopping list will keep track of the item name, quantity, an image, SKU id, date bought, expiration date and category (a pre-determined list of categories). The shopping list will keep track of item names and quantity needed.

Furthermore, our project includes a database in which the information for the inventory and shopping list is stored. Users will be able to add items to the inventory or shopping list from the webpage, which will then be stored in the database. The mobile team will also be using this database.

**Key terms**

|  |  |
| --- | --- |
| CI | Configuration Item, any document that is considered a single entity. |
| SCMP | Software Configuration Management Plan |
| SPMP | Software Project Management Plan |
| STP | Software Test Plan |

**References**

|  |  |
| --- | --- |
| SPMP | Software Project Management Plan, Rev xx, Date: xxxxxx |
| STP | Software Test Plan, Rev xx, Date: xxxxxx |
| RD | Requirements Document (Term Project.doc) |

**Team Guide Template**

**Team Structure:** Democratic coding

**Team Members:**

|  |  |  |
| --- | --- | --- |
| Rachel | [Rhead932@my.madonna.ed](mailto:Rhead932@my.madonna.ed) | 734-358-1180 |
| Yang | [Yzhang092@my.madonna.ed](mailto:Yzhang092@my.madonna.ed) | 734- 837-9363 |
| Nate | Natepuscas@gmail | 734-233-8558 |
| Zach | [Zacharykasenow@gmail.com](mailto:Zacharykasenow@gmail.com) | 734-223-4661 |

**Team Responsibilities:**

|  |  |
| --- | --- |
| **Functionality** | **Team Member** |
| Database setup and management, help Yang with database connection as needed | Rachel |
| Write code for database connection/adding entries to database | Yang |
| Help with writing HTML code for UI, design UI using Bootstrap | Nate |
| Writing HTML code for UI, helping Nate design UI with Bootstrap | Zach |

**Team Communication:**

* All the team members must meet weekly. Each member will take turns hosting and recording meetings.
* An email with a google documents link was issued by the project manager and team members will post and respond to discuss with other members.
* In other emergency situation, members could call to reach each other.

**Additional Responsibilities**

|  |  |
| --- | --- |
| **Position** | **Team Member** |
| Project Management | Rachel |
| Requirement Management | Yang |
| Test Management | Nate |
| Risk Management | Zach |
| Communication Management | Nate |
| Configuration Management | Yang |

**Risk Management**

**Purpose:**

The purpose of the risk management document is to identify the risks and create a mitigation plan to minimize both the chance and impact of the risk.

**Risk Matrix Table:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk Title** | **Type** | **%** | **Impac t** | **Priority** | **Mitigation Plan** | **Assig n** | **Target Date** |
| Can't get UPC scanning to work | Technical | Very low | moderate | 10 | Contact Mobile team and see what needs to be done | Rachel | 4/6 |
| Server goes down | Technical | Very low | significant | 10 | Find servers that are working | Yang | N/A |
| Can't add entries to database | Technical | Very low | significant | 10 | Go to servers to see what issue is | Yang, Rachel | 3/16 |
| Interface can't be fully developed | Technical/Time | Low | significant | 8 | Work with what we do have developed | Nate, Zach | 3/16 |
| images won't show up | Technical | Very low | moderate | 4 | Work without the images | Nate | 4/6 |
| Connection issues between web and mobile | Technical | High | significant | 10 | Instantly contact the other team and determine where the issue is | Yang | 4/6 |
| Server data retracement | Technical | Very low | significant | 3 | Use backed up versions | Yang | 3/1 |
| Weather issues  - power issue | Natural | Very Low | minor | 1 | Can’t do much | Nate | N/A |
| No internet | Technical | Very low | significant | 9 | Can’t do much | Nate | N/A |
| Database design flaw | Technical | Low | significant | 10 | Fix issue ASAP | Yang | N/A |

**Risk Mana**

**gement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Percentage | Risk | | | | |
|  | Insignificant | Small | Average | Large | Detrimental |
| 91-100% | Low | Low | High | Extreme | Extreme |
| 51-90% | Low | Low | Moderate | High | Extreme |
| 11-51% | Minimum | Low | Low | High | Extreme |
| 1-10% | Minimum | Minimum | Low | Moderate | High |
| 0-1% | Minimum | Minimum | Low | Moderate | High |

**Notes:**

The team will assess the risks on a weekly basis to track and update the risk matrix table.

# Configuration Management

## Introduction Purpose

The purpose of this document is to give guidelines and rules on the storage, layout and identification conventions of all documents that will be created in the course of the Software Project.

## Scope

* List the specific documents which need to be written during the course of the project.
* Give naming conventions for these documents.
* Provide a structured way to create, store and update the documents.

# Management

## Organization

The roles directly involved in the configuration management are:

* Configuration Manager (CM)
* Team members

## Responsibilities

The Configuration Manager (CM) is mostly responsible for the technical part, such as the maintenance of the repository and baseline creation. Both the CM and team members are responsible for keeping the documentation uptodate and correct (e.g. code files without compilation errors). The CM may rename and move files that are incorrectly named or placed. However, the CM may not make significant changes to any project or design documents, and is mainly responsible for the naming and storing of these documents, not the actual contents. The CM is also responsible for making regular backups.

# Activities

## Configuration Identification

The Configuration Items (CIs) that will be written during the project are:

* Requirements Document (RD)
* Requirement Analysis Document (RAD)
* Software Configuration Management Plan (SCMP)
* Software Project Management Plan (SPMP)
* Software Test Plan (STP)
* Code
* Minutes and agenda
* Test plans for Unit Test (UT), System Test (ST), Integration Test (IT)
* Miscellaneous

**CNaomnifnigguCroantivoenntMionasnagement**

We will use all lowercase for naming files.

## Version tags

Every time a new version is created, the version tag will be incremented by 1 (i.e 1, 2, 3, 4...). We will start with version 0.0.1 and then gradually work our way up to higher versions. For example the next version would be 1.1.0 and so on.

## Policies for Changes

When a possible change is presented, it will be discussed and the group will come to an agreement. Then the necessary changes will be made.

## Revision Control

If there is a change to this document, the group will agree on changes, and then make the changes as seen fit.

# Schedule

## Repository Backup

Yang will back up the database and the web source code every day.

# Resources

## Tools

The group will utilize Bootstrap, the database, Slack, Google Docs, and various text editors.

Some tools like open-UI, MySQL.

## Plan Maintenance

Yang will maintenance the code and database on weekly basics.