**Milwaukee Center for Independence Food Service Application:**

**Preparation Portal**

**12/7/15**

## 

## Tessa Shakula Philip Schmeiser Tom Jacobson

# 

# **1.** **Introduction**

**1.1 Purpose**

This software design document specifies the architecture and system design of the **preparation portal** of the **MCFI Food Service Application**. This document should provide a detailed description of the designer’s goals and constraints for the **preparation portal**.

**1.2 Scope**

This specification should provide sufficient description for implementers to understand what and how to implement design, and develop the application fitting the requirements in the *MCFI Food Service Application: Preparation Portal Software Requirements Specification* document.

**1.3 Overview**

* System Overview
* Design Considerations
  + Assumptions and Dependencies
  + General Constraints
  + Goals and Guidelines
  + Development Methods
* Architectural Strategies
  + strategy-1 name or description
* System Architecture
  + component-1 name or description
* Policies and tactics
  + policy/tactic-1…
* Detailed System Design
  + module-1…
* Glossary
* Bibliography

**1.4 Definitions and Acronyms**

|  |  |
| --- | --- |
| **Vocabulary** | **Definition** |
| API | An abbreviation of ***a****pplication* ***p****rogram* ***i****nterface*, is a set of [routines](http://www.webopedia.com/TERM/R/routine.html), [protocols](http://www.webopedia.com/TERM/P/protocol.html), and tools for building [software applications](http://www.webopedia.com/TERM/A/application.html). The API will act as a buffer between what the **user** sees and the **database**. |
| Approve | An action done on an order that validates the entered values in an **order form** as seen in model 4.1. |
| Approvable | A document is approvable if the **approve** action can be enacted on the document by a user. |
| Approved orders | An order from a **client** that has been verified and **approved** by a **MCFI Staff** member. |
| Audit | An official inspection of **MCFI’**s records. Expected to be for health or tax purposes. |
| Auto-Populate | **Auto-Populate** is the process of a section of a **spreadsheet** that automatically gets information added to it based on what is inputted into another section or **spreadsheet**. |
| Bill | See model 4.5 |
| Client Portal | The **client portal** is the **project** **module** that allows **clients** to log onto the **interface** to place and cancel **orders**. |
| Clients | The **clients** are the people who order food from **MCFI** to be delivered to a **site**. |
| Compatible | Able to exist or occur together without conflict |
| Database | A **database** is a structure used to store various forms of data. |
| Editable | A document is **editable** if the information contained can be changed by a user. |
| Food Service Process | The **food service process** is the entire process of the **MCFI** food service from ordering, cooking and distributing. |
| Interface | Graphical part of the program that the **user** can see and interact with. |
| Kitchen | A **kitchen** refers to the locations manned by **Kitchen Staff** that produce and deliver the food **orders**. |
| Kitchen Staff | **Kitchen Staff** are the people who work in the kitchens during the **food service process**. |
| Manipulate | **Manipulate** refers to the editing, approving, deleting and viewing of **spreadsheets** in the **preparation portal**. |
| MCFI | **MCFI** is an abbreviation of ‘Milwaukee Center for Independence’. **MCFI** is the not-for-profit which is requesting the **MCFI Application**. |
| MCFI Application | The **MCFI application** is the project that is being created for **MCFI.** The project revolves around three aspects called the **project modules.** The project will be used by **MCFI staff**, **kitchen staff**, and **clients** of **MCFI** upon completion to optimize the ordering, creating, and delivering of food. |
| MCFI Staff | The MCFI staff are the employees who work with MCFI and pass information directly from **clients** and **kitchen staff** in the **food service process**. The **MCFI staff** does not include **kitchen staff**. |
| Order | An order contains counts of meals per day in a week to be made for a **site**. See model 4.1 |
| Order form | The form the client fills out. See model 4.1 |
| Preparation Portal | The **preparation portal** is one of the three **project modules**. The **preparation portal** is where **kitchen staff** and **MCFI staff** will be able to log in and view what **clients** have submitted and prepare the **order**. |
| Printable | A document is **printable** if it can be printed by a user. |
| Production Assignment | See model 4.3 |
| Project Module | Thethree **project modules** are the main divisions of the **MCFI application** which make the project run: the database, client portal, and **preparation portal**. |
| Recipe | See model 4.6 |
| Site | A **site** is a location that food is ordered for and delivered to. |
| SRS | **SRS** is an abbreviation for Software Requirement Specification which is also the name of this document. The **SRS** documents the specifications and implementations of the **preparation portal** for the **MCFI application.** |
| Spreadsheets | A **Spreadsheet** is a digital document, in the form of an excel spreadsheet, that will be used to display information throughout the **preparation portal**. |
| Transport Sheet | See model 4.4 |
| User | The **user** is anyone who is using the **application**; this could be a **MCFI staff** member, **client**, and/or **kitchen staff** member. |
| Viewable | A document is **viewable** if a user can access and view it. |
| Visual Basic.Net | A multi-paradigm, high level programming language, implemented on the .Net Framework, a Microsoft integrated development environment. |
| Web Page | A document connected to the World Wide Web. |
| Weekly Count | All information from **approved orders** is **auto-populated** into the **weekly count**, see model 4.2. |

## 

### 2. System Overview

The **Preparation Portal** of the **MCFI Food Service Application** provides an **interface** to allow **users** to interact with a **database** through a custom **web page** in order to fulfill the requirements detailed in the *MCFI Food Service Application: Preparation Portal Software Requirements Specification* document.

This **interface** will allow **users** to view, edit, approve and print various **spreadsheets** required for the **preparation portal** module of the **MCFI application**.These **spreadsheets** are initially populated after the **client** submits an order through the **client portal**. The **interface** then gives the **user** the power to view individual **spreadsheets**, edit them, and print them.

# 3. Design Considerations

# This project must be designed in such a way that progress can be displayed to **MCFI Staff** and presented to our peers and the project manager, Professor Riley, within the semester of the Software Engineering class.

## 3.1. Assumptions and Dependencies

**3.1.1 Assumptions**

* Users do not have much experience working with a computer.

**3.1.2 Dependencies**

* Microsoft C# .Net 4.5
* Microsoft SQL Server 2012

## 3.2. General Constraints

* Data stored in the **database** must be secure.
* The lack of an internet connection prohibits the use of the **Preparation Portal**.

## 3.3. Goals and Guidelines

**3.3.1 User Experience**

* The user **interface** should be *simple and intuitive* to use.
  + The user **interface** should closely resemble the physical documents the users have previously used.
* The **MCFI application** should allow **users to**

**3.3.2 Internal Simplicity**

* The code should strive for simplicity and modularity.
* The code should be maintainable and updatable.
  + Development of the **preparation portal** must consider that the final release (beyond the scope of this project) may be expanded for more features.
* **User** **input** should be checked and sanitized when possible.
  + Invalid input such as negative numbers of meal counts or text when numbers are required should prompt an error to the user.

## 3.4. Development Methods

The Waterfall software process model will be used to plan, develop, and implement the **preparation portal**. The Waterfall process was chosen because it is based around structure, is linear, and is easy to use to develop a large project that involves many steps. The preparation portal section of the project should be developed incrementally by first solidifying requirements, then moving on to further steps of the model. Nearly all the knowledge of needed requirements and developmental tools is available so the waterfall process model is ideal. For more information on the Waterfall software process model, visit: https://en.wikipedia.org/wiki/Waterfall\_model.

**4.** **Architectural Strategies**

**4.1 Architectural Design**

The **preparation portal** shall follow the Model View Controller design pattern for developing this software. The model is the lowest level of the pattern which is responsible for maintain data. The view will be responsible for displaying the data to the user. The controller will be the code that controls the interactions between the model and view.

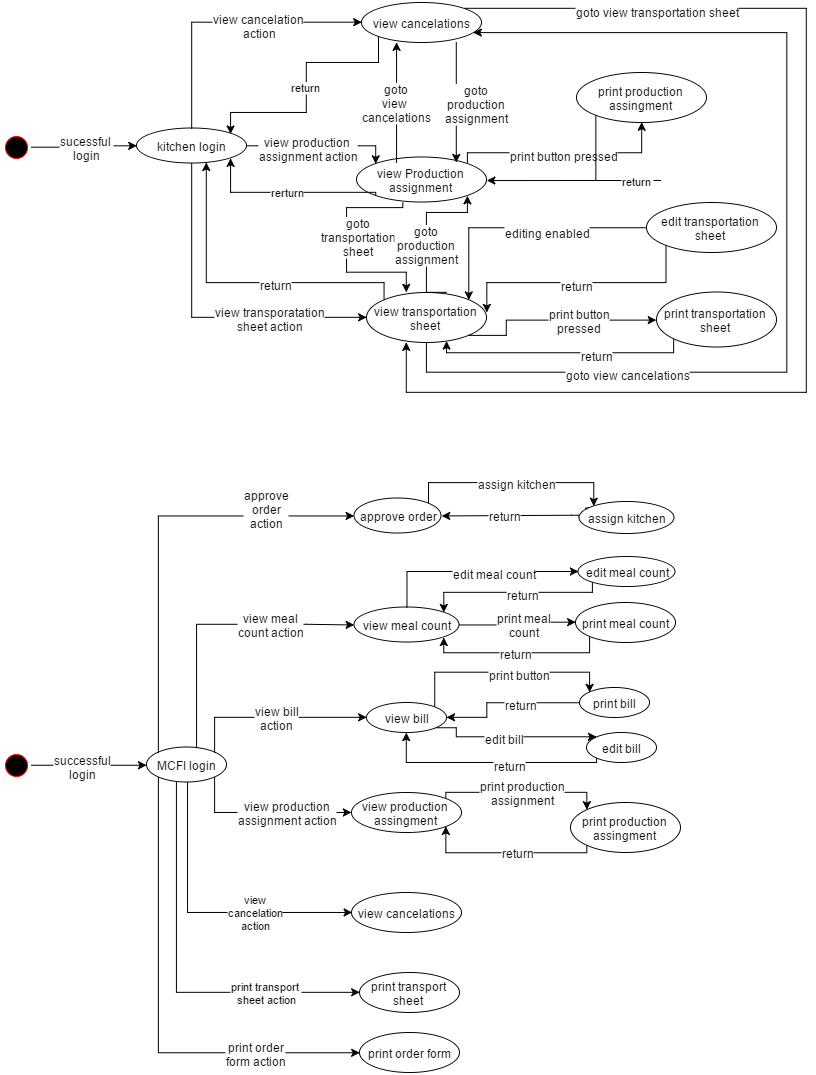
# **5.** **System Architecture**

**5.1.** **Behavioral Diagrams**

**5.1.1 Activity Diagram**

The **MCFI Staff** shall be able to assign a **kitchen** and **approve** new **orders**. They will also be able to edit, view, and print the **weekly counts** form. **MCFI Staff** shall be able to view **cancellations** from any of their clients. They shall also be able to view, edit, and print the **production assignment**. **MCFI Staff** shall be able to view, edit, and print their **clients bills**. An **order form** and **transportation sheet** will be available for **MCFI** to print.

The **kitchen staff** shall be able to view and print their **production assignment**. **Kitchen staff** shall be able to view, edit, and print a **transportation sheet**. **Kitchen staff** should be able to view any relevant **cancellations**.



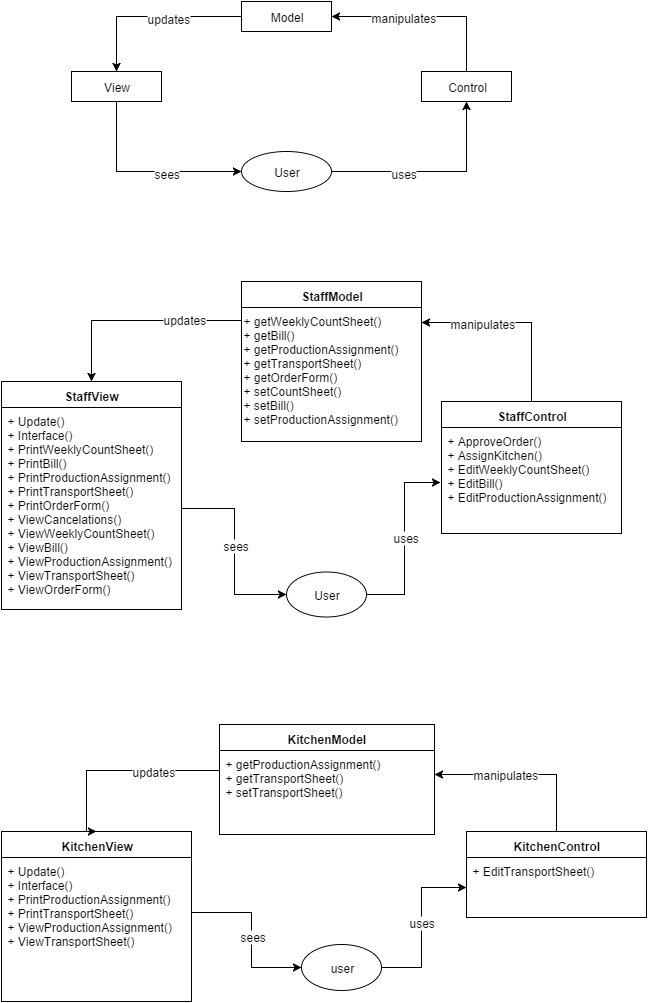
**5.1.2 Sequence Diagram 1**

The **client** shall login. Once logged in successfully they shall be able to retrieve an **order form**. After they fill out the **order form** they will submit that to **MCFI Staff**. There a staff member after a successful login will be able to get the new **orders**, **approve** them, and assign the new **order** a **kitchen**. From there a successfully logged in **kitchen staff** member shall be able to retrieve their **production assignment** and fill out a **transport sheet**. After all these processes are finished the **bill** can be **auto-populated**.

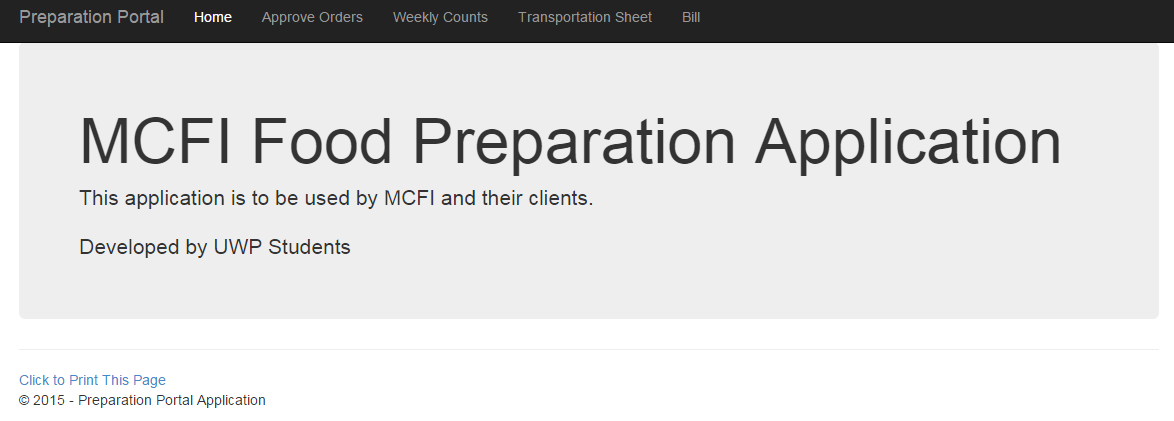
SequenceDiagramPNG.png

**5.1.3 UML Class Diagram**

MVC framework breaks the system down into three parts to run more efficiently, the model, the view, and the control. **MCFI staff** will be able to do five controls, they will be able to approve an order, assign a kitchen, edit the **weekly count** sheet, edit the **bill**, and edit the **production assignment**. The model will retrieve information from the database. **MCFI staff** will be able to access the **weekly count** sheet, the **bill**, the **production assignment**, the **transport sheet**, and the **order form** from the database. For the view, **MCFI staff** will be able to view and print the **weekly counts** sheet, the **bill**, the **production assignment**, the **transport sheet** and the **order form**. **MCFI staff** will also be able to view cancelations.   
 The **Kitchen staff** will have one thing they can control, this will be the ability to edit the **transport sheet**. The kitchen staff will be able to access the **production assignment** and the **transport sheet** from the database. From the view, the **kitchen staff** will be able to view and print the **production assignment** and the **transport sheet**.



**5.1.4 Web Form Layout**

The web form layout shall having appropriate navigational tabs to navigate through the pages. What tabs are available shall depending on whether the user is a MCFI or kitchen staff member. The home page shall state the user’s name in the upper right hand corner.   
 

# **6.** **Detailed System Design**

# **7. Requirements Matrix**

|  |  |
| --- | --- |
| **Functional Requirements from Campus Navigation Requirements Specification 3.1 (id CN-RS3.1)** | **Covered in section** |
| 1. **Unapproved orders** shall be **viewable** by **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 2. **MCFI Staff** shall be able to edit all unapproved **orders** by selecting an edit button. | See section 5.1.1  See section 5.1.2  See section 5.1.3 |
| 3. The **weekly counts** shall be **viewable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 4. **MCFI Staff** shall be able to print all unapproved **orders** by selecting a print button. | See section 5.1.1  See section 5.1.2 |
| 5. **MCFI Staff** shall be able to select an approval button once a unapproved order is ready to be approved. | See section 5.1.1  See section 5.1.2  See section 5.1.3 |
| 6. Once a unapproved order is approved it will be added to the **weekly counts**. | See section 5.1.1  See section 5.1.2  See section 5.1.3 |
| 7. The **weekly counts** shall be **editable** by **MCFI Staff**  by clicking an edit button. | See section 5.1.1  See section 5.1.2 |
| 8. The **weekly counts** shall be **viewable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 9. The **weekly counts** shall be **printable** by the **MCFI Staff** by selecting a print button. | See section 5.1.1  See section 5.1.2 |
| 10. The weekly counts shall be **viewable** by the **Kitchen Staff**. | See section 5.1.1  See section 5.1.2 |
| 11. The weekly counts shall be **printable** by the **Kitchen Staff** by selecting a print button. | See section 5.1.1  See section 5.1.2 |
| 12. The **production assignment** shall be **editable** by **MCFI Staff**  by clicking an edit button. | See section 5.1.1  See section 5.1.2 |
| 13. The **production assignment** shall be **viewable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 14. The **production assignment** shall be **printable** by the **MCFI Staff** by selecting a print button. | See section 5.1.1  See section 5.1.2 |
| 15. The **production assignment** shall be **viewable** by the **Kitchen Staff**. | See section 5.1.1  See section 5.1.2 |
| 16. The **production assignment** shall by **printable** by the **Kitchen Staff** by selecting a print button. | See section 5.1.1  See section 5.1.2 |
| 17. The **production assignment** shall be auto populated for each specific kitchen site. | See section 5.1.1  See section 5.1.2  See section 5.1.3 |
| 18. The **transportation sheet** shall be **viewable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 19. The **transportation sheet** shall be **editable** by **MCFI Staff**  by clicking an edit button. | See section 5.1.1  See section 5.1.2 |
| 20. The **transportation sheet** shall be **printable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 21. The **transportation sheet** shall be **viewable** by the **Kitchen Staff.** | See section 5.1.1  See section 5.1.2 |
| 22. The **transportation sheet** shall be **editable** by **Kitchen Staff**  by clicking an edit button. | See section 5.1.1  See section 5.1.2 |
| 23. The **transportation sheet** shall be **printable** by the **Kitchen Staff**. | See section 5.1.1  See section 5.1.2 |
| 24. The **transportation sheet** shall be auto populated for each client site. | See section 5.1.1  See section 5.1.2  See section 5.1.3 |
| 25. The **bill** shall be **viewable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 26. The **bill** shall be **editable** by **MCFI Staff**  by clicking an edit button. | See section 5.1.1  See section 5.1.2 |
| 27. The **bill** shall be **printable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 28. The **bill** shall be auto populated. | See section 5.1.1  See section 5.1.2  See section 5.1.3 |
| 29. **Cancellations** shall be **viewable** by the **MCFI Staff**. | See section 5.1.1  See section 5.1.2 |
| 30. **Cancellations** shall be viewable by the **Kitchen Staff**. | See section 5.1.1  See section 5.1.2 |

# **8.** **Bibliography**

1. [Internet: Definition of Database, Dictionary.com](http://en.wikipedia.org/w/index.php?title=Dijkstra%27s_algorithm&oldid=626530538)*http://dictionary.reference.com/browse/database*
2. Internet: Definition of API, Vangie Beal  
   *http://www.webopedia.com/TERM/A/API.html*