Functional Specification

Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the heart of South Leinster

Project: Smart Shopping Application

Student: Jason O'Hara

Student Number: C00168956

Supervisor: Paul Barry

Contents

1. Introduction

1.1 Smart Shopping Application

The Smart Shopping Application is a cross-platform, mobile shopping list application. Its key features are:

- Allow a user to create a shopping list in a quick and easy manner.
- Allow a user to save this list to a cloud database.
- Allow the user to add information regarding the location of a product in a given shop to a cloud database.
- Allow a user to sort a shopping list based on the location of products in a given shop.

The functionality that separates this application from the many other basic shopping list applications available is the list-sorting functionality. This uses crowd-sourced data to maintain a database describing the location of products across a selection of different shops and supplies users with the ability to integrate this information directly into their personal shopping list.

The intention is to develop this application in a cross-platform manner with a particular focus on mobile platforms. This application will be developed across three iterations from October 2015 until April 2016 by a single developer.

1.2 Purpose of This Document

The purpose of this document is to give an overview of the expected functionality of this application at the beginning of each iteration. It will also detail any non-functional requirements that must be implemented in conjunction with this functionality and to what extent the functionality specified was altered during implementation.

The implementation of this functionality will be described in greater technical detail in the Design Document of this project, the technologies used will be discussed in detail in the Research Document and the results of this project will be discussed in detail in the Final Report.

2. Overview

2.1 Technology Overview

As discussed in detail in the Research Document of this project, the principal technologies used in this project are:

- **Python:** The programming language used for both frontend and backend development.
- **Kivy**: A Python library used to develop a cross-platform frontend user interface.
- **PythonAnywhere:** A service allows the creation of Python web applications in the cloud.
- **MySQL:** A relational database technology used for storing information in the cloud. The MySQL database used in this project comes as part of the PythonAnywhere service.

While this application is intended to be cross-platform Android is the primary platform.

2.2. Architecture Overview

The architecture of this project is summarized in the context diagram below with an Android device used as an example:

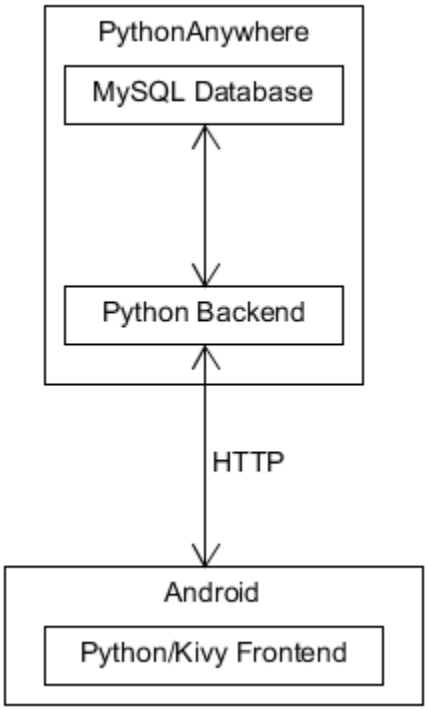

Fig 2.1: A basic architecture overview

3. Functionality

The following sections describe the functionality planned before each iteration as well as any differences between what was planned and what was delivered. Brief and casual use cases are used to describe functionality as needed.

3.1 Iteration 1

This iteration began on the 5th of October 2015 and concluded on the 17th of December 2015.

3.1.1 Add to Shopping List

Informal Description
This involves the creation of a basic interface where the user can type in the name of a product that will then be added to an on-screen list.

Brief Use Case
Actors: User

Description: This use case begins when a user wishes to add an item to their shopping list. The user will enter the name of the product they wish to add to their list. The system will then clearly show that the product specified is now on the user's list.

Non-Functional Requirements

1. This use case should take no longer than 5 seconds for a user with elementary experience of the system.
2. This use case should be successful upon the user's first attempt 95% of the time.

Changes to Planned Implementation
This use case was significantly changed as the user manually typing in the name of a product was too time-consuming and created too many variables to deal with. Rather than the user typing in the name of a product it was decided that the user would select the product they wanted to add to their list from a menu. This menu is populated upon start-up with data stored in the cloud database. The revised use case can be seen below.

Brief Use Case (Revised)
Actors: User
Description: This use case begins when a user wishes to add an item to their shopping list. The user will specify that they wish to add to their shopping list. The system will display a list of product categories for the user to choose from. The user will select a product category. The system will display a sub-list of products to chose from. The user will select a product. The system will clearly show that the product specified is now on the user's list.

3.1.2 Sync Shopping List to Cloud

Informal Description
This involves the user selecting that they wish to sync their shopping list with the cloud. The list will then be sent to the cloud and saved over any older list (if any). Whenever a user opens the application the version of their shopping list they see is the one last synced with the database.

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user wishes to sync their current shopping list to the cloud database. The user will specify that they wish to sync their current shopping list. The system will then save the shopping list to the cloud database.

Non-Functional Requirements

1. This use case should take no longer than 5 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user's first attempt 95% of the time, assuming a stable internet connection.

3.1.3 Basic Shopping List Sorting

Informal Description
This involves the user selecting that they wish for their current list to be sorted. The user's list will then be placed in the same order as a set of pre-loaded data stored in the cloud database. For example: If a user's shopping list is *"Apples", "Pears", Oranges*" and the pre-loaded cloud data is *"Pears", "Bananas"," Apples", "Oranges"*, the user's list will be returned as *"Pears", "Apples", "Oranges".*

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user wishes to sort their shopping list based on information in the cloud database. The user will specify that they wish to order their current shopping list. The system will retrieve the shop information from the cloud database. The system will display the user's shopping list re-ordered based on information received from the cloud database.

Non-Functional Requirements

1. This use case should take no longer than 5 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user's first attempt 95% of the time, assuming a stable internet connection.

3.2 Iteration 2

This iteration began on the 18th of December 2015 and concluded on the 26th of February 2016.

3.2.1 Sorting Shopping List in More Complex Manner

Informal Description
This functionality will expand upon the basic sorting functionality done in the last iteration. The user will select from a list of geographic locations and then from a sub-list of shops at that location. The user's list will then be sorted based on the cloud data for that particular shop.

In addition, rather than simply being sorted into a particular order, the products on a user's list will be clearly categorised based on what aisle they are in the shop selected. Any additional items added to the list after this use case will also be categorised in such a manner.

The shop and location menus are populated on start-up based on information in the cloud database.

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user wishes to sort their shopping list based on information in the cloud database. The user will select the "Sort List" functionality of the system. The system will display a list of geographic locations to chose from. The user will select a location. The system will display a list of shops to chose from. The user will select a shop. The system will sort and categorise the user's shopping list based on information in the cloud database for that particular shop. The system will display the user's list.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user's first attempt 95% of the time, assuming a stable internet connection.

3.2.2 Move Product Within Shop

Informal Description
While a user has a shop selected they may select an item on their list and change its location. This means that they will may select a new aisle for this product from the existing aisles stored for the current shop. Once this is done the aisle information in the Cloud Database (and by extension the data downloaded by all future users) will be altered to reflect the change.

Casual Use Case
Actors: User, Cloud Database

Description: This use case begins when a user wishes to alter the location of a product on their list within the currently selected shop. The user will select a product on their list and then select the “Change Location” functionality. The system will display a list of the current shops aisles to chose from. The user will select an aisle. The system will modify the location of the item in the cloud database. The system will display the user’s list, now refreshed to reflect their modification.

Alternatives: If there is no current shop selected the system will prompt the user to select a shop.

Non-Functional Requirements
1. This use case should take no longer than 5 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

Changes to Planned Implementation
There were some significant alterations to the planned product moving functionality. For the process described above to work well, it would be necessary to allow users to also create and delete aisles. This would have the potential to get quite messy and overly complex.

The product moving functionality was simplified so that every shop has 50 potential aisles. Only the aisles with products in them are ever visible to users. When a user wants to modify the location of a product they will select an aisle number between 1 and 50. If this aisle does not exist yet it will be created, otherwise the product will be added to the existing aisle.

In order to make aisle names more intelligible and intuitive a process was developed were aisles are dynamically named based on the category of products currently in them. For example: *“Aisle 5 - “Fruit, Beverages”*

3.2.3 Login

Informal Description
When a user opens the application for the first time on a new device they will be presented with a login screen. Assuming they have an existing account they will need to enter their username and password to continue to the main application.

Casual Use Case
Actors: User, Cloud Database

Main Success Scenario: This use case begins when a user wishes to log in. The system will prompt the user for their username and password. The user will enter their username and password. The system will verify this information with the cloud database. The system will download the user's information from the cloud database. The system will display the user's main screen.

Alternatives: If the username and/or password entered were invalid the system will notify the user and not move from the login screen.

Non-Functional Requirements
1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user's first attempt 95% of the time, assuming a stable internet connection.

3.2.4 Logout

Informal Description
The user will select that they want to logout from the "My Account" menu. This will clear all the application's locally stored data for that user and show the login screen.

Brief Use Case
Actors: User

Description: This use case begins when a user wishes to log out. The user will select the "Logout" functionality. The system will log the user out and display the login screen.

Non-Functional Requirements
1. This use case should take no longer than 5 seconds for a user with elementary experience.
2. This use case should be successful upon the user's first attempt 95% of the time.

3.2.5 Create Account

Informal Description
This functionality will be accessible from the login screen, it is a basic sign-up system that doesn't require any email or phone number verification.

Casual Use Case
Actors: User, Cloud Database

Main Success Scenario: This use case begins when a user wishes to create an account. The user will select the "Create Account" functionality. The system will prompt the user for a username and a password to be entered twice. The user will enter a username once and a password twice. The system will verify this information locally and with the cloud database. The system will create a new user in the cloud database. The system will notify the user that their account has been created. The system will return the user to the login screen.

Alternatives: The system will display an error message if the username of password to not match specific format criteria, if the two passwords entered do not match or if the username entered is already the name of a registered user in the cloud database.

Non-Functional Requirements
1. This use case should take no longer than 20 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user's first attempt 90% of the time, assuming a stable internet connection.

3.2.6 Create Shop (Website)

Informal Description
This functionality is accessed from the administration website, it allows an administrator to create a shop that will then be visible to all users and can then have aisle information populated within it. If the administrator creates a shop in a location that does not currently exist this location will be created also.

Brief Use Case
Actors: Administrator, Cloud Database

Description: This use case begins when an administrator wishes to create a new shop. The administrator will select the “Create Shop” functionality of the administration website. The system will prompt the administrator for the name and location of the new shop. The administrator will enter the name and location of the new shop. The system will add the new shop to the cloud database. The system will notify the administrator that the new shop has been created.

Non-Functional Requirements
1. This use case should take no longer than 20 seconds for a trained administrator, assuming a stable internet connection.
2. This use case should be successful upon the administrator’s first attempt 90% of the time, assuming a stable internet connection.

3.2.7 Delete Shop (Website)

Informal Description
The administrator will select a shop to be deleted from a list of all shops.

Brief Use Case
Actors: Administrator, Cloud Database

Description: This use case begins when an administrator wishes to delete shop. The administrator will select the “Delete Shop” functionality of the administration website. The system will present the administrator with a list of all current shops to chose from. The administrator will select a shop for deletion. The system will delete all information regarding the selected shop from the cloud database. The system will notify the administrator that the shop has been deleted.

Non-Functional Requirements
1. This use case should take no longer than 10 seconds for a trained administrator, assuming a stable internet connection.
2. This use case should be successful upon the administrator’s first attempt 90% of the time, assuming a stable internet connection.

3.2.8 Move Product Within Shop (Website)

Informal Description

This is essentially the same functionality as the application has for specifying/modifying the location of a particular product in a particular shop. It has been implemented through the administration website in a way that makes it easier to add a large amount of data quickly.

Brief Use Case

Actors: Administrator, Cloud Database

Description: This use case begins when an administrator wishes to modify a product's location within a given shop. The administrator will select the "Move Product" functionality of the administration website. The system will present the administrator with a form requesting they select a shop, a product and an aisle to place it in. The administrator will select a shop, a product and an aisle. The system will update the cloud database to reflect this change. The system will notify the administrator that the change has been made.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a trained administrator, assuming a stable internet connection.
2. This use case should be successful upon the administrator's first attempt 95% of the time, assuming a stable internet connection.

3.3 Iteration 3

This iteration began on the 26th of February 2016 and concluded on the 15th of April 2016. It is the final iteration of this project.

3.3.1 Login (Website)

Informal Description
For security reasons it was necessary to add a simple login system to the administration website. Administrator accounts are created directly by the database administrator.

Casual Use Case
Actors: Administrator, Cloud Database

Main Success Scenario: This use case begins when an administrator wishes access the administration website while not already logged in. The administrator will attempt to open a page on the administration website. The system will prompt the administrator for their username and password. The administrator will enter their username and password. The system will verify this username and password using the cloud database. The system will display the page the administrator initially requested.

Alternatives: If the administrator enters invalid details the system will display an error message and prompt the administrator to re-enter their username and password.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a trained administrator, assuming a stable internet connection.
2. This use case should be successful upon the administrator's first attempt 95% of the time, assuming a stable internet connection.

Changes to Planned Implementation
In order to simplify this process on a technical level this use case was altered so that when an administrator logs in they are always redirected to the home page of the administration website, regardless of the page they were initially attempting to access..

3.3.2 Logout (Website)

Informal Description
An administrator will remain logged in while the browser session is open or until they use the logout feature accessed via the home page.

Brief Use Case
Actors: Administrator

Description: This use case begins when an administrator wishes to log out of the administration website. The user will select the “Logout” feature of the system. The system will log the user out. The system will display a message confirming the user has logged out

Non-Functional Requirements

1. This use case should take no longer than 5 seconds for a trained administrator, assuming a stable internet connection.
2. This use case should be successful upon the administrator’s first attempt 95% of the time, assuming a stable internet connection.

3.3.3 Add List to My Lists

Informal Description
This involves a user saving a list to their “My Lists” section of the application. This part of the application is under the “My Account” menu. The list will then be available to be loaded from that screen in all future sessions, using the name specified during creation.

Casual Use Case
Actors: User, Cloud Database

Main Success Scenario: This use case begins when a user is on the “My Lists” screen and wishes to add a list. The user select the “Add List” functionality. The system will prompt the user for the name of the new list. The user will enter the name of the new list. The system will validate the list name internally and against the cloud database. The system will save the list to the cloud database under the name specified for the current user. The system will display the “My Lists” screen, including the newly added list.

Alternatives: The system will display an error message if the list name does not match specific format criteria, or if the list name entered is already the name of a list the user owns in the cloud database.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

3.3.4 Select List From My Lists

Informal Description
When a user selects a saved list from My Lists it will load that list from the cloud database as the current list and display the main list screen.

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user is on the “My Lists” screen and wishes to select a list. The user will select the list they wish to load. The system will retrieve the list specified from the cloud database. The system will add the list specified to the main list screen. The system will display the main list screen.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

3.3.5 Delete List From My Lists

Informal Description
This is refers to the functionality that allows a user to delete a saved list from “My Lists”.

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user is on the “My Lists” screen and wishes to delete a list. The user will select the “Delete List” functionality of the system. The system will indicate that the user may select a list to delete from the My Lists screen. The user will select the list they wish to delete. The system will delete the list from the cloud database. The system will remove the deleted list from the list of lists on the My Lists screen.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

Changes to Planned Implementation
Due to time constraints, this functionality had to be altered slightly so that when a user selects that they wish a delete a list, they will select the list they wish to delete from a separate screen similar to the My Lists screen. They will then return to the My Lists screen. This is a slightly less elegant solution that was easier to implement.

3.3.6 Add Shop to My Shops

Informal Description
This involves a user saving the currently selected shop to their “My Shops” section of the application. This part of the application is under the “My Account” menu. Any shop on the My Shops screen can be quickly selected and the current list will be ordered based on it as if it were selected from the usual shop menu.

Casual Use Case
Actors: User, Cloud Database

Main Success Scenario: This use case begins when a user is on the “My Shops” screen and wishes to add a shop. The user select the “Add Shop” functionality. The system will save the current shop to the cloud database for the current user. The system will refresh the “My Shops” screen in order to display the newly added shop.

Alternatives: If the user has no shop selected or the shop selected is already a saved shop for the current user the system will display an error message.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

3.3.7 Select Shop From My Shops

Informal Description
When a user selects a saved shop from the My Shops screen the current list will be ordered based on it, as if it were selected from the main shop menu.

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user is on the “My Shops” screen and wishes to select a shop. The user will select the shop they wish to sort based on. The system will retrieve the shop information from the cloud database. The system sort the current list based on the shop information received. The system will display the user’s current list.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

3.3.8 Delete Shop From My Shops

Informal Description
This is refers to the functionality that allows a user to delete a saved shop from “My Shops”.

Brief Use Case
Actors: User, Cloud Database

Description: This use case begins when a user is on the “My Shops” screen and wishes to delete a shop. The user will select the “Delete Shop” functionality of the system. The system will indicate that the user may select a shop to delete from the My Shops screen. The user will select the shop they wish to delete. The system will delete the shop from the user’s saved shops in the cloud database. The system will refresh the My shops screen, showing that the shop selected is now deleted.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.

Changes to Planned Implementation
Due to time constraints, this functionality had to be altered in a similar way to the “Delete List from My Shops” functionality. When a user selects that they wish a delete a shop, they will select the shop they wish to delete from a separate screen similar to the My Shops screen. They will then return to the My Shops screen. As before, this is a slightly less elegant solution that was easier to implement.

3.3.9 Delete Account

Informal Description
This is for when a user wants to completely delete their account. After they confirm via two prompts on the My Account Screen all details regarding their account will be removed from the database.

Casual Use Case
Actors: User, Cloud Database

Main Success Scenario: This use case begins when a user is on the “My Account” screen and wishes to delete their account. The user select the “Delete Account” functionality. Display a prompt confirming the user wishes to delete their account. The user will confirm the wish to delete their account. The system will delete all details of the current user in the cloud database. The system will show the login screen.

Alternatives: If when prompted the user indicates they do not want to delete their account they are returned to the My Account screen.

Non-Functional Requirements

1. This use case should take no longer than 10 seconds for a user with elementary experience, assuming a stable internet connection.
2. This use case should be successful upon the user’s first attempt 95% of the time, assuming a stable internet connection.