**Graded Unit Stage 1B: Solution Planning**

Michael Johnston

(1800329)

Contents

[Introduction 2](#_Toc33707329)

[Class Responsibility Collaborator (CRC) Cards 3](#_Toc33707330)

[UML Class Diagram 3](#_Toc33707331)

[Activity Diagrams 3](#_Toc33707332)

[Use-case Diagram 4](#_Toc33707333)

[Wireframe Design of App 4](#_Toc33707334)

[A short discussion on the interface design 4](#_Toc33707335)

[Appendix (Research Log) 5](#_Toc33707336)

[References 5](#_Toc33707337)

# Introduction

This report serves to document and collate the methods of planning the business and view models of the "Mixing Desk" app. This document will often point to files which will come bundled with this report due to some of those files not being easily depicted in this document. The methods used to analyse the business and view model of the proposed app included:

* Class Responsibility Collaborator (CRC) cards (business model)
* Unified Modelling Language (UML) class diagram (business model)
* Activity Diagrams
* Re-iteration of the use-case model
* Wireframe modelling of the user interface

The rest of this report may give brief explanations of these modelling techniques though it is assumed that the reader will be familiar with reading them.

# Class Responsibility Collaborator (CRC) Cards

To get a general feel for the classes and other internal workings of the app would operate, CRC cards were created. These describe roughly what classes will be necessary in the finished version of the app. The CRC cards are presented in the file entitled CRCBusinessModelForApp.pptx.

# UML Class Diagram

The file businessModelUMLClassDiagram.pdf shows the expected relationships between classes, their expected attributes and methods along with data types of member attributes and parameters. The finished code may not contain explicit member attributes due to the Property types of C# hiding the attributes behind the Property.

# Activity Diagrams

The files addingFlavouringToPersonalStashActivityDiagram.pdf, browsingRecipesAndMovingToMixingActivityDiagram.pdf, usingMixingCalculatorAndSharingRecipeActivityDiagram.pdf and viewingPersonalStashFlavouringsActivityDiagram.pdf show activity diagrams for the each of the main functionalities of the proposed app. These diagrams both show what the user's expected input-steps will be and what some of the internal logic of the app will be in response to the user.

# Use-case Diagram

Shown in the file stage1ATopLevelUseCaseDiagram.png is the top-level use-case diagram as made for the first stage of this project.

# Wireframe Design of App

The files mixingCalculatorWireframe.png, personalStashWireframe.png, recipesThisMonthWireframe.png, recipesThisWeekWireframe.png, recipesThisYearWireframe.png and specificRecipeWireframe.png are include along with this document and are shown further on in this document.

As we have previously stated in the inception report, one of the constraints in developing this app is that it must be made using universal windows platform (UWP) and must be coded in C#. This is the databinding model and it is how the view (the interface presented to the user) is linked to the business model (the internal logic of the app). To be even more specific as to how this link is achieved, event handlers within the code of the app will respond to user input.

## A short discussion on the interface design

In the spirit of one of the functional requirements of this app (The app should be easy to navigate) the use of sidebar navigation was chosen to allow easy navigation between the main sections of the app. This sidebar does not expand instead, to access the differently filtered list of recipes the user simply navigates to the recipe page and uses a button on each recipes page to navigate to the next recipes page in a navigational loop. This keeps the sidebar from having three occurrences of the word "Recipes" potentially confusing users. The rest of the app uses simplistic interface features such as textboxes, lists and buttons which should be easily familiar to potential users.

# Appendix (Research Log)

This section documents the research that was done in the analysis of this app's business and view model.

**12/02/2020:**

This developer researched materials (Visual Paradigm, n.d.) on how to make activity diagrams and what they depict.

**26/02/2020:**

This developer researched UML class diagrams (Salma, 2017) and learned about relationships in UML class diagrams in particular how to specify multiplicity, composition and aggregation. This developer also learned that you should include the data types in UML class diagrams. The rest of the app uses simplistic interface features such as textboxes, lists and buttons which should be easily familiar to potential users.

# References

Salma. (2017, September 1). *UML Class Diagrams Tutorial, Step by Step*. Retrieved from Medium: https://medium.com/@smagid\_allThings/uml-class-diagrams-tutorial-step-by-step-520fd83b300b

Visual Paradigm. (n.d.). *What is Activity Diagram?* Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/