

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Vision Document

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	Introduction.....	4
2	Positioning	4
2.1	Problem Statement.....	4
2.2	Product Position Statement.....	4
3	Stakeholder Descriptions	5
3.1	Stakeholder Summary.....	5
3.2	User Environment.....	5
4	Product Overview	6
4.1	Product Perspective.....	6
4.2	Assumptions and Dependencies	6
4.3	Needs and Features	7
4.4	Alternatives and Competition	7
5	Other Product Requirements	8
Appendix A	References.....	9
Appendix B	Glossary	10

Table of Figures

Figure 4-1 System Overview	6
----------------------------------	---

Touch For Food

Vision Document

Version 7.12

Revision History

Date	Rev.	Description	Author(s)
2012-09-17	0.00	Document Creation	Katrina Anderson
2012-09-20	0.01	Contributed Section 3 Stakeholder Descriptions	Mikhail Levkovsky
2012-09-22	0.02	Contributed Sections 2.1 and 2.2	Cynthia Donato
2012-09-22	0.03	Contributed Section 4	Christian Daher
2012-09-22	0.04	Contributed Sections 1 and 5	Katrina Anderson
2012-09-23	0.05	Revised Section 3 Stakeholder Descriptions	Mikhail Levkovsky
2012-09-29	0.06	Made grammar/spelling corrections. Fixed formatting issues.	Josh Hum
2012-09-29	0.07	Revised Section 1, 2 & 5. Created works cited.	Katrina Anderson
2012-09-30	0.08	Revised Section 3	Cynthia Donato
2012-11-04	1.09	Reviewed and made minor corrections	Josh Hum
2012-12-16	3.10	Reviewed document and reduced redundancies	Katrina Anderson
2012-12-17	3.11	Reviewed document and corrected defect in Section 5	Matthew Tam
2013-02-06	7.12	Review Document and made necessary corrections and indicated features which have been scoped out	Cynthia Donato

1 Introduction

The software solution proposed in this document is a web application called Touch For Food (TFF), which is aimed at restaurants with the goal of increasing their staff efficiency and to provide a higher level of service to their clients using mobile technologies. TFF will use NFC tag technology to allow users to view menus of subscribed restaurants, place orders and request assistance from their waiter/waitress. TFF will also allow restaurants to outsource tasks from staff to customers, personalising a client's experience and extending the reach of their advertising thanks to social media. This document is responsible for: describing the high level goals for TFF, identifying the stakeholders, providing an overview of the desired product and determining basic high level non-functional requirements [1].

2 Positioning

2.1 Problem Statement

The problem of	incorporating the convenience of mobile technology and social media into a restaurant client's dining experience.
Affects	restaurant clientele with mobile devices.
The impact of which is	clients are forced to wait for a waiter to physically appear at their table in order to place food orders, manage bills or request other services from staff.
A successful solution would be	to create an application accessible through a web browser in a mobile device and downloadable as an application for most mobile devices. This would allow clients dining at a restaurant to order food, request services from the staff, integrate their dining experience with social networking sites and manage their bills.

2.2 Product Position Statement

For	the general public that own mobile devices.
Who	want to incorporate the convenience of mobile technology into their dining experience.
Touch for Food	is an application
That	will allow clients to place food orders, view food reviews, communicate with staff and manage their bills from their mobile devices.
Unlike	ordering through the waiter/waitress
Our product	is available on a device that clients already carry around with them everywhere. It will allow them to drastically speed up their dining experience by eliminating the need to wait for a waiter/waitress to physically come over to the table in order to communicate a need. Our product will also provide food reviews and statistics from other dinners and the ability to create your own review of the food and/or the restaurant once your dining experience is complete.

3 Stakeholder Descriptions

3.1 Stakeholder Summary

Name	Description	Responsibilities
Customer	A user who uses TFF to interact with a restaurant or as a social media tool.	-To use TFF to place orders and review meals and restaurants.
Restaurant Manager	The person who runs operations at a restaurant.	-To provide CloudNine the data that is necessary for certain aspects of TFF (i.e. menus, calorie count, specials...). -To provider the waiters/customers with the NFC tags/QR codes so TFF can be used in their restaurant.
Waiter	The individual that serves customers at a restaurant.	-To provide the customers with the NFC tags/QR codes to use TFF in the restaurant. -To manage orders and bills.
Dr. Constantine Constantinides	SOEN 490 coordinator and instructor.	-To provide timely feedback on the progress of the Capstone project based on deliverables.
Dr. Peter Grogono	SOEN 490 customer stakeholder.	-To provide feedback on design decisions based on demonstrations and deliverables.
Online distributor	The platform that will allow TFF to be distributed to users (i.e. App Store, Google Play...)	-To allow users to download TFF on their respective platforms. -To provide a legally binding agreement between the user and TFF.
Team CloudNine	Software engineering students.	-Analyze possible solutions for the TFF software. -To create a software solution based on the designs of TFF. -Ensure a high quality software product that fulfills the needs that TFF describes.

3.2 User Environment

The target market of the TFF application is restaurant clientele and restaurants themselves. Restaurant clientele will use TFF in two separate working environments. The first environment is in a restaurant, where they will use TFF to order food and request other services from the restaurant. The second environment is anywhere outside of a restaurant, where clients can interact with the social media aspects of TFF, meaning they can leave and read reviews. Restaurants will use TFF on location and will have the unique environmental constraint of providing wireless internet access to their staff and clientele.

Currently the largest mobile system platforms are Android, iOS and Windows. TFF will need to integrate with these major system platforms either with custom applications or via mobile browser. Lesser known platforms, like Blackberry and Symbian, will not be a priority. Possible future integration in Quebec can be with the current major restaurant management system: Maitre'D.

4 Product Overview

4.1 Product Perspective

Each restaurant client will connect to the TFF system by scanning a QR code or tapping an NFC tag. The QR code or NFC tag will be accessible by the client from a spot on the restaurant's table. The user can then interact with the system using their mobile device. The TFF system will be expected to do most of the computations on the server side and will be connected to the database. The restaurant terminal will be the gateway for the restaurants to do managerial operations on TFF.

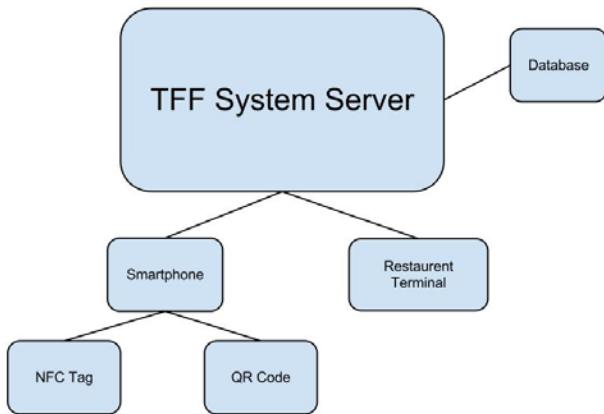


Figure 4-1 System Overview

4.2 Assumptions and Dependencies

Assumptions	Dependencies
Internet or Data connection available	The TFF application uses an internet connection to communicate with the TFF system server.
Camera or NFC enabled smartphone	The TFF application can work using only an NFC tag, but if the user does not have an NFC enabled mobile device, a camera is required to scan the QR code.
The restaurant owns a computer connected to the internet	The restaurant terminal requires a computer with an internet connection in order to interact with the TFF system server.

4.3 Needs and Features

Need	Priority	Features	Planned Release (Milestone#)
Order food (customer)	High	Order management, bill management	3
Make reservation (customer)	Medium	Food reservation, table reservation	5
Call waiter (customer)	Low	Assistance	5
Manage personal page (customer)	High	Review order history, managing friends, social networking	4
View menu (customer)	High	Menu management	2
Comment (customer)	High	Social networking, rating system	3
View restaurant stats and reviews (customer)	High	Social networking, rating system,	2
Accountability, identify priorities for customers (restaurant)	Medium	Customer management	4
Manage menu (restaurant)	High	Menu management	2
Manage tables (restaurant)	Medium	Table reservation	4
Manage bills (restaurant)	High	Bill management	3
Restaurant page (restaurant)	High	Social networking, menu management, order management	4
Restaurant reporting and statistics (restaurant)	Low	Reporting system	5

4.4 Alternatives and Competition

Alternatives or Competition	Benefits	Disadvantages
Open Table	Review reserved table, comment, statistics, location based, smartphone app	Weak social networking, no food ordering/reservation
Eveve	Reserve table, table management	No review, no comment, no food reservation, no mobile app, not independent application
Google places	Review, comment, social networking, location based, smartphone app	No reservation system for food or table, no table management

5 Other Product Requirements

Architecturally, the TFF application is concerned with speed, reliability, security and concurrency.

With respect to speed, the time that TFF will take to query the database to retrieve menus, record orders, save user credentials and other operations are crucial to the application's success. For the application to be useful, data transfer should be fairly instant.

Application services, such as menu lookup and ordering, should be available 99% of the time when users are connected to the internet through their mobile devices. TFF requires high availability as it is competing with the current standard of physical restaurant menus, which is 100% availability.

Concerning security, the TFF application may contain sensitive client information, such as food. Measures must be taken to protect restaurant clientele for their health and privacy concerns.

Lastly, restaurants have limitations in supply, staff and capacity. Methods of managing simultaneous demands must be instituted to address these limitations.

Any maintenance needed for TFF should be done outside operating hours for the restaurant, which is after the store closes and before the store opens: restaurants with breakfast, lunch and supper would undergo maintenance between the time after dinner service and before breakfast.

TFF should be fairly intuitive; however, a user manual for the TFF application should be available online. The application will be distributed online through an official website and through app markets for Windows, Apple and Android mobile devices. Labeling specifying minimum system requirements must be provided.

Appendix A References

- [1] K. Anderson, C. Donato, J.Hum, M. Levkovsky, A. Lloyd, P. Modafferi, "F.S.T.S," Concordia University, Montreal, Canada, Vision Document, v6.16, 2012.

Appendix B Glossary

Refer to the SRS document - Appendix B Glossary and Appendix C Acronyms for a complete list of terms and definitions.

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Supplementary Requirements & Specifications

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	Introduction.....	5
1.1	Purpose	5
1.2	Scope.....	5
1.3	Definitions, Acronyms and Abbreviations	5
1.4	References.....	5
1.5	Overview.....	5
2	Functionality	5
2.1	Restaurant Client User Controls	5
2.2	Restaurant Administrative Controls.....	6
3	Usability	6
3.1	Required training Time for a Restaurant Client.....	6
3.2	Required Training Time for a Restaurant Administrator.....	6
4	Reliability.....	6
4.1	Categorization of Bug Types	7
5	Performance	7
5.1	Response Time for Ordering Food	7
5.2	Orders per Second.....	7
5.3	Concurrent Orders.....	8
5.4	Customer Capacity.....	8
5.5	Heavier Load Performance	8
5.6	Native App Hard Drive Space	8
5.7	Real Time Check in	8
5.8	Throughput for Optimal Usage Mode.....	8
5.9	Sub Optimal Performance Mode	8
6	Supportability.....	8
6.1	Coding Standards & Naming Conventions.....	8
6.2	Maintenance Group Response Time	8
7	Design Constraints	9
7.1	Mobile platform Development.....	9
7.2	Web application Front End and Back End Development	9
7.3	Iterative Development Cycles (Agile)	10
7.4	JIRA for Agile Development Issue Tracking and Management.....	10
7.5	SVN with Assembla Server for Version Control.....	10
7.6	Visual Studio Development	10

8	Online User Documentation and Help System Requirements	11
9	Purchased Components	11
10	Interfaces.....	11
10.1	User Interfaces	11
10.2	Hardware Interfaces	12
10.3	Software Interfaces	13
10.4	Communications Interfaces	13
11	Licensing Requirements.....	13
11.1	Mobile Licenses.....	13
11.2	Domain Name	13
11.3	Copyright	13
11.4	Legal disclaimer.....	13
Appendix A	References.....	14
Appendix B	Glossary	15
Appendix C	Acronyms.....	15

List of Tables

Table 4-1	Categorization of Bug Types.....	7
Table 7-1	Operating system features to be used	9

Touch For Food

Supplementary Requirements & Specifications

Version 7.19

Revision History

Date	Rev.	Description	Author(s)
2012-09-17	0.00	Document Creation	Katrina Anderson
2012-09-20	0.01	Contributed Purchased Components, Supportability, and Glossary	Josh Hum
2012-09-22	0.02	Added reliability, online user documentation and help system requirements sections	Cristian Asenjo
2012-09-22	0.03	Added interfaces, licensing requirements, and legal, copyright and other notices	Ryan Nasr
2012-09-22	0.04	Added Introduction, Functionality and Usability	Matthew Tam
2012-09-22	0.05	Added Design Constraints and Performance	Patrick Modafferi
2012-09-23	0.06	Added Categorization of Bug Rate to Reliability, updated Glossary	Josh Hum
2012-09-25	0.07	Reviewed Ryan's part	Patrick Modafferi
2012-09-29	0.08	Reviewed Pat's part	Cristian Asenjo
2012-09-29	0.09	Made spelling, grammar and formatting corrections	Josh Hum
2012-09-30	0.10	Reviewed Cristian's part	Matthew Tam
2012-11-04	1.11	Reviewed doc and made minor corrections	Josh Hum
2012-12-16	1.12	Added information to design constraints section.	Cristian Asenjo
2012-12-17	1.13	Changed caption for table 4-1	Cristian Asenjo
2012-12-17	1.14	Updated the glossary.	Patrick Modafferi
2012-12-17	1.15	Added 2 extra types of bug classes.	Cristian Asenjo
2012-12-17	3.16	Fixed defect, fixed formatting and reviewed entire document	Matthew Tam
2012-12-20	4.17	Reviewed document and reduced redundancies	Katrina Anderson
2013-02-10	7.18	Reviewed document and updated	Cynthia Donato
2013-02-12	7.19	Reviewed document	Josh Hum

1 Introduction

1.1 Purpose

This document shall list the non-functional, technical, quality, legal, technological and other types of requirements pertaining to TFF. Other important functional requirements will also be included here. These requirements serve as a reference to ensure that the team adheres to performance, quality and other technical standards. It also makes sure that these aspects of the application are thought out and written early in the development process. Please refer to the user stories found in the Management Document and the architecture artefacts found in the SAD, for the major functional requirements.

1.2 Scope

The contents of this document discuss all additional requirements to accompany the vision document, management documents and SAD.

1.3 Definitions, Acronyms and Abbreviations

Please see Appendix B Glossary and Appendix C Acronyms of this document.

1.4 References

Please see Appendix A, References, of this document.

1.5 Overview

The supplementary specifications document is split into thirteen sections. It begins with the Functionality section, which describes the functional requirements of the TFF application in natural language. The next four sections outline the quality attributes of the system: Usability, Reliability, Performance and Supportability. More requirements are specified in the Design Constraints, Online User Documentation, Purchased Components and Interface requirements sections make up the center portion of the specifications document. The necessary legal and regulatory qualifications are outlined in the Licensing Requirements, Legal, Copyright and Other Notices and Applicable Standards sections, before finishing off with a Glossary of terms, acronyms and abbreviations [1].

2 Functionality

2.1 Restaurant Client User Controls

2.1.1 Description

The TFF application shall allow a restaurant client to place their food orders to the restaurant, manage how the bill will be separated, call for assistance, manage their personal page by reviewing past meals ~~and adding food friends~~, leave a customer satisfaction card, and review ~~and recommend restaurants~~.

2.1.2 Input

Restaurant client controls shall be accessed through a mobile device.

2.1.3 System Processing

The TFF application shall handle all requests from a restaurant client.

2.1.4 Output

The restaurant's administrators shall be notified of the restaurant client's requests

2.2 Restaurant Administrative Controls

2.2.1 Description

The TFF application shall allow a restaurant administrator to manage: the items on their menus; ~~table reservations~~; customer orders and bills; ~~customization of their restaurant page; statistics and prioritize customers.~~

2.2.2 Input

Administrative controls shall be accessed through a computer with access to an internet connection.

2.2.3 System Processing

The TFF application shall handle all requests from an administrative user.

2.2.4 Output

Administrators will see the changes applied to their restaurant menus, orders and bills.

3 Usability

3.1 Required training Time for a Restaurant Client

Since the TFF application is supposed to be intuitive to use, the training time for a restaurant client will be two hours.

3.2 Required Training Time for a Restaurant Administrator

The TFF application will introduce a new component that will expedite restaurant service. The level of service should not be dropped by using the application; therefore the training time for a restaurant administrator will be one day.

4 Reliability

The reliability of this app is dependent on the following factors:

- The Mean Time Between Failure (MTBF) of the web services server.
- The load on the server at any given time.
- The stability and speed of the data connection kept by the mobile device.
- The stability and speed of the Internet connection used by the restaurant.

Of those four factors, only the first two are directly within the control of the TFF development team. Issues such as server maintenance and software upgrades can be done during the times the restaurant is closed in order to minimize the downtime of the system's functionality. This MTBF could be further minimized by using server redundancy solutions in order to be better prepared for situations such as hardware and power failures. Taking the previously mentioned issues into account, the MTBF of the system should be no less than 10 months.

Server load must also be taken into account, especially when multiple restaurants and customers are making use of the system at the same time. Load balancing techniques should be considered such as server side HTTP compression and HTTP caching, while on the client side caching and storage of static resources should be maximized in order to reduce requests made to the server.

4.1 Categorization of Bug Types

The bug type categorization style being used during the development of TFF is modeled after the categories available with the JIRA project management system. [2]

Categorization	Description	Example
Trivial	A trivial bug is categorized as an error that does not affect the functionality of TFF. Most errors in this category will have to do with the application's visual output.	Misspelled text.
Minor	A minor bug is categorized as an error that does not hamper the performance of TFF, and can be easily worked around.	Login link is broken from the main page but it works from a restaurant's page.
Major	A major bug is categorized as an error that will return incorrect or undesirable results. Most errors in this category will be a result of faults in program logic.	User can sign up but the details aren't properly saved into the database.
Critical	A critical bug is categorized as an error that will cause unrecoverable errors, performance issues, or render parts/all of TFF unusable. Most errors in this category will be a result of hardware crashes or corrupted data.	Submitting a form puts the application in an infinite loop state.
Blocker	A blocker bug is categorized as an error that will block development altogether.	Missing columns in the database preventing the proper models from being generated.

Table 4-1 Categorization of Bug Types

5 Performance

5.1 Response Time for Ordering Food

Time elapsed between the customer submitting an order and the order appearing on the restaurant's order list should not exceed 10 seconds.

5.2 Orders per Second

The system should be able to accommodate 2-3 orders per second for a brief time interval. For example, if a table of 12 decided to order all at the same time. For the brief 10-15 seconds of the table ordering, there could be multiple transactions per second.

5.3 Concurrent Orders

The system should be able to support up to 25 simultaneous orders being made from inside the restaurant. This situation seems unlikely, but large groups of people could collude to arrange a simultaneous order.

5.4 Customer Capacity

The system should be able to handle as many as 200 users browsing concurrently without a noticeable loss of performance.

5.5 Heavier Load Performance

When experiencing traffic heavier than expected (see previous requirements), the system should continue to function, but a slowdown of up to 50% should be expected.

5.6 Native App Hard Drive Space

The native application should not occupy more than 20MB of hard drive space on the user's phone.

5.7 Real Time Check-in

~~When a user checks in at the restaurant, the event should be recorded and displayed on the restaurant owner's side of the software in real time.~~

5.8 Throughput for Optimal Usage Mode

Under optimal performance conditions, the system should be able to transfer data based on the capabilities of the mobile device from which it is being used. Older iPhone 3GS can reach up to 7.2Mb/s wireless speed whereas the newer iPhone 5 on LTE could reach up to 50Mb/s on some service providers.

5.9 Sub Optimal Performance Mode

When the system fails to meet its target performance and becomes unusable, the restaurant should still be able to serve their clients using traditional pen and paper. Error messages should clearly indicate that the system may be too busy or is having issues with connection to avoid confusion.

6 Supportability

6.1 Coding Standards & Naming Conventions

The TFF application is being developed using Visual Studio 2010 IDE and will comply with "Microsoft All Rules" rule set. As such developers will be required to follow Microsoft's C# Programming Guide [3].

6.2 Maintenance Group Response Time

- All minor bugs/defects discovered will be addressed in a future iteration.
- All major bugs/defects will be addressed on a case by case basis but will be scheduled for an appropriate iteration.
- All critical or blocker bugs/defects that are a hindrance to development will be addressed within the current sprint.

Please see section 4.1 of this document for categorization of bug types.

7 Design Constraints

7.1 Mobile platform Development

The TFF customer application will run on any of the major mobile operating systems using their integrated web browsers. Additionally, a more feature-filled application will be written for Android, iOS and Windows Phone. The following table illustrates the platform and hardware specific features that will be used in each operating system.

Operating System/Feature	Network	Push Notifications	Storage	SD Card	Location Services	NFC Tag Reading	QR Code Reading	Camera
Android	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
iOS	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Windows Phone	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes

Table 7-1 Operating system features to be used

7.1.1 Java for Android Development

When developing a wrapper/version for the application on Android devices, the programming language enforced by this SDK is Java. The SDK is most compatible with Eclipse where the emulator can be launched. Alternatively, a physical Android device could be used to run and test the application deployed from the Eclipse project.

7.1.2 Objective-C for iPhone Development

When developing a wrapper/version for the application on iPhone devices, the programming language enforced by this SDK is Objective-C. Requirements for this SDK include having an operating system and the development kit from Apple (Xcode). Due to the price of this development kit, the team may not be able to accomplish this task unless a license is provided.

7.1.3 C# for Windows Phone Development

When developing a wrapper/version for the application on Windows Phone devices, the programming language enforced by this SDK is C#. The development kit installs and runs using Visual Studio, the same IDE being used for the web application side.

7.2 Web application Front End and Back End Development

7.2.1 HTML5/CSS3 with .Net for Web Application Front End

In order to create dynamic pages usable across all mobile devices targeted by the product, a web based application will be built using HTML5 with CSS3 and a specific portion of the .Net framework included in Visual Studio. The IDE includes various types of projects for the design and creation of web sites. The project type we will be using is the MVC3 Web Application, which makes use of the Razor viewing engine. This engine allows developers to use C# code directly within the HTML view in order to access useful functions and make ties with the controller in a much easier fashion.

7.2.2 C# with Entity Framework Data Access for the Back End

The Entity Framework provided by Visual Studio allows for the creation of base model classes which are based on the existing database design. These model classes can then be used to generate a basic controller skeleton class, which is then customized to provide the functionality required in that specific portion of the application.

7.3 Iterative Development Cycles (Agile)

For the type of project being undertaken, it is preferable to have an iterative development cycle. This type of approach allows for modified requirements and progress tracking, which suits the nature of the TFF application and the developer team's schedules. Most members of the developer team have experience using the agile methodology, which will help in avoiding scheduling pit falls.

Most of the requirements are not set in stone. There is a large chance that scope could change and that requirements fulfilled do not match exactly what was listed in the beginning of the project. This is not necessarily a bad thing, but it does make it rather difficult to use the waterfall and spiral models for the development process.

7.4 JIRA for Agile Development Issue Tracking and Management

In order to better accommodate the previous requirement (Agile), the team will use an online issue tracker called JIRA [2]. Team members will report bugs, assign tasks, log hours and track sprint progress with this tool.

7.5 SVN with Assembla Server for Version Control

Developers will require an Assembla account along with Tortoise SVN software to update, commit and merge revisions of the software source code [4]. This version control is crucial to the development cycle as it allows for many people to work on the same set of files as well as providing the capability to revert code to previous versions when needed.

7.6 Visual Studio Development

For the development of the web application component of the project, Visual Studio will be used since the backend will be in C#. Microsoft technologies are designed to be compatible with one another.

8 Online User Documentation and Help System Requirements

The user documentation for restaurant customers will be available online (when accessed via the mobile browser) as well as offline (when accessed via the mobile application itself). For restaurants, the user documentation will be available online, when accessed via a web browser and offline through a PDF file.

The customer's help system will give an overview of the features available on the mobile site as well as the features available on the app. The following help topics should be addressed in the help system:

- Scanning of NFC tags or QR codes
- Creation of a user profile
- Logging in/out of a user profile
- Placing an order
- Contacting a waiter/waitress
- Paying for an order
- Creating a food review

The restaurant's help system should give an overview of the features available on the web site. The following help topics should be addressed in the help system:

- Adding/removing/modifying tables
- Enabling/disabling use of tables
- Adding/removing/modifying menu items
- Enabling/disabling menu items for order
- Adding/removing a TFF user to the blacklist

9 Purchased Components

NFC tags will need to be purchased for testing. iOS developer licenses will also be needed. Since only two members of the team own a Mac, only two licenses will be required. These licenses are provided free for university students. A one-time purchase of JIRA and the JIRA Agile plugin will also be needed to manage the software development process.

10 Interfaces

10.1 User Interfaces

The core system application will be based on a web application that will run on popular existing browsers. There will also be wrappers to convert the web application to native applications for the iPhone, Android and Windows mobile devices. There will be two main user interfaces, one for the customer and one for the restaurant.

The customer user interface will consist of a section view described by the following:

- Profile management
- Menu viewing
- Order management
- Bill management
- ~~Table reservations~~
- Social networking section which will include user history, commenting, restaurant statistics and reviews.

The restaurant user interface will consist of:

- ~~Customer management~~
- Menu management
- Table management
- ~~Restaurant page management~~
- Order viewing and management
- Reporting and statistics

10.2 Hardware Interfaces

10.2.1 Display and Graphics

A mobile device is necessary to view the TFF customer application. The restaurant will require a screen to view the TFF restaurant application.

10.2.2 Input devices

Mobile device touch screen for the TFF customer application and a keyboard and mouse for the TFF restaurant application.

10.2.3 Database and storage

The TFF application will run with the ‘MSSQL’ database.

10.2.4 TFF System Server

The server hosting the application shall meet the following requirements:

- CPU: Intel Xeon
- RAM: Minimum - 1 GB, Recommended - 2GB
- Operating System: Windows Server
- Connection: Internet

10.2.5 TFF Customer Application

The mobile device using the application shall meet the following requirements:

- Camera or NFC enabled
- iPhone: 3gs and above
- Android: Running Android OS v2.3 and above
- Windows phone: Version 7.0 and above

10.2.6 Additional hardware

The mobile device will be using NFC tags or QR stickers to launch the TFF application on the customers’ mobile device.

10.3 Software Interfaces

The TFF system will be communicating with a ‘MSSQL’ database on the server to store and process information. The system will require the use of JavaScript on the client’s browser. ‘ASP.NET’ will also be used on the client side to display and process user operation. Native applications will also be created for the iPhone, Android and Windows mobile devices.

10.4 Communications Interfaces

The system shall use a data or internet connection to communicate with the TFF system server. It will also require NFC communication for NFC enabled mobile devices.

11 Licensing Requirements

11.1 Mobile Licenses

The application will require licensing from the Apple developer program for the iPhone native app. It will also require an Android market license for native Android applications and a Windows market license for native Windows applications.

11.2 Domain Name

The TFF system will require a domain name so that it is publicly accessible.

11.3 Copyright

The application name ‘Touch for Food’ will be copyrighted. TFF may not be distributed or modified without CloudNine’s consent.

11.4 Legal disclaimer

CloudNine is not responsible for allergic reactions or any food related illness.

Appendix A References

- [1] K. Anderson, C. Donato, J.Hum, M. Levkovsky, A. Lloyd, P. Modafferi, "F.S.T.S SRS," Concordia University, Montreal, SRS Document v6.16, 2012.
- [2] Atlassian. (2012, December) Issue & Project Tracking Software. [Online].
<http://www.atlassian.com/software/jira/overview>
- [3] Microsoft. (2012, December) C# Coding Conventions (C# Programming Guide). [Online]. <http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>
- [4] Assembla, Inc. (2012, Decemeber) Assembla; Home Page. [Online]. <https://www.assembla.com/home>

Appendix B Glossary

Term	Definition
Category	Placeholder for a set of foods fitting under the same grouping or page on the menu (i.e. Breakfast or Desert).
Client	A client of a restaurant who is or can be a potential user of the TFF app
Customer	A customer of a restaurant who is or can be a potential user of the TFF app
JIRA	The software process management system that CloudNine is using
Mobile Device	Any mobile phone or tablet that has a data connection and internet browser. It must also have NFC capability or the ability to scan QR codes.
QR Code	A matrix barcode that can be scanned using a smartphone camera
Smartphone	Any mobile phone or tablet that has a data connection and internet browser. It must also have NFC capability or the ability to scan QR codes.
Social media	A website that allows interactions between people. Could be Facebook, TFF profile, or any other social networking site.
User	Anyone who uses the TFF app. Can be a customer or the restaurant itself.

Appendix C Acronyms

Acronyms	Meaning
IDE	Integrated development environment
NFC	Near Field Communication
QR	Quick response
SDK	Software development kit
SVN	Subversion
TFF	Touch For Food

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Software Architecture & Design

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	Introduction.....	7
1.1	Purpose	7
1.2	Scope.....	7
1.3	Definitions, Acronyms and Abbreviations	7
1.4	References.....	7
1.5	Overview.....	7
2	Architectural Representation.....	7
2.1	Use Case View.....	8
2.2	Logical View.....	8
2.3	Process View	8
2.4	Physical/Deployment View	8
3	Architectural Goals and Constraints	9
3.1	MVC vs. 3-Tier Architecture.....	9
4	Use Case View	10
4.1	Actor Goal List	10
4.2	Client Use Case Model	11
4.3	Fully Dressed Use Cases.....	12
5	Logical View.....	25
5.1	Domain Model	25
5.2	Class Diagram.....	26
5.3	ERD Diagram	29
5.4	Sequence Diagrams.....	30
5.5	Communication Diagrams	37
5.6	System Sequence Diagrams.....	39
6	Process View.....	42
6.1	Activity Diagrams.....	42
6.2	State Diagrams	48
7	Physical View	50
7.1	Deployment Diagram.....	50
8	Size & Performance	51
9	Quality.....	51
Appendix A	References.....	52
Appendix B	Prototypes	53

List of Figures

Figure 3-1 Diagram for a typical MVVM Pattern that goes wits with MVC [1].....	9
Figure 4-1 Client Use Case Model	11
Figure 4-2 Restaurant Use Case Model	11
Figure 5-1 Domain Model	25
Figure 5-2 Class Diagram	27
Figure 5-3 Class Diagram Part 2.....	28
Figure 5-4 ERD Diagram.....	29
Figure 5-5 Populate Menu Sequence Diagram	30
Figure 5-6 Call Waiter Sequence Diagram.....	31
Figure 5-7 Log In Sequence Diagram.....	32
Figure 5-8 Edit Category Concurrency Sequence Diagram.....	33
Figure 5-9 Edit Order Item Concurrency Sequence Diagram.....	34
Figure 5-10 Menu Proxy List Sequence Diagram	35
Figure 5-11 Review Sequence Diagram	36
Figure 5-12 Collaboration Diagram for populating a menu from scratch	37
Figure 5-13Collaboration Diagram for populating menu with pre-existing categories and items.....	37
Figure 5-14 Communication Diagram for Deleting a Menu Item	38
Figure 5-15 System Sequence Diagram for Place Order	39
Figure 5-16 System Sequence Diagram for Topping/Sides Additions	40
Figure 5-17 System Sequence Diagram for Create Menu	41
Figure 6-1 View Menu.....	42
Figure 6-2 Add Food Item to Order.....	43
Figure 6-3 Remove Food Item from Order.....	44
Figure 6-4 Logging In.....	45
Figure 6-5 Placing an Order.....	46
Figure 6-6 Processing an Order	47
Figure 6-7.1 State Diagram for Order Food.....	48
Figure 6-8 Call Waiter State Diagram	49
Figure 7-1 Deployment Diagram.....	50
Figure B-1 Manage Order Prototype	53
Figure B-2 Menu Editor Main Page	54
Figure B-3 Category Management Page.....	54
Figure B-4 Item Management Page	55
Figure B-5 Order From Menu.....	56

List of Tables

Table 2-1 Diagram Types Vs. Views.....	8
Table 4-1 Actor Goal List.....	10

Touch For Food

Software Architecture & Design

Version 7.47

Revision History

Date	Rev.	Description	Author(s)
2012-09-17	0.00	Document Creation	Katrina Anderson
2012-09-26	0.01	Contributed to Section 3 Architectural Goals and Constraints	Mikhail Levkovsky
2012-09-26	0.02	Contributed to Section 10 Quality	Mikhail Levkovsky
2012-09-28	0.03	Contributed to Section 1 and 2	Patrick Modafferi
2012-09-29	0.04	Contributed to Section 8 (Physical View) and 9 (Size & Performance)	Cristian Asenjo
2012-09-30	0.05	Contributed to Section 4 (Actor Goal List, Fully Dressed UC2.1 – UC2.4, Use Case Model)	Cynthia Donato
2012-09-30	0.06	Contributed to Section 4 UC1.1 Formatted section. Added Table of Figures.	Katrina Anderson
2012-10-01	0.07	Contributed to Section 5.2 and 5.3. added the figures to Table of Figures	Christian Daher/ Ryan Nasr
2012-10-14	0.08	Contributed to Section 7.1	Matthew Tam
2012-10-20	1.09	Updated to sprint 1Contributed section 4.3	Patrick Modafferi/ Mikhail Levkovsky
2012-10-26	1.10	Added Sequence Diagrams Populate Menu and updated actor goal list	Mikhail Levkovsky
2012-10-27	1.11	Added Collaboration Diagrams Populate Menu	Patrick Modafferi
2012-10-31	1.12	Added Use Case for updating customer profile. Fixed numbering for use case 3.1	Cynthia Donato
2012-11-03	2.13	Added Use Case for creating a user profile. Renumbered section 4.3 to have the user profile creation first and the user profile updating second.	Cristian Asenjo
2012-11-03	2.14	Added Use Case for login to personal profile. Renumbered section 4.3 to have the user login first and the user profile updating second	Matthew Tam
2012-11-04	2.15	Reviewed and made minor corrections	Josh Hum
2012-11-18	2.16	Added Use Case 4.4	Katrina Anderson
2012-12-16	3.17	Added clarification for load balancing solution.	Cristian Asenjo
2012-12-19	4.18	Removed outdated use cases, updated use case models, included use case 2.1	Katrina Anderson
2012-12-19	4.19	Added use case 2.2	Cynthia Donato
2012-12-20	4.20	Updated class diagram & domain model	Josh Hum & Matthew Tam
2012-12-21	4.21	Reviewed document and reduced redundancies	Katrina Anderson
2013-01-10	5.22	Added Appendix B Prototypes	Katrina Anderson
2013-01-11	5.23	Added UC5.1, Also updated the Use case diagrams	Christian Daher/ Ryan Nasr
2013-01-14	5.24	Updated Section 3	Mikhail Levkovsky
2013-01-16	5.25	Added Prototypes to the Appendix	Patrick Modafferi

2013-01-18	6.26	Added UC7.1 Call a waiter, updated actor goal list, updated use case models and fixed numbering for figures in appendix B.	Katrina Anderson
2013-01-27	6.27	Added UC 6.1	Patrick Modafferi
2013-01-23	6.28	Added UC 8.1 Manage Bill, updated use case models, updated actor goal list.	Christian Daher/ Ryan Nasr
2013-01-26	6.29	Adjusted UC7.1	Katrina Anderson
2013-01-27	6.30	Added sequence diagram 5.3.2 and state diagram 6.4.2	Cynthia Donato
2013-01-27	6.31	Review and update sections	Patrick Modafferi
2013-01-28	6.32	Added logging in activity diagram	Cristian Asenjo
2013-01-28	6.33	Added logging in sequence diagram	Cristian Asenjo
2013-01-28	6.34	Added Review prototype	Patrick Modafferi
2013-02-02	7.35	Added Communication Diagram and Sequence Diagram	Mikhail Levkovsky
2013-02-02	7.36	Added ERD diagram	Mikhail Levkovsky
2013-02-02	7.37	Updated UC2.2	Patrick Modafferi
2013-02-04	7.38	Added Sequence Diagram	Mikhail Levkovsky
2013-02-07	7.39	Added a diagram for MVVM	Patrick Modafferi
2013-02-08	7.40	Updated Class diagram	Christian Daher
2013-02-10	7.41	Added Activity Diagram 6-5	Matthew Tam
2013-02-11	7.42	Added Sequence diagrams 5-6 and 5-7	Cynthia Donato
2013-02-12	7.43	Added System Sequence diagram – Create Menu	Josh Hum
2013-02-12	7.44	Added Placing Order activity diagram	Cristian Asenjo
2013-02-12	7.45	Updated Domain Model	Cynthia Donato
2013-02-12	7.46	Reviewed SAD and added Create Review Prototype	Mikhail Levkovsky
2013-02-12	7.47	Reviewed SAD for formatting mistakes	Josh Hum

1 Introduction

1.1 Purpose

The purpose of this document is determining, on an architectural level, how each part of the TFF application will work. These architectural diagrams can represent inner workings of the system or higher level design. When it comes to implementing, developers should refer to this document to save time and maintain consistency.

There are many types of diagrams some help determine the flow of the system and some explain how components interact with one another. The domain model and class diagram shows how objects are related. Finally, some diagrams show more physical elements like the deployment diagram.

Another reason for this documentation is scalability, maintainability and reusability. In order to allow for someone to potentially expand on this project, reuse it for something else or maintain the finished product, the documentation must be able to serve as a road map or a guide for any future developer.

1.2 Scope

The scope of the architecture artifacts cover the main components of the system. Each iteration, the components planned for that sprint, will be designed. The resulting architecture and design diagrams will be added to the document. In this way, the document will incrementally be built up to a final document containing diagrams for every component in the system.

1.3 Definitions, Acronyms and Abbreviations

Refer to the SRS document - Appendix B Glossary and Appendix C Acronyms for a complete list of terms and definitions.

1.4 References

Please see Appendix A, References, of this document.

1.5 Overview

The document is organized into five main views for architecture; Use Case, Logical, Development, Process and Physical (See section 2 for more details). As needed, other items will be discussed such as Quality, Size and Performance. This document will also discuss architecture styles and constraints as well as a comparative analysis to justify some decisions that were taken. As a whole, the document should completely represent all the architecture of the system.

2 Architectural Representation

Section 3 of this document provides a summary of the architecture used for the construction of TFF. The remaining sections represent the 4+1 views of the TFF system. This section of the document summarises what architectural artefacts can be found in each view.

Table 2-1 Diagram Types Vs. Views

Use Case	Logical	Process	Physical
Actor-Goal List	Operational Contracts	Activity Diagrams	Deployment Diagram
Use Case Diagrams	Domain Model	State Diagrams	
Use Cases	Class Diagram		
	System Sequence Diagrams		
	Sequence Diagrams		
	Communication Diagrams		

2.1 Use Case View

The use case view represents important requirements through fully detailed use cases. Use case designations are also represented through actor goal lists and use case models.

2.2 Logical View

The logical view determines how the layers, packages, classes and other software elements are organized in the system.

2.3 Process View

The process view illustrates processes and threads in the system. Could show how some elements interact and collaborate throughout a process.

2.4 Physical/Deployment View

The physical view represents the components, processes, communications and important structures of the system.

3 Architectural Goals and Constraints

TFF will be designed using an MVC architecture. The reason for choosing a MVC architecture is to achieve:

- Decoupling of presentation, data and domain logic
- Allows for multiple people to work on different parts of the same project
- Promotes low coupling between different components
- Promotes organization and code reuse
- Promotes consistent and well defined interfaces between each layer

Well-structured MVC architecture will also allow us to swap out or update any component independently of the others as long as the interfaces are respected. The View Model Model architecture will also be employed, which is an extra component to the classic MVC architecture.

MVC consists of:

- Model – The models will be database driven, that will be used to represent the state of the application
- View – Determines how to present the model data to the user
- Controller – Passes the information between the user requests and the model, and vice-versa.
- View Model – Will be used when extra logic is required for the view to properly render. (see Figure 3-1)

Some constraints of the MVC architecture are:

- Tracing end-to-end flows of data can be challenging as the system grows

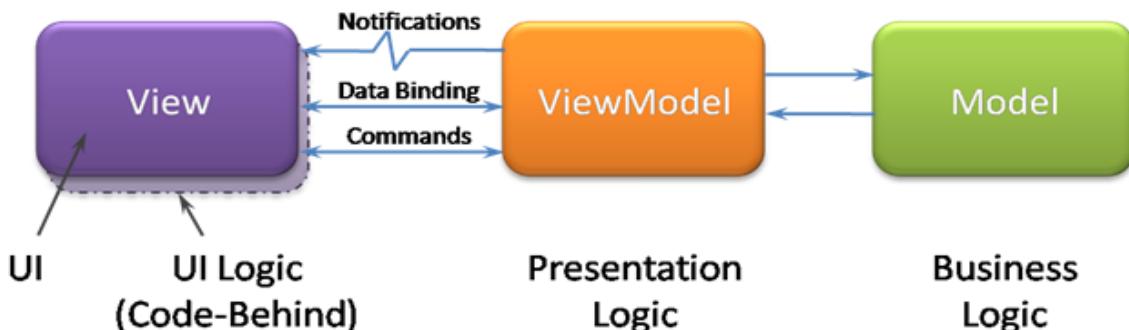


Figure 3-1 Diagram for a typical MVVM Pattern that goes wits with MVC [1]

3.1 MVC vs. 3-Tier Architecture

Since 3-Tier Architecture is used for large scale enterprise solutions which require physical separation, MVC has been chosen to develop TFF. MVC allows us to represent each component of the system as its own encapsulated part. If needed, MVC can be converted to a 3-Tier Architecture.

4 Use Case View

4.1 Actor Goal List

Table 4-1 Actor Goal List

Actor	Goal
User	<ol style="list-style-type: none">1. View Menu2. Place Order3. Deliver Order Item4. Create User Profile5. Login to Personal Profile6. Update Customer Setting7. Lookup Customer Restaurant and Order History8. Submit Review9. Call A Waiter
Restaurant	<ol style="list-style-type: none">1. Populate Menu2. Manage Tables3. Call A Waiter4. Manage Bills

4.2 Client Use Case Model

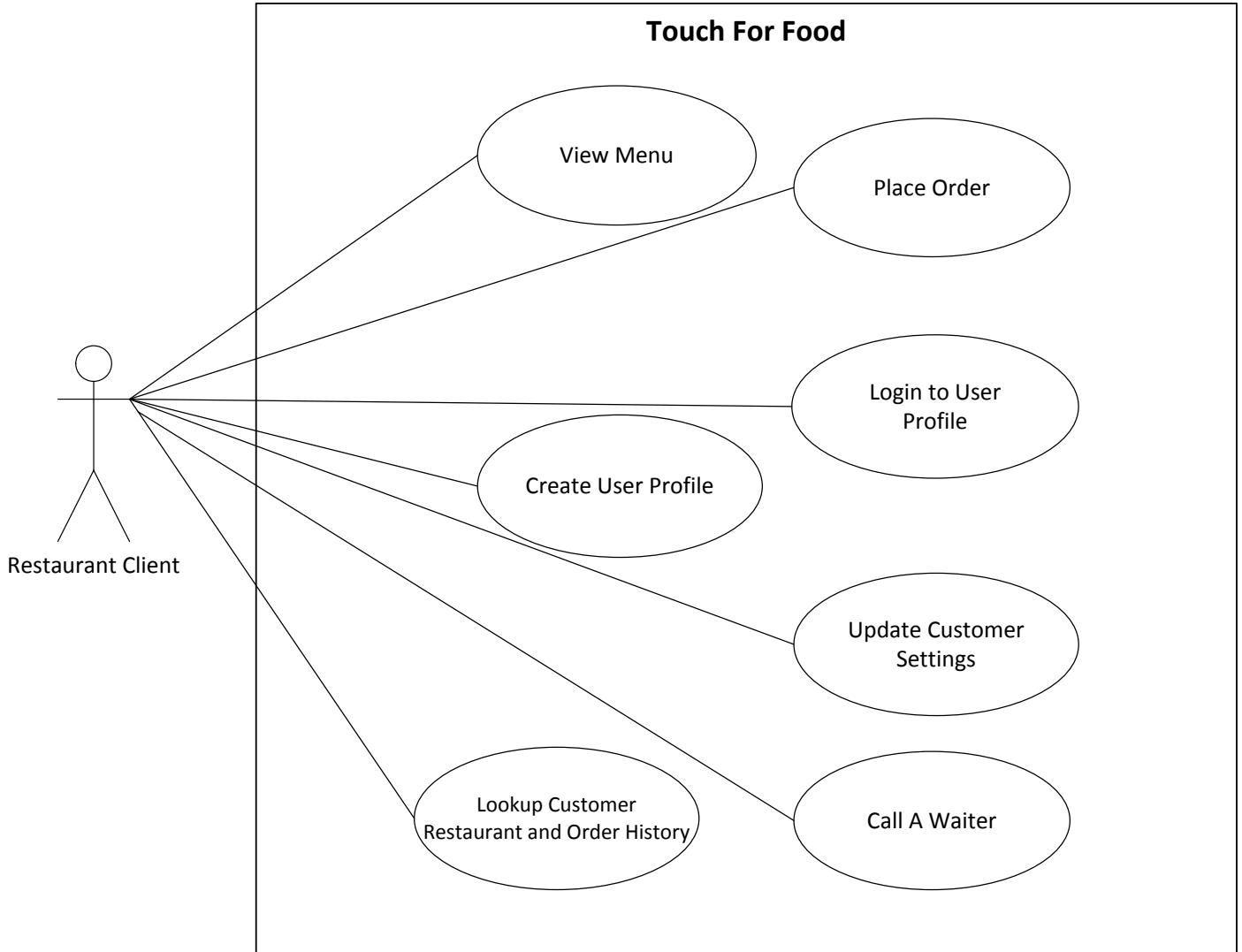


Figure 4-1 Client Use Case Model

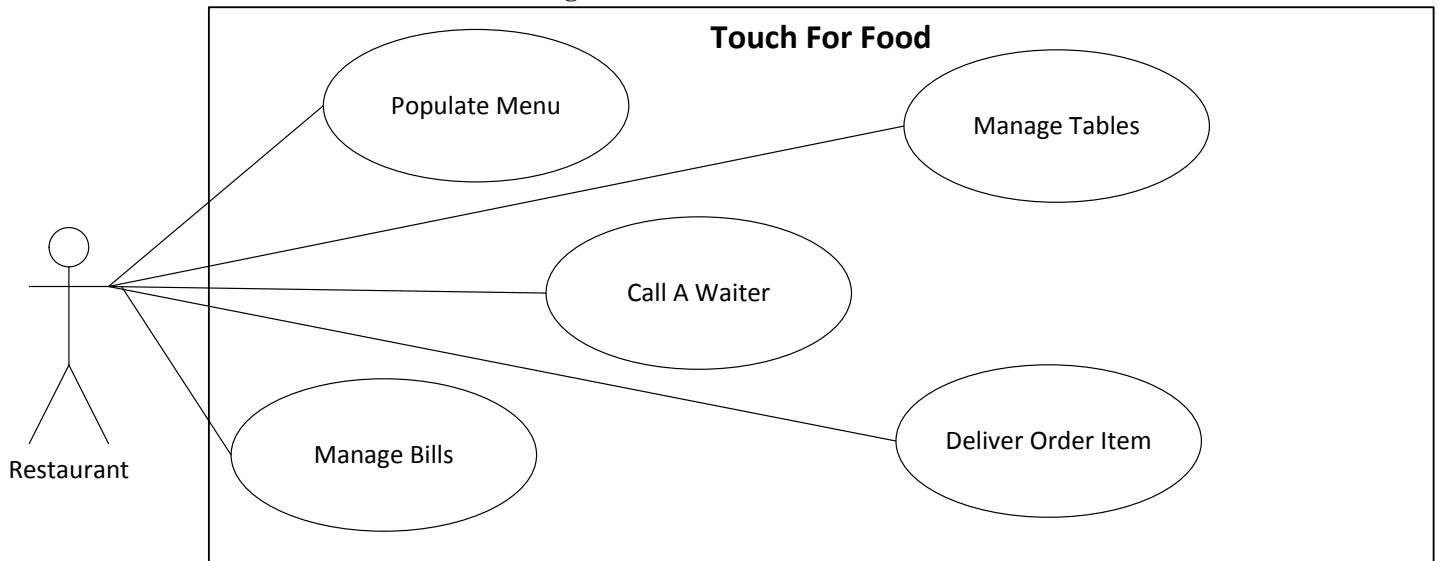


Figure 4-2 Restaurant Use Case Model

4.3 Fully Dressed Use Cases

4.3.1 UC1.1 View Menu

ID: UC1.1

Use Case: View Menu

Description: Restaurant client wishes to view all menu options.

Level: User Goal

Primary Actor: User

Supporting Actors: None

Stakeholders and Interests:

User: Their interest is to view their meal and beverage options for a restaurant.

Restaurant: Their interest is for clients to see meal and beverage options.

Pre-Conditions:

1. The user is logged into their Touch for Food account.
2. The user has scanned the NFC tag or QR code at the given restaurant.

Post Conditions:

Success end condition:

1. All menu options are displayed

Failure end condition:

1. No menu options are displayed
2. An error message is displayed

Minimal Guarantee:

1. An informative message is displayed to the user.

Main Success Scenario:

1. User initiates restaurant page search
2. System returns restaurant page
3. User navigates to the menu section of the restaurant page.
4. System displays restaurant menu in order of course.

Extensions:

4.a. Client chooses to sort menu by price

- 4.a.1 User selects to have menu items appear in order of price.
- 4.a.2 System displays restaurant menu items in order of price

4.b Client chooses to sort menu by popularity

- 4.b.1 User selects to have menu items appear in order of popularity.
- 4.b.2 System displays restaurant menu items in order of popularity

Special Requirements:

See Section 2 Functionalityof Supplementary Specifications document

See Section 3 Usabilityof Supplementary Specifications document

See Section 4 Reliabilityof Supplementary Specifications document

See Section 5 Performanceof Supplementary Specifications document

See Section 6 Supportabilityof Supplementary Specifications document

See Section 7 Design Constraintsof Supplementary Specifications document

See Section 10 Interfacesof Supplementary Specifications document

4.3.2 UC2.1 Place Order

ID:UC2.1

Use Case:Place Food Order

Description: User wishes to add, remove and modify menu items to create and place an order with the restaurant.

Level:User-goal

Primary Actor:User

Secondary Actor: None

Stakeholders and Interests:

User: Their interest is to place an accurate order quickly and effortlessly.

Restaurant: Their interest is to receive an accurate and detailed food order from the customer.

Pre-Conditions:

1. The user is physically present at a restaurant.
2. The user is registered with Touch for Food.
3. The user is logged into their Touch for Food account.
4. The user has scanned the NFC tag or QR code at the given restaurant.

Post Conditions:

Success end condition

1. The food order is conveyed to the restaurant staff.

Failure end condition:

1. The order remains unchanged.
2. The user is notified that the system failed to convey the order to the restaurant staff.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. User indicates that they want to place an order.
2. System displays the restaurant's menu.
3. User selects food items and indicates that they would like to add them to their order.
4. System requests that the user provide special requests or notes.
5. User specifies special requests or notes.
6. System notifies the user that the items have been added to their order.
7. User finalises and places order.
8. System notifies the user that the order has been placed successfully.

Extensions:

8.a The user indicates that they would like to remove item from order

- 8.a.1 System displays the user's order.
- 8.a.2 User selects a food item and indicated that they would like to remove it from their order.
- 8.a.3 System requests that the user confirms the removal of the item.
- 8.a.4 User confirms they would like to remove the item.
- 8.a.5 System notifies the user that the item has been deleted.
- 8.a.6 User finalises and places order.
- 8.a.7 System notifies the user that the order has been placed successfully.

8.b The user indicates that they would like to remove item from order

- 8.b.1 System times out and does not change order (user did not enter changes for 5 minutes)

8.c The user indicates that they would like to add item to order

- 8.c.1 System displays the user's order.
- 8.c.2 User selects a food item and indicated that they would like to add it to their order.
- 8.c.3 System requests that the user provide special requests or notes.
- 8.c.4 User specifies special requests or notes.
- 8.c.5 System notifies the user that the item has been added to their order.
- 8.c.6 User finalises and places order.
- 8.c.7 System notifies the user that the order has been placed successfully.

8.d The user indicates that they would like to add item to order

- 8.d.1 System times out and does not change order (user did not enter changes for 5 minutes).

Special Requirements:

- See Section 2 Functionalityof Supplementary Specifications document
- See Section 3 Usabilityof Supplementary Specifications document
- See Section 4 Reliabilityof Supplementary Specifications document
- See Section 5 Performanceof Supplementary Specifications document
- See Section 6 Supportabilityof Supplementary Specifications document
- See Section 7 Design Constraintsof Supplementary Specifications document
- See Section 10 Interfaceeof Supplementary Specifications document

4.3.3 UC2.2 Deliver Order Item

ID:UC2.2

Use Case: Deliver Order Item

Description:The restaurant wishes to process individual order items related to an order.

Level:User-goal

Primary Actor:Restaurant personnel

Secondary Actor: Restaurant client

Stakeholders and Interests:

Restaurant Personnel: Their interest is to be sure that orders are handled in a timely fashion.

User: Their interest is to have all their order processed correctly by the restaurant.

Pre-Conditions:

1. A member of the restaurant personnel is logged into the restaurant's Touch for Food account.
2. At least one order has been placed for the given restaurant.
3. The placed order has at least one order item.

Post Conditions:

Success end condition:

1. The order item has a status of "Done" indicating it is ready for delivery to a table.

Failure end condition:

1. The order item has a status of "Rejected".
2. The user is notified that the ordered menu item is no longer available.

Minimal Guarantee:

1. The order item status is changed from "Pending".

Main Success Scenario:

1. The restaurant personnel indicates that they would like to view the currently placed orders
2. The system displays the tables with users currently tapped in
3. The restaurant personnel selects a table for which they would like to see the orders
4. The system displays all pending orders for that table along with order items associated
5. The restaurant personnel indicates that they would like to accept a given order item.
6. The system displays the order item as having a status of "Processing".
7. The restaurant personnel indicates that they would like to set the order to "Ready".
8. The system displays the order item as having a status of "Done".

Extentions:

5.a The restaurant personnel indicates that they would like to decline the order item.

- 5.a.1 The system asks the restaurant personnel to indicate if the declined menu item should be removed from the menu.
- 5.a.2 The restaurant personnel indicates their choice.
- 5.a.3 The user is notified that the order item is no longer available.

7.a The restaurant personnel indicates that they would like to cancel the order item.

- 7.a.1 The system asks the user to indicate if the declined menu item should be removed from the menu.
- 7.a.2 The restaurant personnel indicates their choice.
- 7.a.3 The user is notified that the order item is no longer available.

Special Requirements:

See Section 2 Functionalityof Supplementary Specifications document

See Section 3 Usabilityof Supplementary Specifications document

See Section 4 Reliabilityof Supplementary Specifications document

See Section 5 Performanceof Supplementary Specifications document

See Section 6 Supportabilityof Supplementary Specifications document

See Section 7 Design Constraintsof Supplementary Specifications document

See Section 10 Interfacesof Supplementary Specifications document

4.3.4 UC3.1 Populate Menu

ID:UC3.1

Use Case:Populate Menu

Description:User wishes to add categories and items to a menu.

Level:User-goal

Primary Actor:Client

Secondary Actor: None

Stakeholders and Interests:

User: Their interest is to be able to see what items are available for them to order

Restaurant: Their interest is to be able to create fully customized menus

Pre-Conditions:

1. The client is logged in to their TFF account
2. The system contains at least one menu with a default category
3. The client has navigated to the menu editor
4. The client has selected the option to create/edit a menu

Post Conditions:

Success end condition

1. The system saves the menu with at least one category and one menu item

Failure end condition

1. The existing draft order remains unchanged.
2. The user is notified that the system failed and to contact support.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. The system takes the client to the selected menu page that contains at least the default category.
2. Client adds one or more categories to the menu.
3. The system displays the menu category list.
4. Client chooses a category they would like to add items to
5. The system displays a list of existing menu items in the selected category.
6. Client adds one or more items to the menu's category.
7. The system displays items in category list.
8. Client selects to finalize.
9. The system saves the menu with all modifications.

Extensions:

2.a The client chooses to create a new category

- 3.a.1 The system displays a category creation form.
- 3.a.2 The client enters a name for the category and confirms the creation.
- 3.a.3 The system saves the category and adds it to the menu.

2.b The client selects a category from a list of previously created categories

- 3.a.1 The system adds the selected category to the menu.

5.a The client chooses to create a new menu item

- 3.a.1 The system displays a menu item creation form.
- 3.a.2 The client enters a name, description and price for the menu item and confirms the creation.
- 3.a.3 The system saves the category and adds it to the menu.

5.b The client selects a menu item from a list of previously created ones

- 3.a.1 The system adds the selected category to the menu.

Special Requirements:

See Section 2 Functionality of Supplementary Specifications document

See Section 3 Usability of Supplementary Specifications document

See Section 5 Performance of Supplementary Specifications document

See Section 7 Design Constraints of Supplementary Specifications document

4.3.5 UC4.1 Create User Profile

ID:UC4.1

Use Case:Create User Profile

Description:User wishes to create a user profile.

Level:User-goal

Primary Actor:User

Secondary Actor: None

Stakeholders and Interests:

User: Their interest is to create a user profile.

Pre-Conditions:

1. The user is at the Create User Profile page.

Post Conditions:

Success end condition

1. The user profile is created correctly.

Failure end condition:

1. The user profile is not created.
2. The user is notified that the system failed to create a new user profile.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. User navigated to the Create User Profile section.
2. System displays an empty user information form.
3. User fills out the user information form and indicates it would like to confirm the creation of the profile.
4. System notifies the user that the profile was created.

Extensions:

3.a The user indicates that they would like to cancel without creating the profile

- 3.a.1 System redirects to the main Touch For Food page.

Special Requirements:

See Section 2 Functionalityof Supplementary Specifications document

See Section 3 Usabilityof Supplementary Specifications document

See Section 4 Reliabilityof Supplementary Specifications document

See Section 5 Performanceof Supplementary Specifications document

See Section 6 Supportabilityof Supplementary Specifications document

See Section 7 Design Constraintsof Supplementary Specifications document

See Section 10 Interfacesof Supplementary Specifications document

4.3.6 UC4.2 Login to Personal Profile

ID:UC4.2

Use Case:Login to Personal Profile

Description:User wishes to log in to their personal profile.

Level:User-goal

Primary Actor:Restaurent

Secondary Actor: Restaurent client

Stakeholders and Interests:

User: Their interest is to view their personal profile.

Pre-Conditions:

1. The user is at the User Login page.

Post Conditions:

Success end condition

1. The user's personal profile page is displayed.

Failure end condition:

1. The user's personal profile is not displayed.
2. The user is notified that the system failed to display their personal profile.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. User navigates to the User Login section.
2. System displays two text fields for username and password.
3. User inputs their username and password and attempts to login.
4. System verifies the username and password.
5. User views their personal profile.

Extensions:

3.a The user inputs the wrong username and/or password

- 3.a.1. System verifies the username and password and redirects user to the user login page with an error message:
“Username or Password is incorrect, please enter your credentials again”.

3.b The user forgets their password and clicks on the “Forgot Password” link

- 3.b.2. System redirects user to the Forgot Password section and displays options to recover their password

3.c The user wants to create an account

- 3.c.1. Refer to UC4.1 Create User Profile

Special Requirements:

See Section 2 Functionalityof Supplementary Specifications document

See Section 3 Usabilityof Supplementary Specifications document

See Section 4 Reliabilityof Supplementary Specifications document

See Section 5 Performanceof Supplementary Specifications document

See Section 6 Supportabilityof Supplementary Specifications document

See Section 7 Design Constraintsof Supplementary Specifications document

See Section 10 Interfacesof Supplementary Specifications document

4.3.7 UC4.3 Update Customer Settings

ID:UC4.3

Use Case:Update Customer Settings

Description:User wishes to update their personal profile.

Level:User-goal

Primary Actor:User

Secondary Actor: None

Stakeholders and Interests:

User: Their interest is to update their personal profile.

Pre-Conditions:

1. The user is logged into their Touch for Food account.

Post Conditions:

Success end condition

1. The user profile is update correctly.

Failure end condition:

1. The account user account information remains unchanged.
2. The user is notified that the system failed to update their profile.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. User navigated to the Edit Profile section.
2. System displays an editable version of the user profile.
3. User makes necessary updated to name, email and/or display name and indicates that they would like ot save the changes.
4. System notifies the user that the profile has been updated.

Extensions:

3.a The user indicates that they would like to cancel without saving

- 3.a.1 System redirects to the user profile page.

Special Requirements:

See Section 2 Functionalityof Supplementary Specifications document

See Section 3 Usabilityof Supplementary Specifications document

See Section 4 Reliabilityof Supplementary Specifications document

See Section 5 Performanceof Supplementary Specifications document

See Section 6 Supportabilityof Supplementary Specifications document

See Section 7 Design Constraintsof Supplementary Specifications document

See Section 10 Interfacesof Supplementary Specifications document

4.3.8 UC4.4 Lookup Customer Restaurant and Order History

ID:UC4.4

Use Case:Lookup Customer Restaurant and Order History

Description:User wishes to look up restaurants they visited and what they ordered.

Level:User-goal

Primary Actor:User

Secondary Actor: None

Stakeholders and Interests:

User: Their interest is to review history of events.

Pre-Conditions:

1. The user is logged into their Touch for Food account (UC4.2).

Post Conditions:

Success end condition

1. Order details are displayed.

Failure end condition:

1. Order details are not displayed.
2. The user is notified that the system failed to lookup order details.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. User navigated to the order history section of their profile page.
2. System displays a list of restaurants the user has visited.
3. User selects the restaurant he wants order details from.
4. System displays a list of orders made by the user at that restaurant.
5. User selects the order he wants details from.
6. System displays order details.

Special Requirements:

See Section 2 Functionalityof Supplementary Specifications document

See Section 3 Usabilityof Supplementary Specifications document

See Section 4 Reliabilityof Supplementary Specifications document

See Section 5 Performanceof Supplementary Specifications document

See Section 6 Supportabilityof Supplementary Specifications document

See Section 7 Design Constraintsof Supplementary Specifications document

See Section 10 Interfaceeof Supplementary Specifications document

4.3.9 UC5.1 Manage Tables

ID:UC5.1

Use Case:Manage Tables

Description:Restaurant would like to be able to manage their tables.

Level:User-goal

Primary Actor:Restaurant

Secondary Actor: Restaurant Client

Stakeholders and Interests:

Restaurant Client: Their interest is to be able to manage tables.

Restaurant: Their interest is to be able to use the NFC/QR code to view the restaurant's menu.

Pre-Conditions:

1. The Restaurant is logged into their restaurant admin Touch for Food account (UC4.2).

Post Conditions:

Success end condition

1. Tables were organized as per the Restaurant's desire.

Failure end condition:

1. Tables were not created, modified, or deleted as per the Restaurant's desire.
2. The Restaurant is notified of inability to organize tables.
3. The Restaurant client is unable to view the menu for the restaurant they are currently at.

Minimal Guarantee

1. An informative message is displayed to the Restaurant.
2. Restaurant Client is asked if they would like to call a waiter over for help.

Main Success Scenario:

1. Restaurant user navigates to the table management page.
2. System displays a list of tables the restaurant currently has including their respective URL links.
3. Restaurant user Creates new tables as desired.
4. System displays the new list of tables with their respective URL links.
5. Restaurant user writes the URL link that were newly generated to the NFC, or orders them from TouchForFood.
6. Restaurant receives the NFC tags/QR code shipments and sticks them to the new tables.
7. Restaurant Client uses the NFC/QR code to quickly view the menu of the restaurant.
8. System displays the currently active menu of the current restaurant.

Extensions:

3.a. Client chooses to sort menu by price

- 3.a.1 Restaurant user modifies or deletes a table.
- 3.a.2 System displays the new list of tables with their respective URL links.

Special Requirements:

- See Section 2 Functionalityof Supplementary Specifications document
- See Section 3 Usabilityof Supplementary Specifications document
- See Section 4 Reliabilityof Supplementary Specifications document
- See Section 5 Performanceof Supplementary Specifications document
- See Section 6 Supportabilityof Supplementary Specifications document
- See Section 7 Design Constraintsof Supplementary Specifications document
- See Section 10 Interfacesof Supplementary Specifications document

4.3.10 UC6.1 Submit a Review

ID: UC6.1

Use Case: Submit a Review

Description: User wishes to leave a review for what was ordered during the dining experience

Level: User-goal

Primary Actor: Restaurant Client

Secondary Actor: Restaurant, System

Stakeholders and Interests:

User: Their interest is to submit an accurate review of what was served to them by the restaurant.

Restaurant: Their interest is to acquire valuable information about user preferences.

Pre-Conditions:

1. The user has placed an order (see UC2.1)
2. The restaurant has delivered the order to the user
3. The user consumes the contents of the order

Post Conditions:

Success end condition

1. The review is posted and accurately reflects the user's experience.

Failure end condition:

1. The review is not stored.
2. The user is notified that the system failed.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. User indicates that they want to create a review.
2. System displays the order for which the user can submit a review on.
3. User enters a rating and comments related to the order selected.
4. User submits the review.
5. System warns the user that the review is final.
6. User confirms the decision.
7. System stores the review and associates it to the correct user, order and restaurant.

Extensions:

N/A

Special Requirements:

See Section 2 Functionality of Supplementary Specifications document

See Section 3 Usability of Supplementary Specifications document

See Section 5 Performance of Supplementary Specifications document

See Section 7 Design Constraints of Supplementary Specifications document

See Section 10 Interfaces of Supplementary Specifications document

See Section 11 Licensing Requirements of Supplementary Specifications document

4.3.11 UC7.1 Call A Waiter

ID: UC7.1

Use Case: Call A Waiter

Description: Client wishes to request service from a waiter and the restaurant wishes to know when their services are needed.

Level: User-goal

Primary Actor: Restaurant Client

Secondary Actor: Restaurant

Stakeholders and Interests:

Restaurant Client: Their interest is to receive service from restaurant wait staff.

Restaurant: Their interest is to be able to serve their clientele.

Pre-Conditions:

1. The client is logged in to their TFF account and assigned to a table.

Post Conditions:

Success end condition

1. The system indicates that the client has been served.

Failure end condition

1. The user is notified that the system failed to send or update the request for service.

Minimal Guarantee

1. An informative message is displayed to the user.

Main Success Scenario:

1. The client sends a request for service from the restaurant wait staff.
2. The system displays a confirmation screen for the service call along with an optional field to convey a message to the restaurant wait staff.
3. The client confirms the request for service.
4. The system notifies the restaurant of the request for service.
5. The restaurant indicates that they have fulfilled the request for service.
6. The system updates and displays the status of the request for service to fulfilled.

Extensions: N/A

Special Requirements:

See Section 2 Functionality of Supplementary Specifications document

See Section 3 Usability of Supplementary Specifications document

See Section 5 Performance of Supplementary Specifications document

See Section 7 Design Constraints of Supplementary Specifications document

See Section 10 Interfaces of Supplementary Specifications document

See Section 11 Licensing Requirements of Supplementary Specifications document

4.3.12 UC8.1 Manage Bill

ID: UC8.1

Use Case: Manage A Bill

Description: Restaurant wishes to split the payment of a single order into different bills.

Level: User-goal

Primary Actor: Restaurant

Secondary Actor: Restaurant Client

Stakeholders and Interests:

Restaurant Client: Their interest is to be able to pay only for their own food.

Restaurant: Their interest is to be able to split the payment of an order into different bills.

Pre-Conditions:

1. The restaurant is logged in to their TFF admin account and is viewing an order that has been delivered.

Post Conditions:

Success end condition

1. The system splits the order payment into separate bills.

Failure end condition

1. The system is unable to split the order as the restaurant client desires.

Minimal Guarantee

2. An informative message is displayed to the user.

Main Success Scenario:

7. The restaurant client asks the waiter to split the order into separate bills.
8. Waiter creates and manages bills according to the needs of the client.
9. The system provides an overview of the bill organization that was created.
10. The waiter prints out the bills and hands them to the client(s).

Extensions:

Special Requirements:

See Section 2 Functionality of Supplementary Specifications document

See Section 3 Usability of Supplementary Specifications document

See Section 5 Performance of Supplementary Specifications document

See Section 7 Design Constraints of Supplementary Specifications document

See Section 10 Interfaces of Supplementary Specifications document

See Section 11 Licensing Requirements of Supplementary Specifications document

5 Logical View

5.1 Domain Model

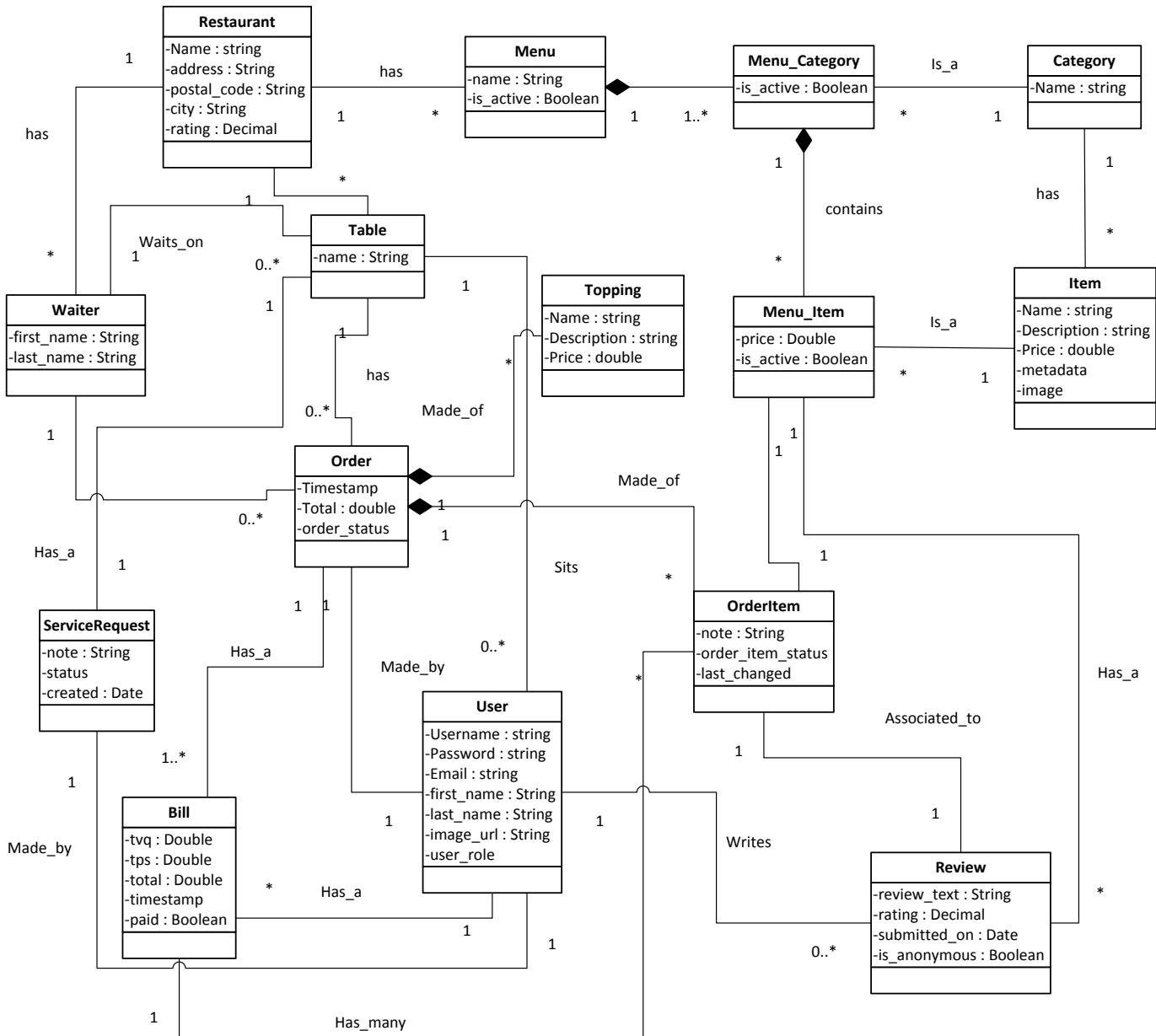
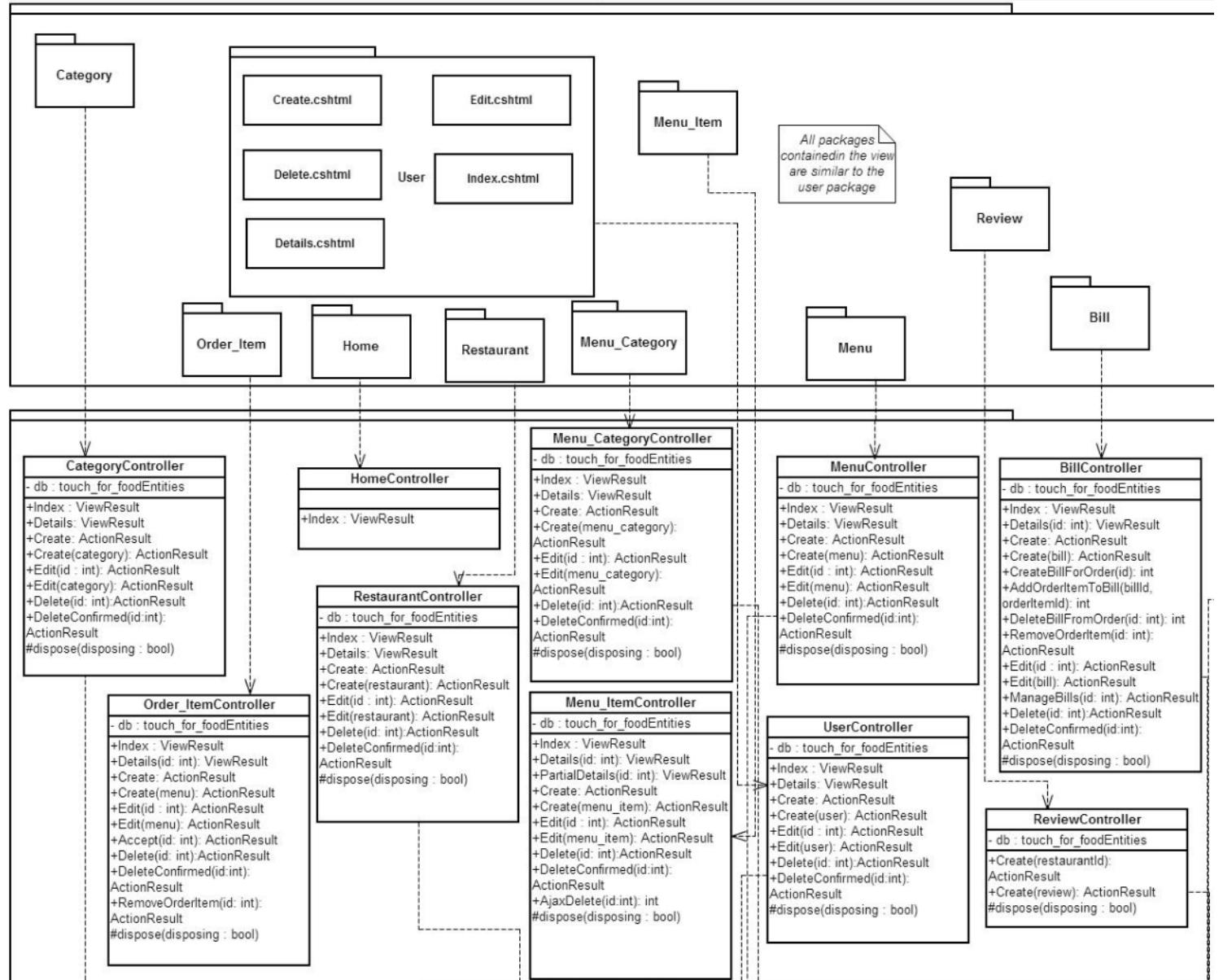


Figure 5-1 Domain Model

5.2 Class Diagram



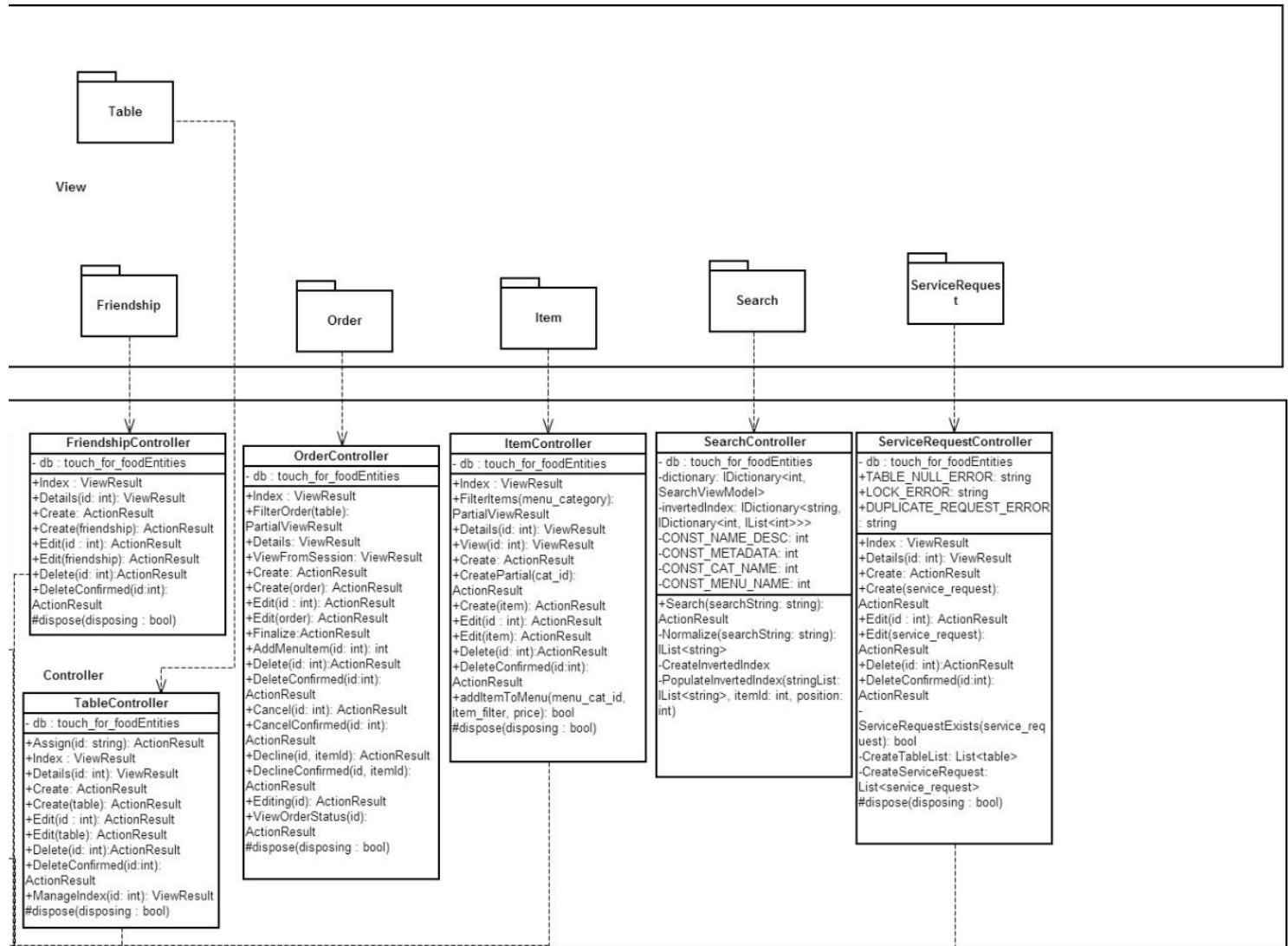


Figure 5-2 Class Diagram

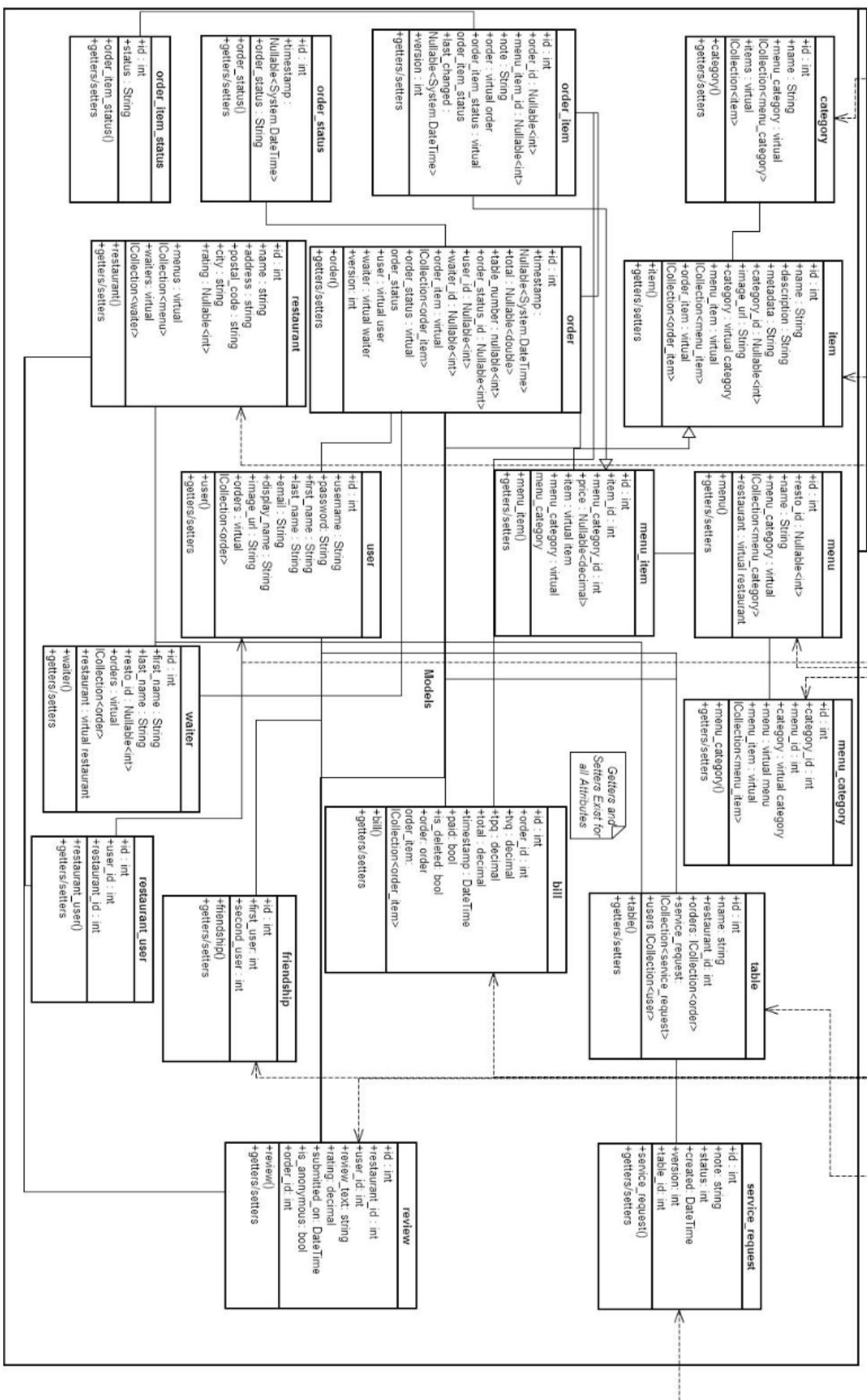


Figure 5-3 Class Diagram Part 2

5.3 ERD Diagram

touch_for_food.edmx

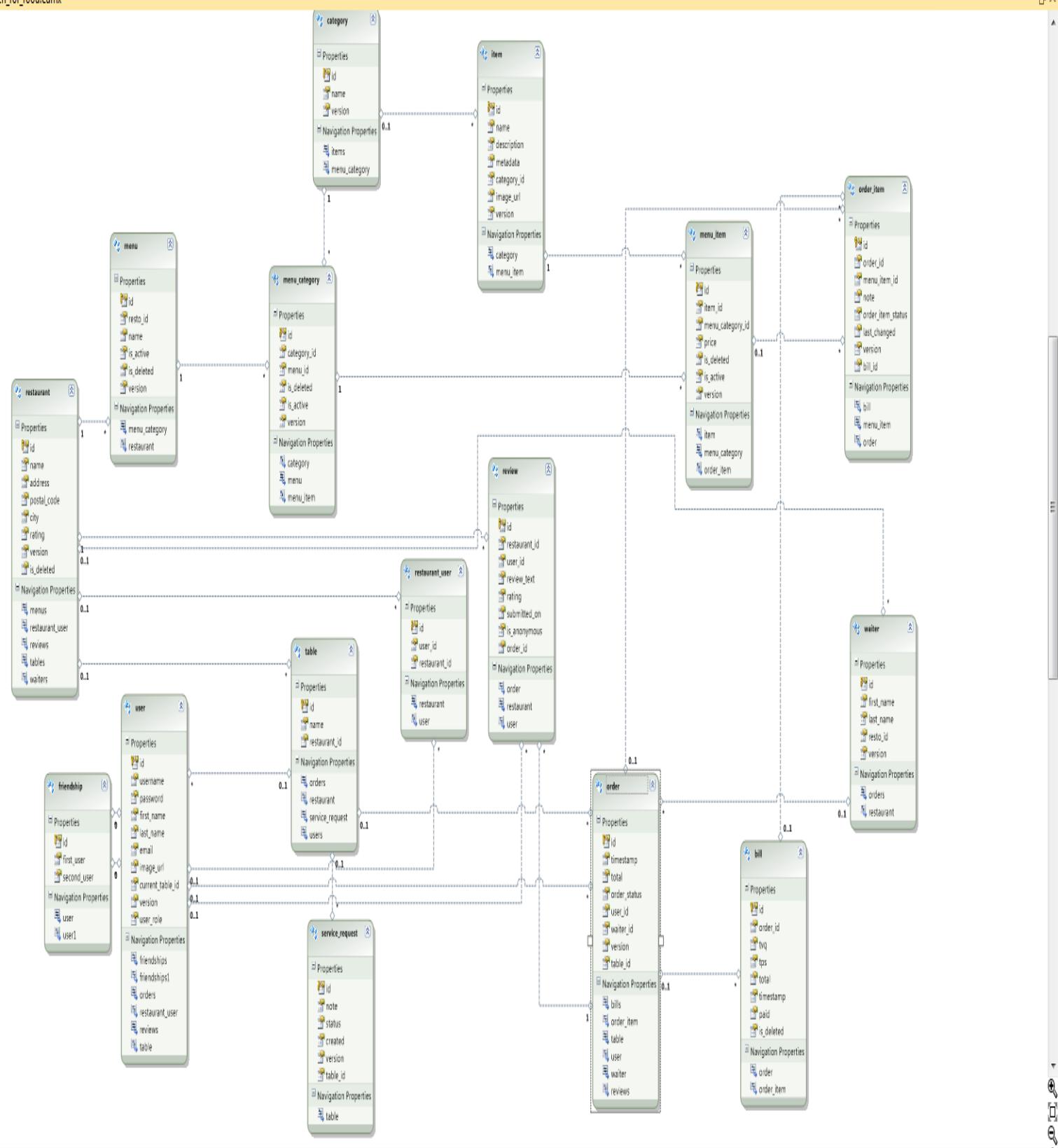
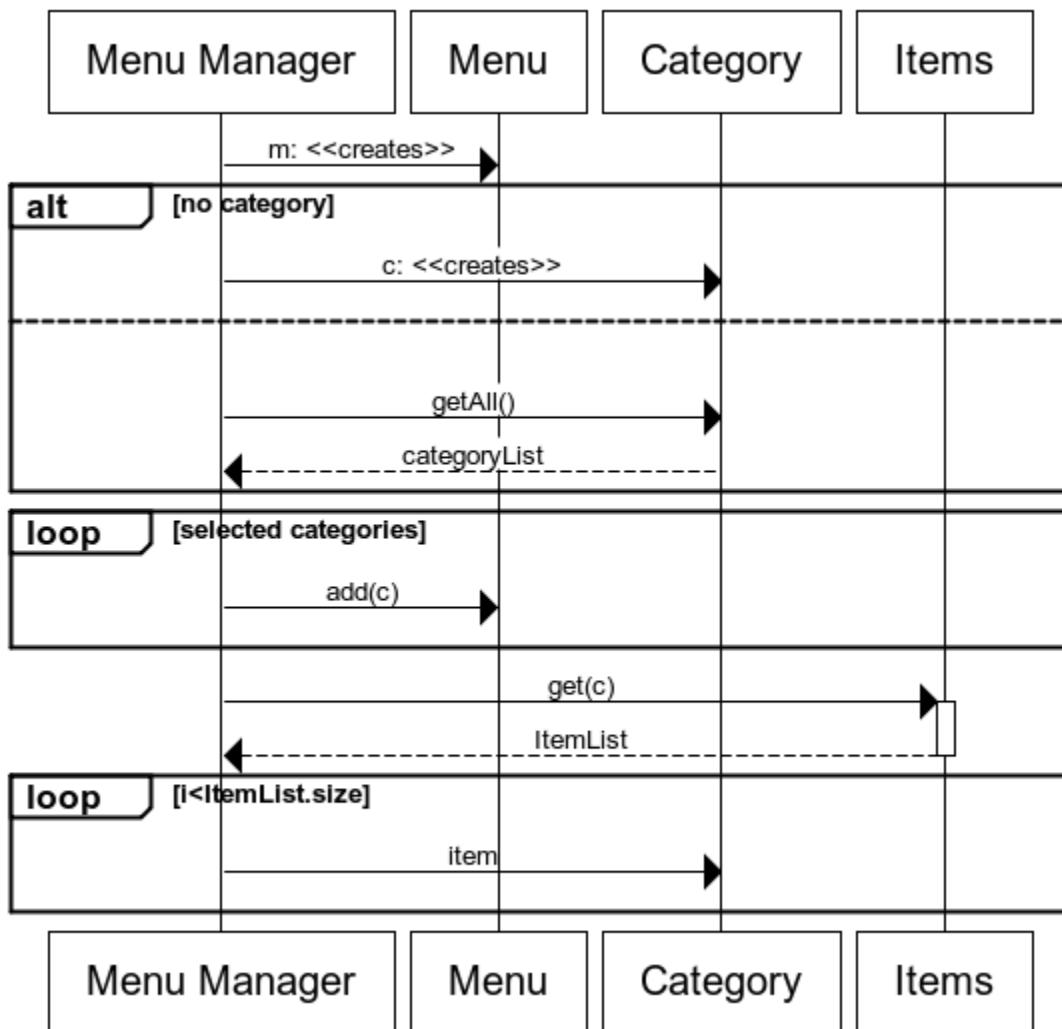


Figure 5-4 ERD Diagram

5.4 Sequence Diagrams



www.websequencediagrams.com

Figure 5-5 Populate Menu Sequence Diagram

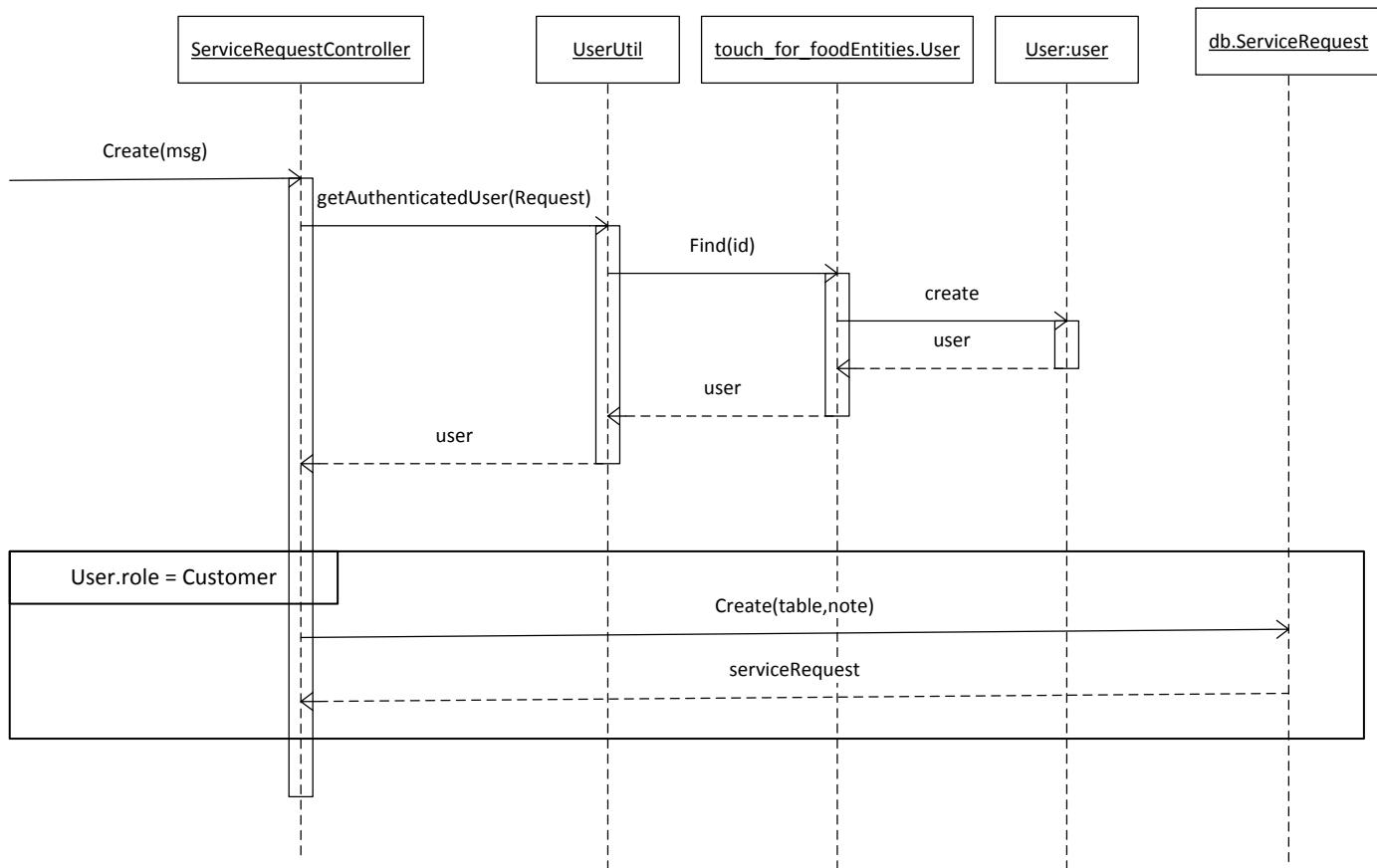


Figure 5-6 Call Waiter Sequence Diagram

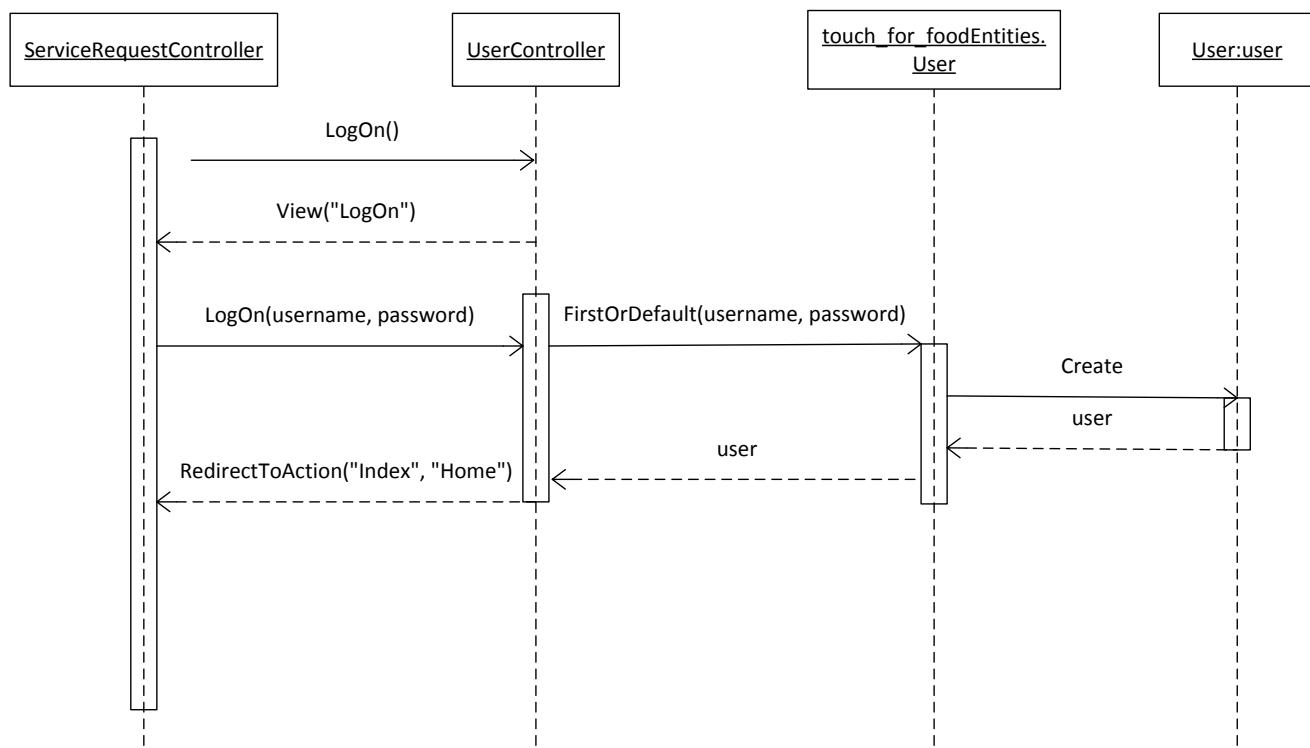


Figure 5-7 Log In Sequence Diagram

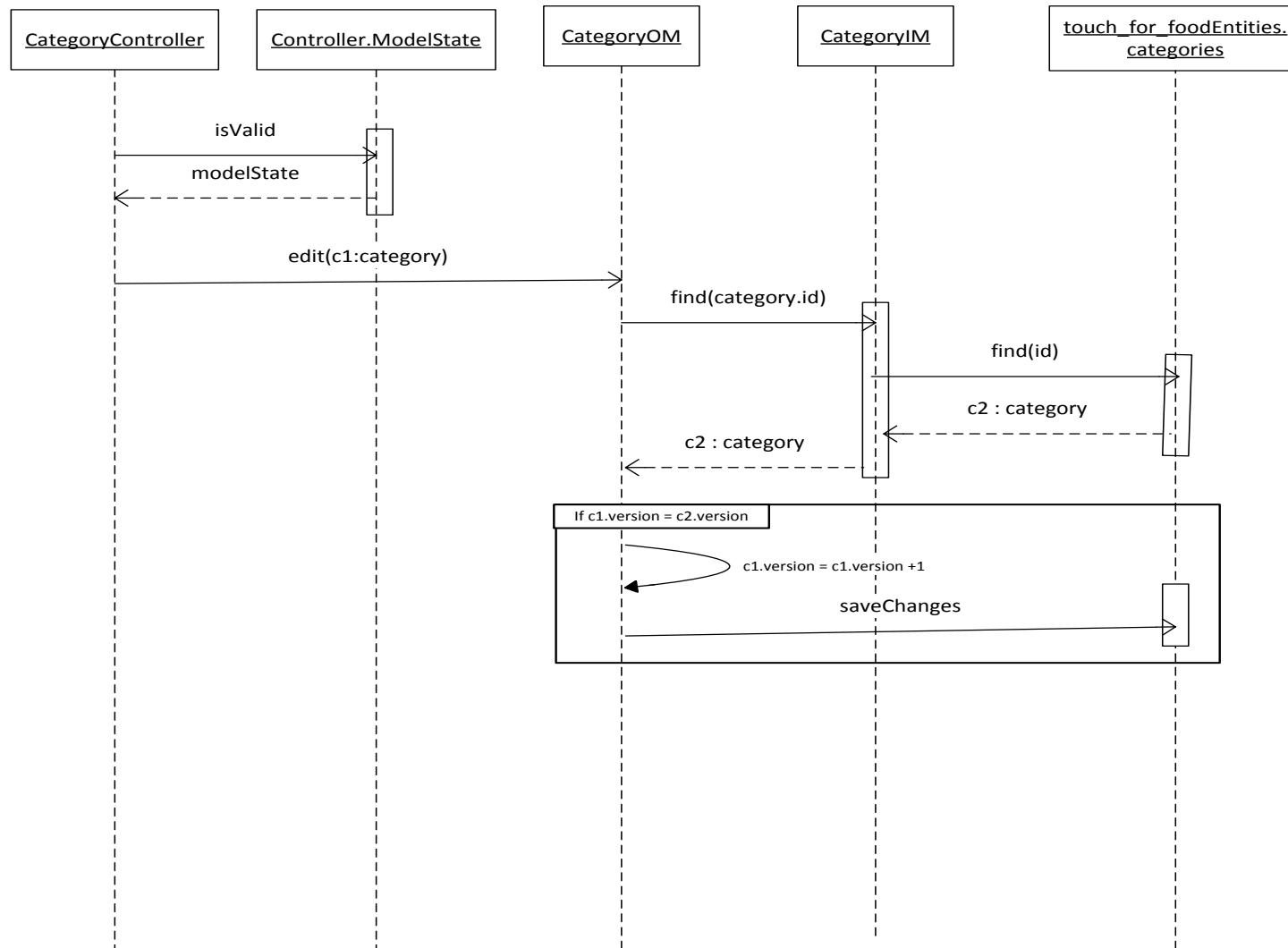


Figure 5-8 Edit Category Concurrency Sequence Diagram

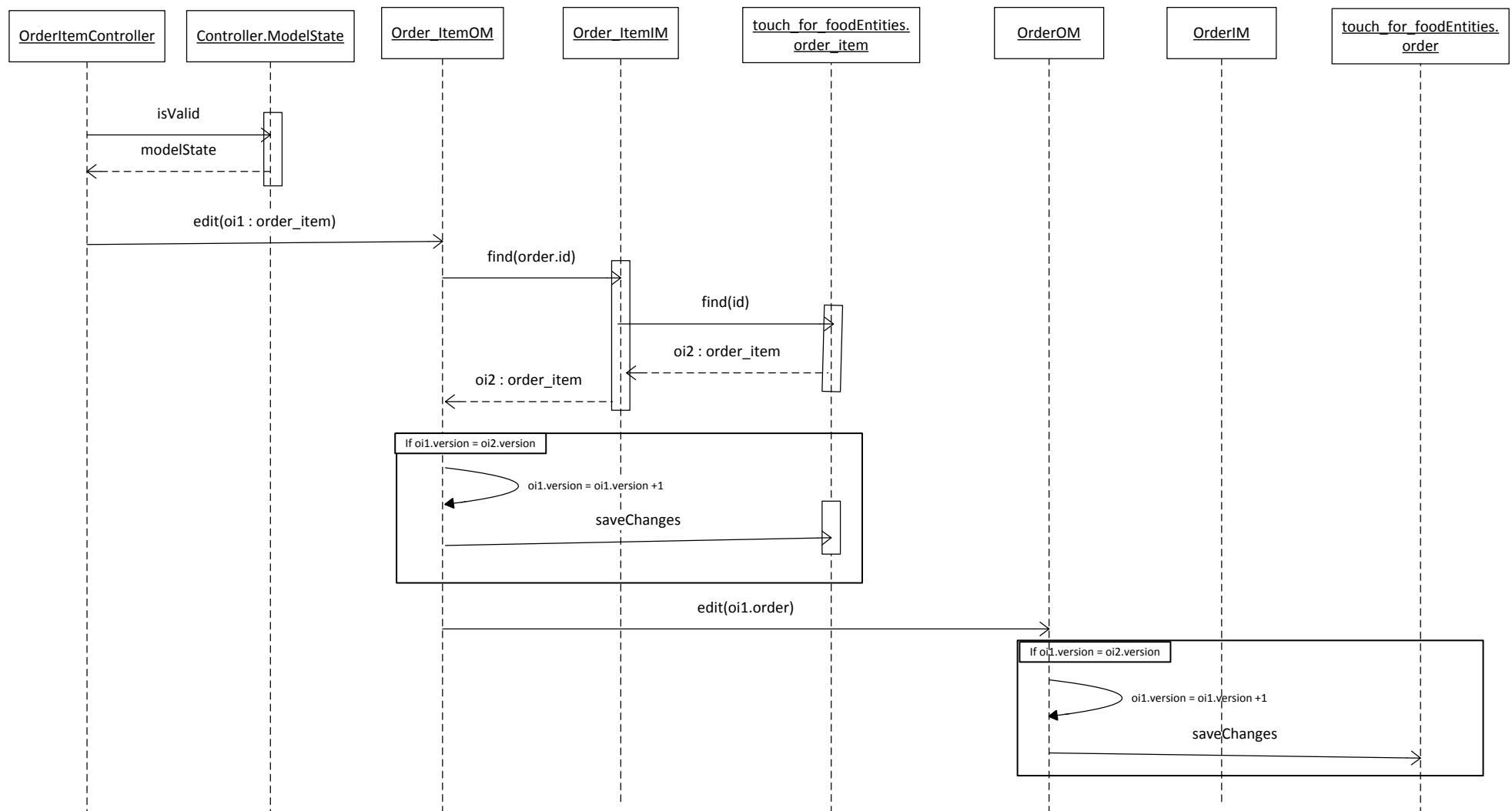


Figure 5-9 Edit Order Item Concurrency Sequence Diagram

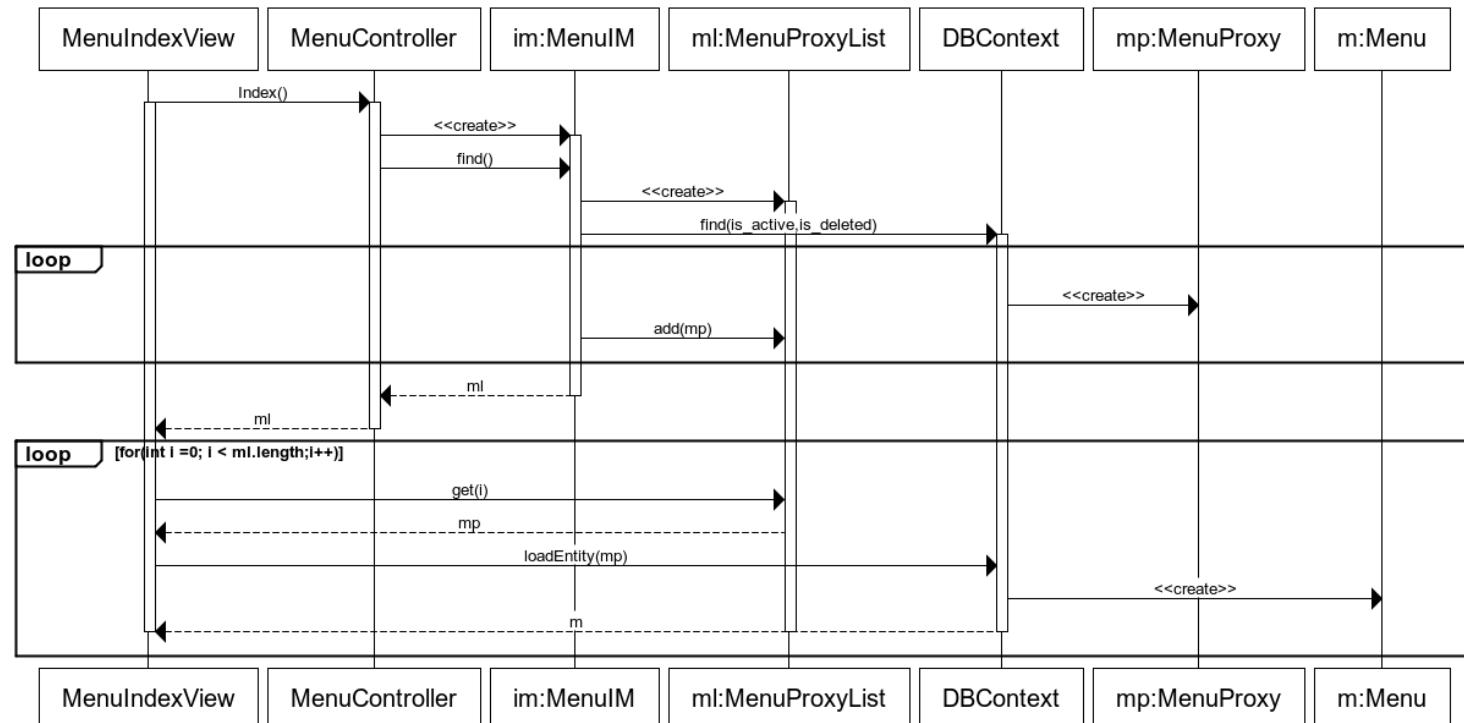
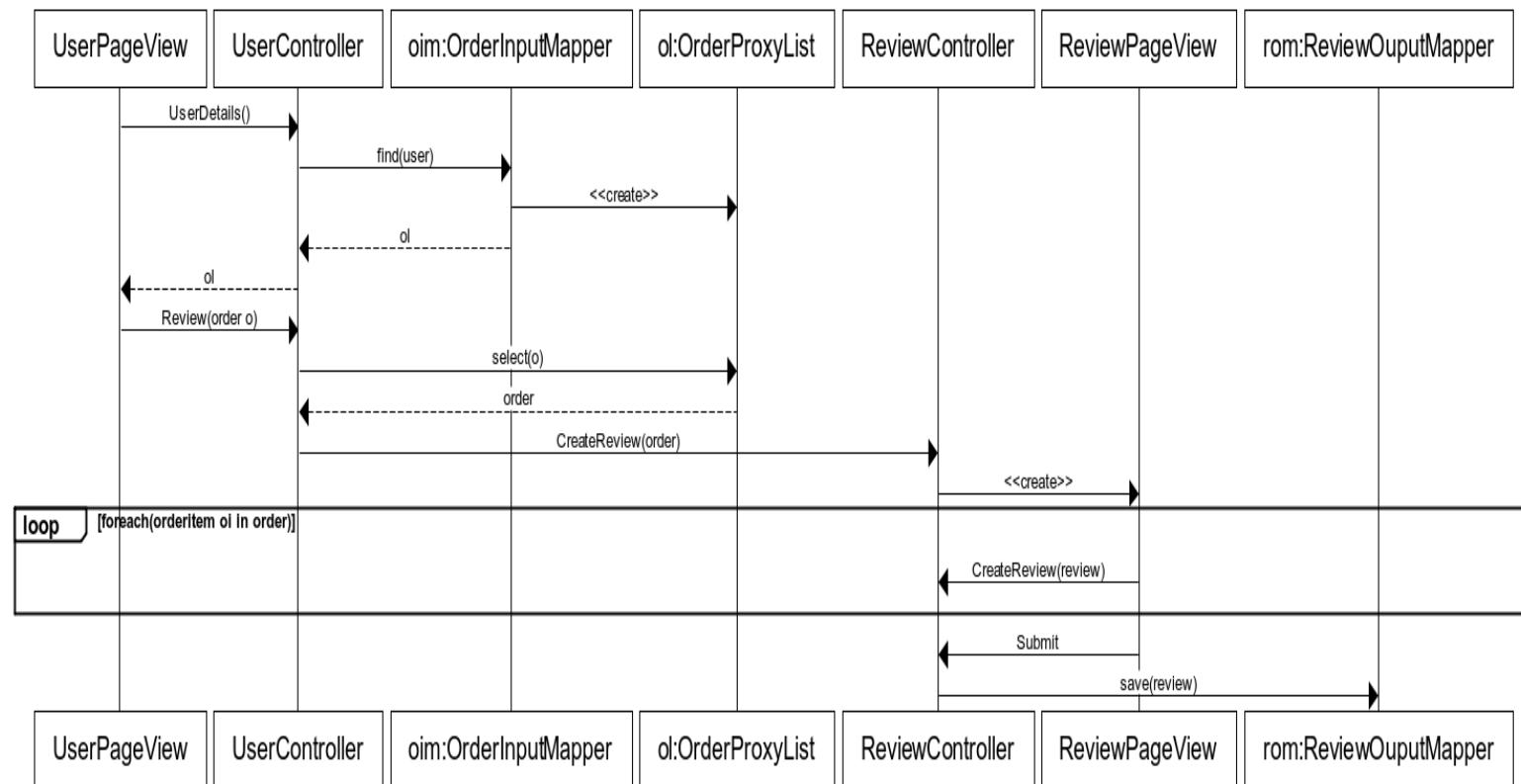


Figure 5-10 Menu Proxy List Sequence Diagram

www.websequencediagrams.com



www.websequencediagrams.com

Figure 5-11 Review Sequence Diagram

5.5 Communication Diagrams

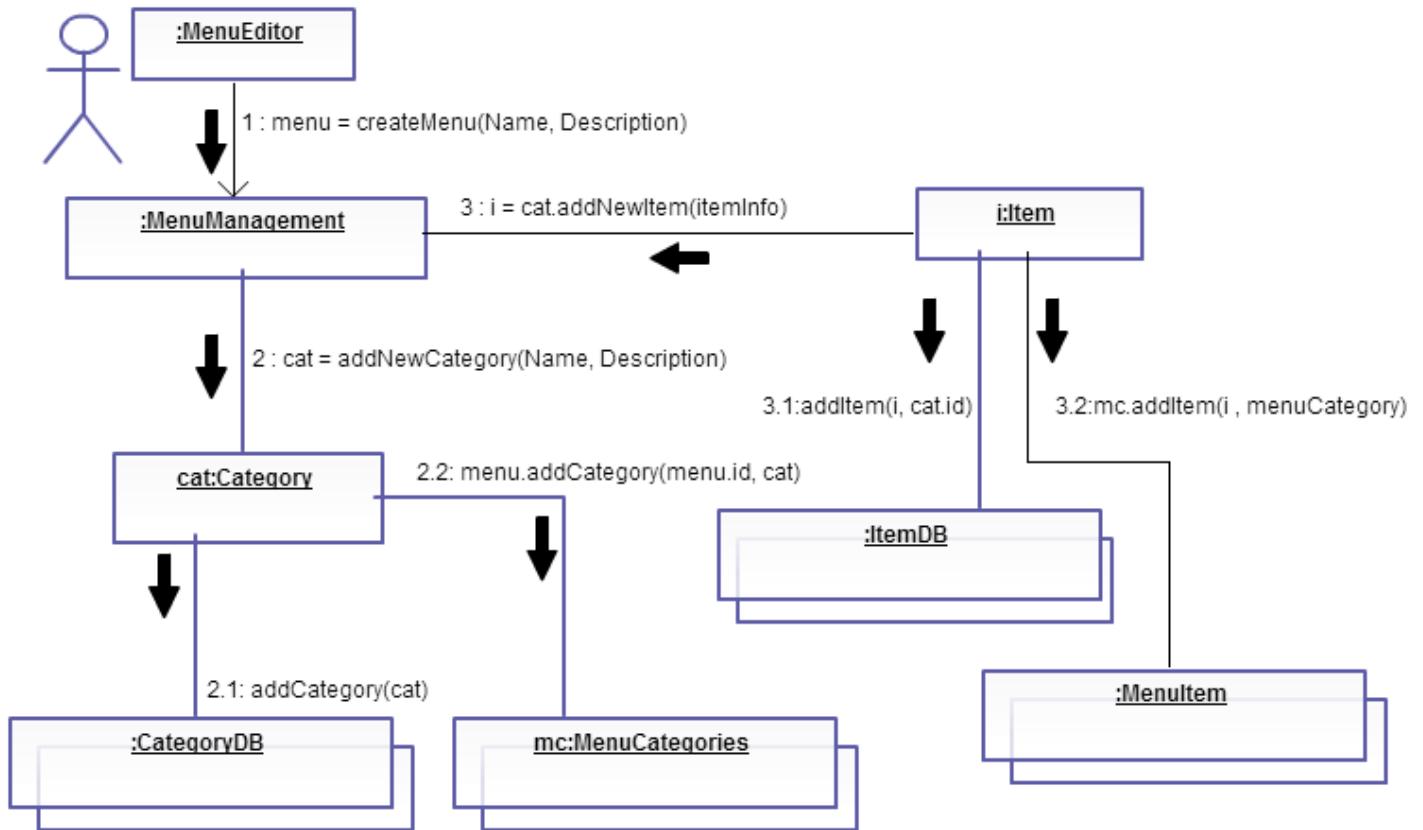


Figure 5-12 Collaboration Diagram for populating a menu from scratch

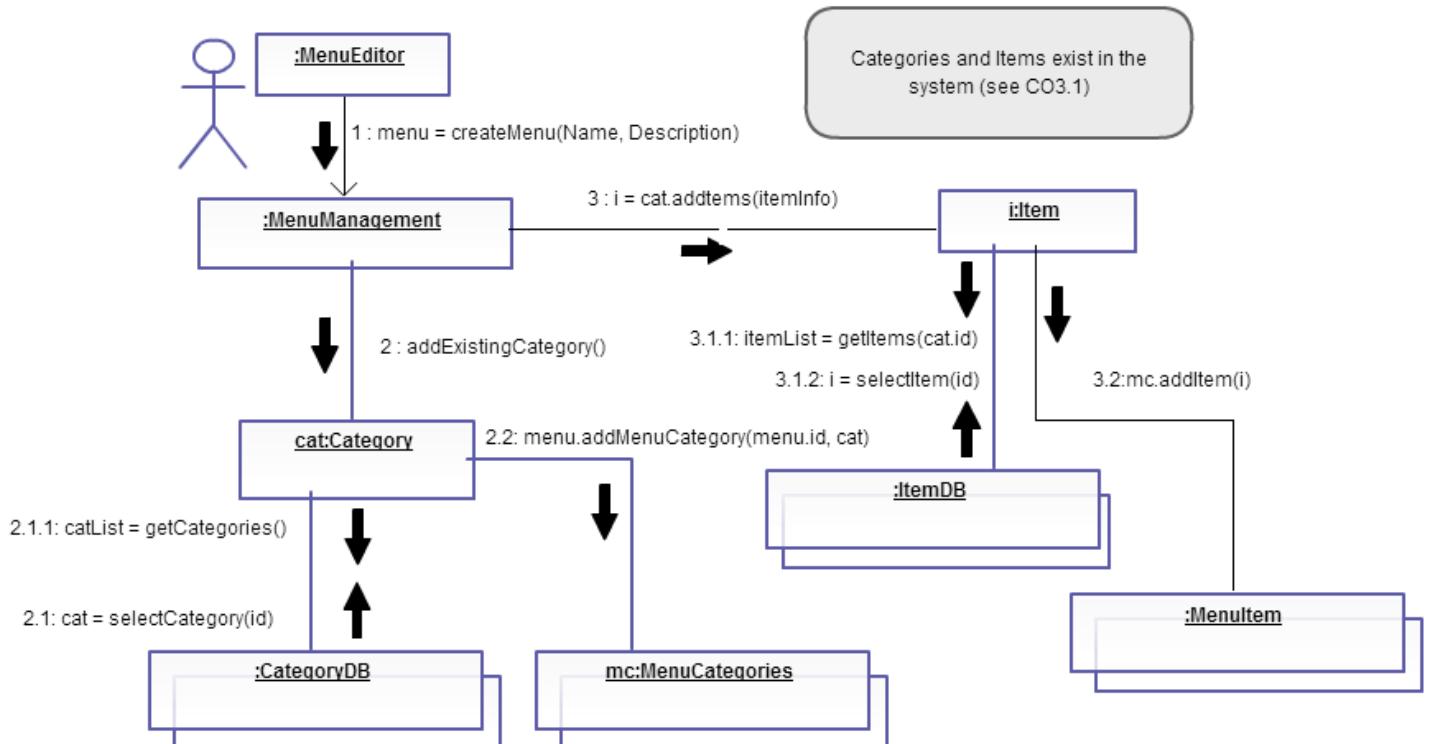


Figure 5-13 Collaboration Diagram for populating menu with pre-existing categories and items

Menu Item Deletion

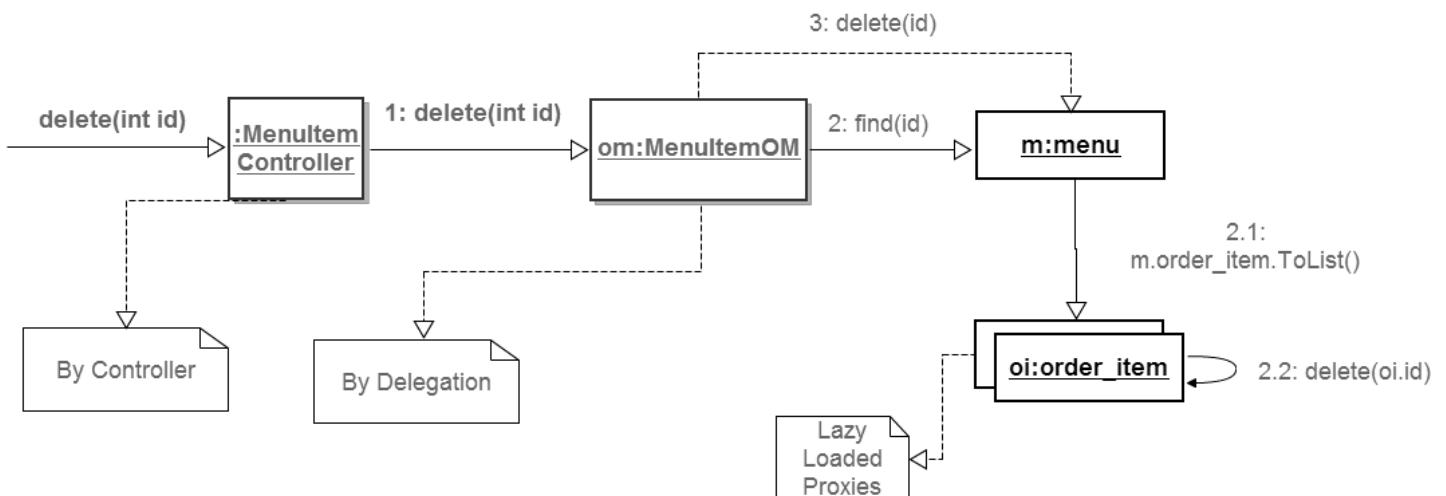


Figure 5-14 Communication Diagram for Deleting a Menu Item

5.6 System Sequence Diagrams

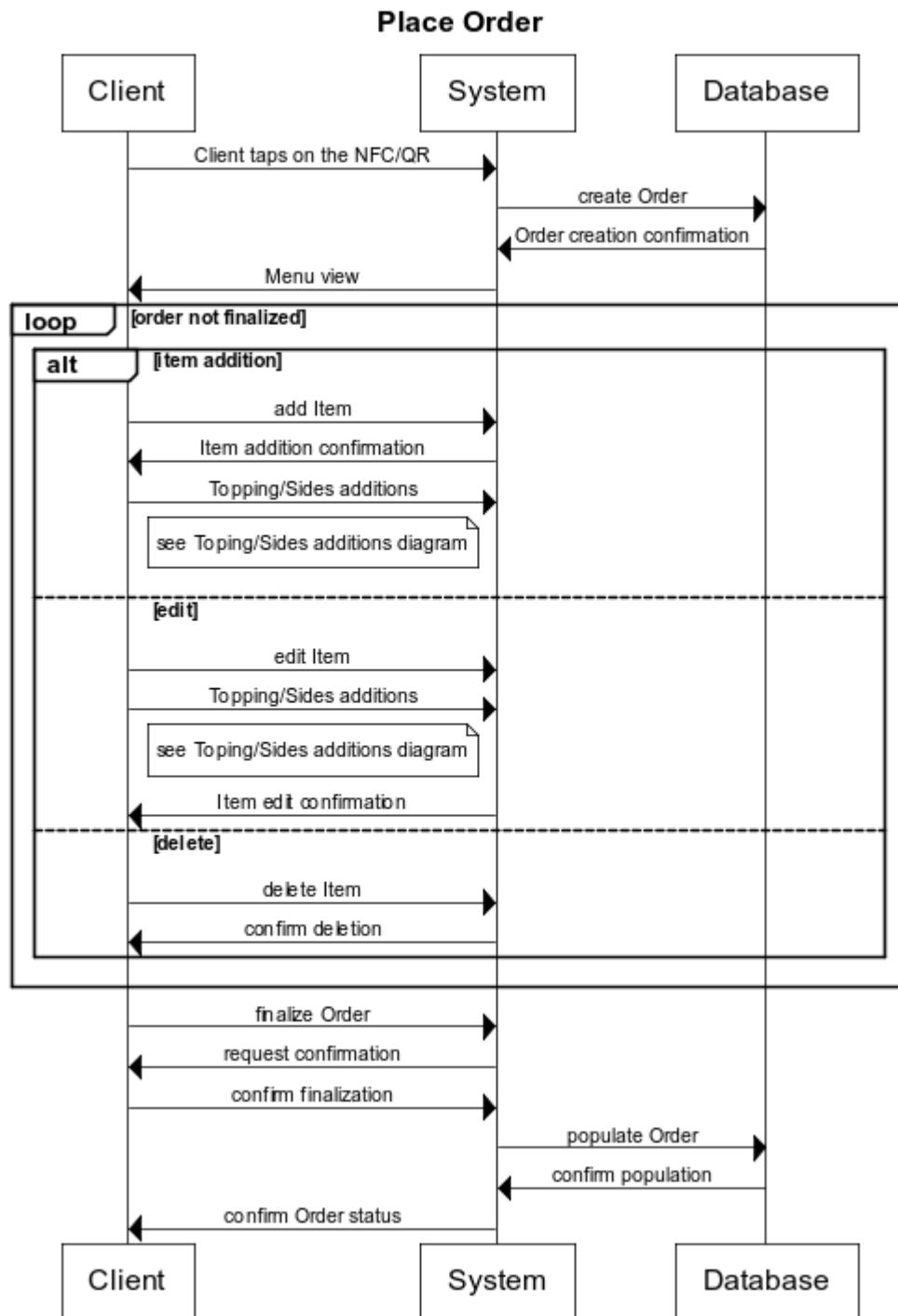


Figure 5-15 System Sequence Diagram for Place Order

Topping/Sides additions

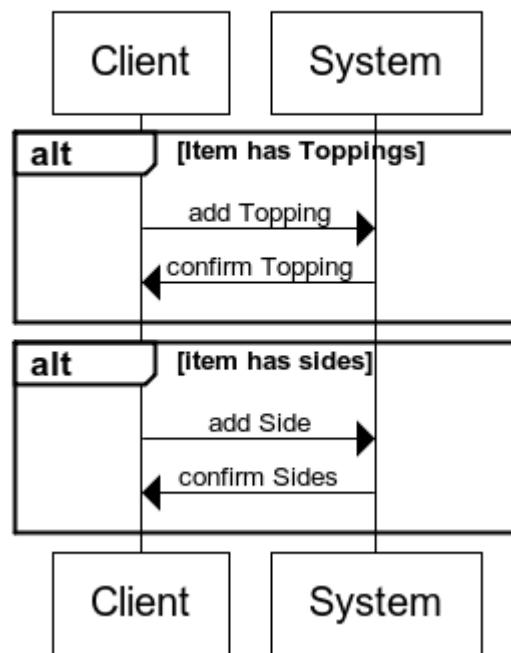
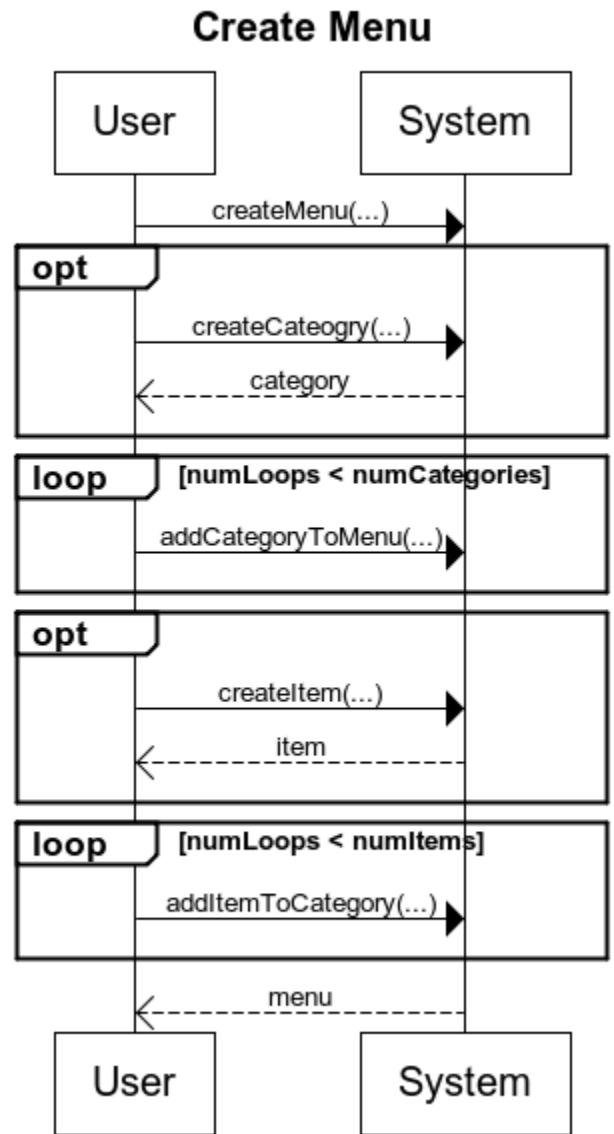


Figure 5-16 System Sequence Diagram for
Topping/Sides Additions



www.websequencediagrams.com

Figure 5-17 System Sequence Diagram for Create Menu

6 Process View

6.1 Activity Diagrams

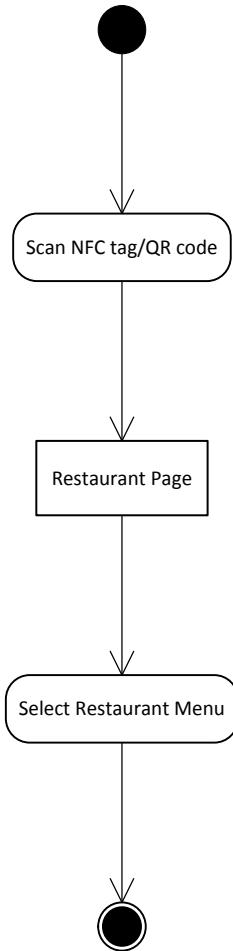


Figure 6-1 View Menu

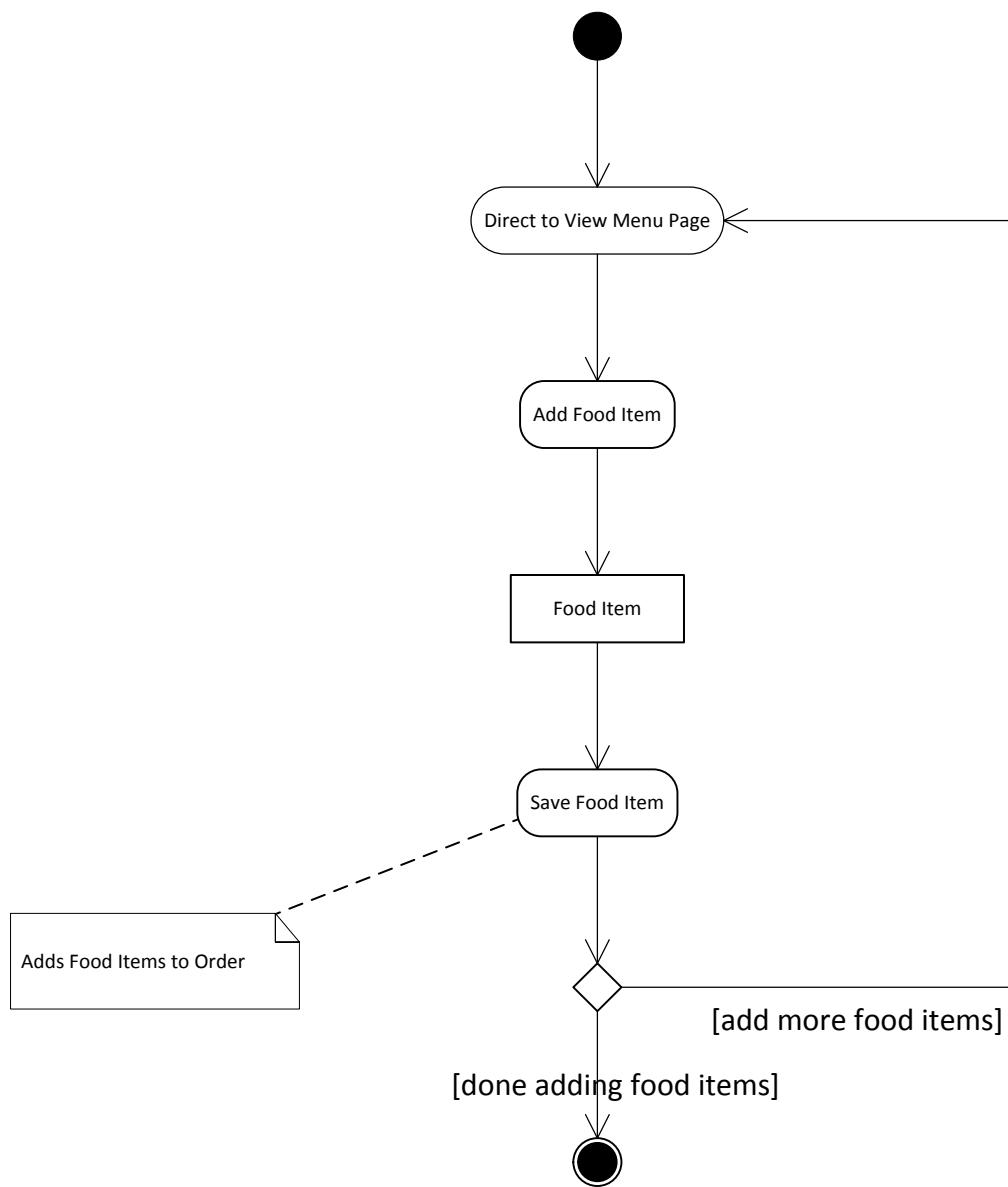


Figure 6-2 Add Food Item to Order

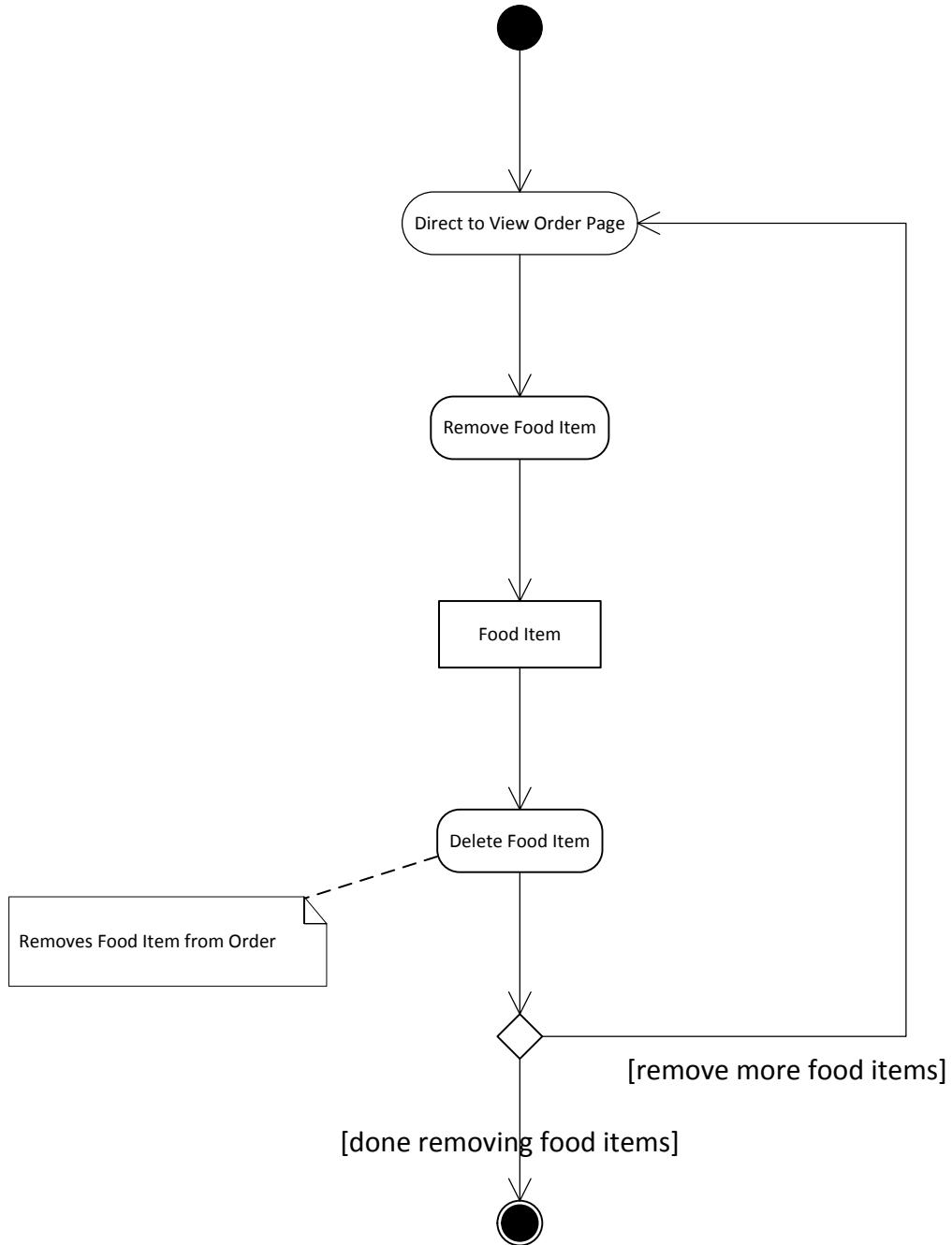


Figure 6-3 Remove Food Item from Order

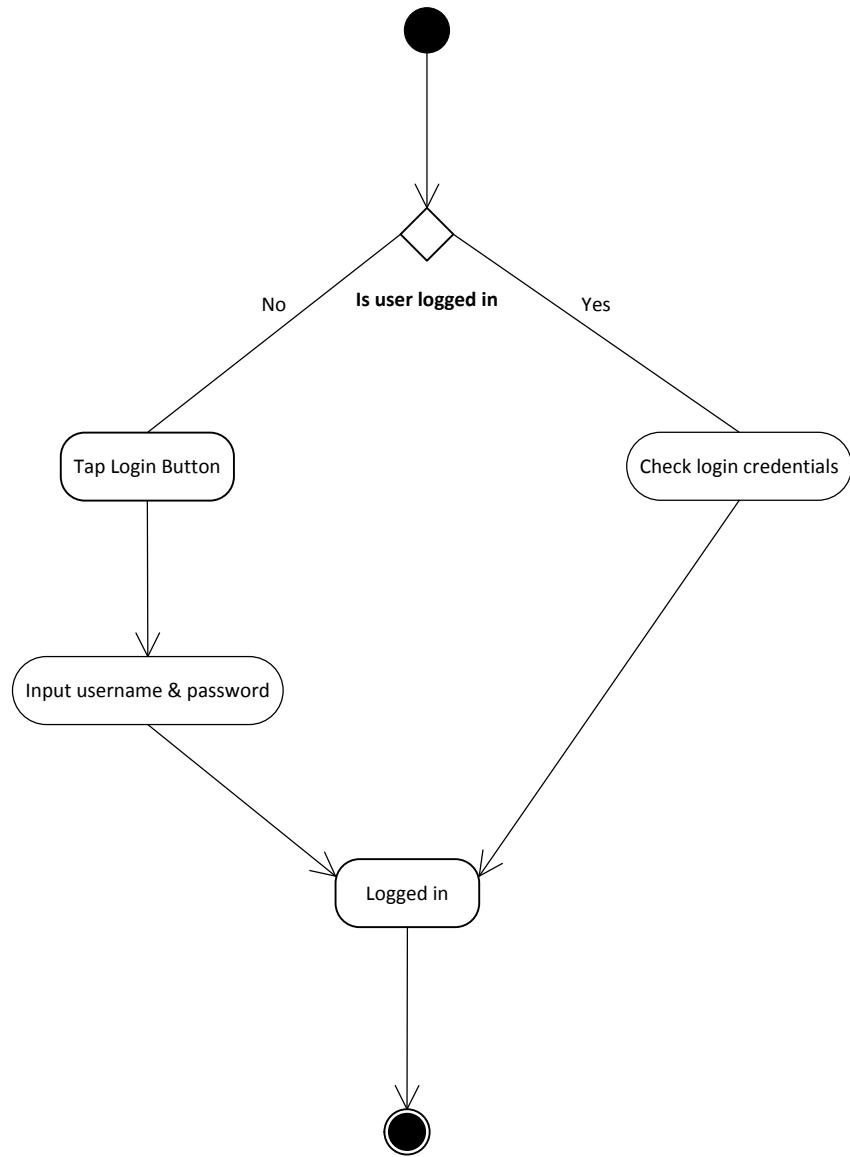


Figure 6-4 Logging In

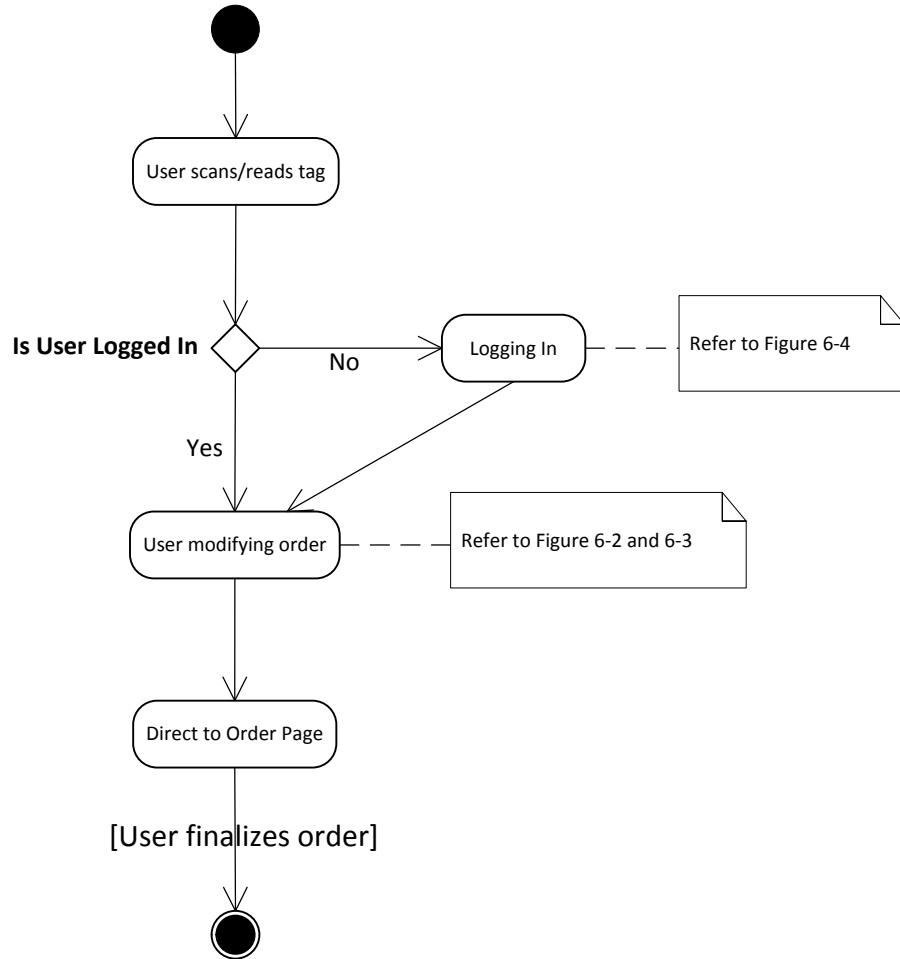


Figure 6-5 Placing an Order

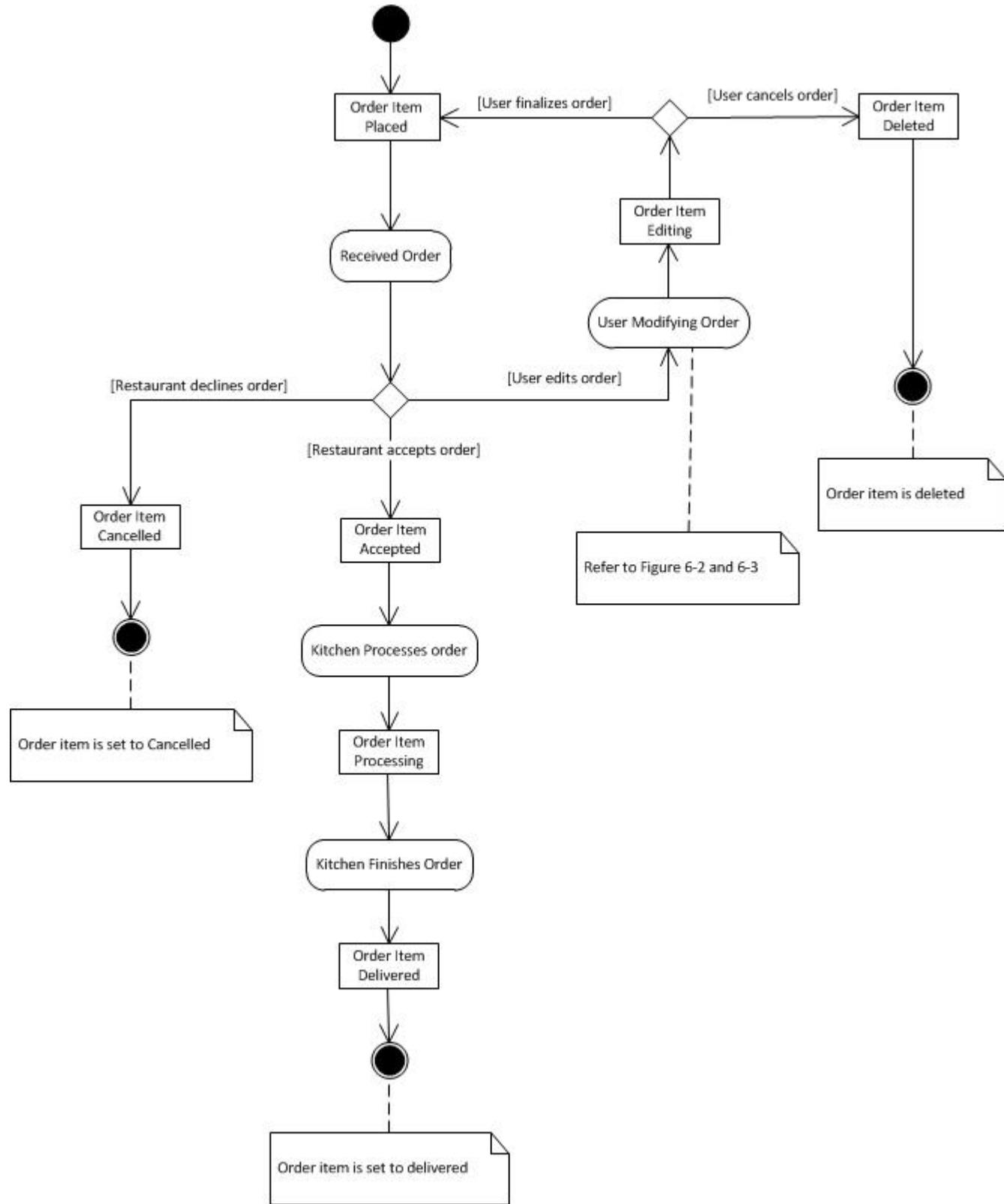


Figure 6-6 Processing an Order

6.2 State Diagrams

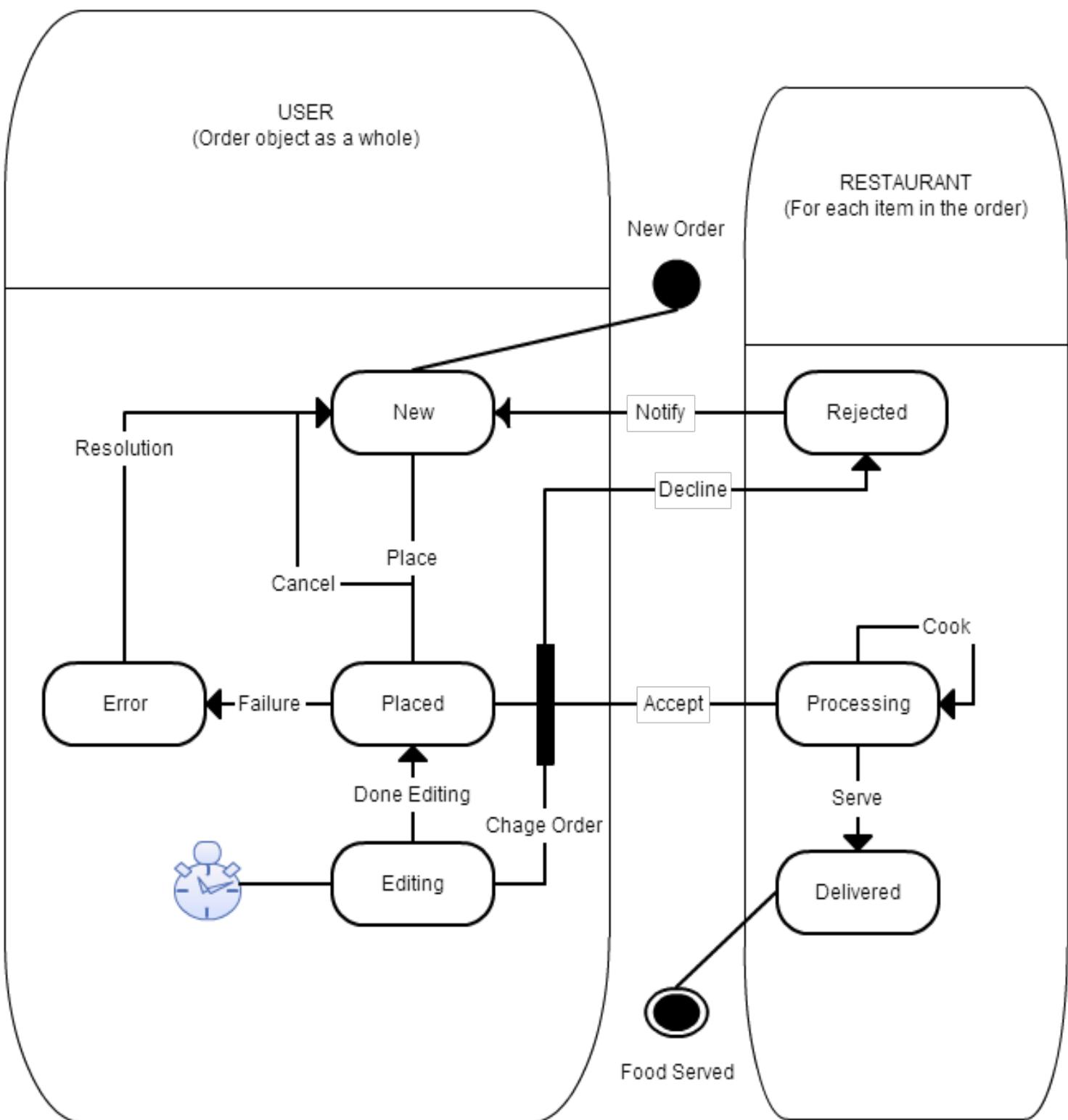


Figure 6-7.1 State Diagram for Order Food

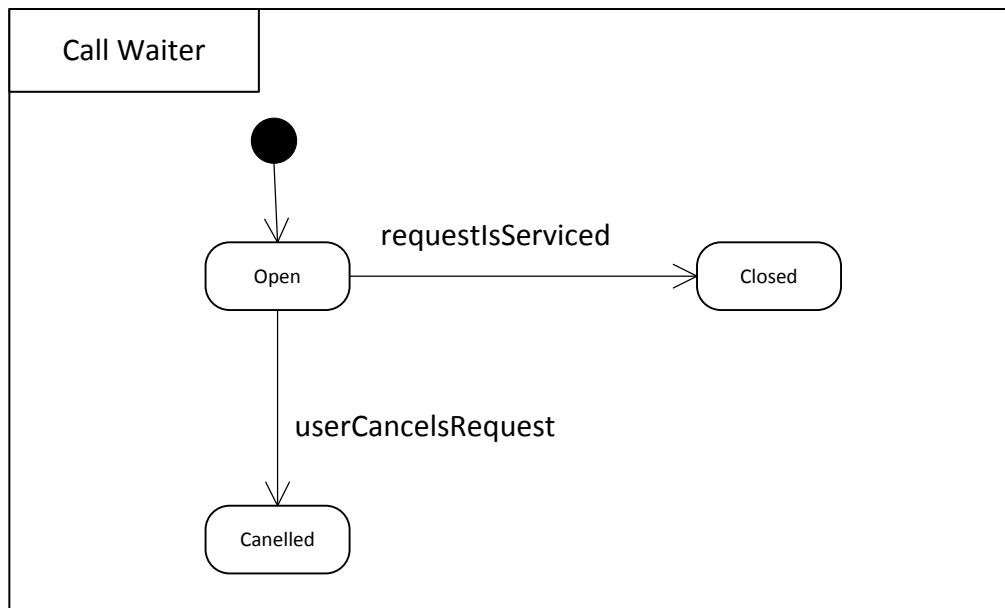


Figure 6-8 Call Waiter State Diagram

7 Physical View

7.1 Deployment Diagram

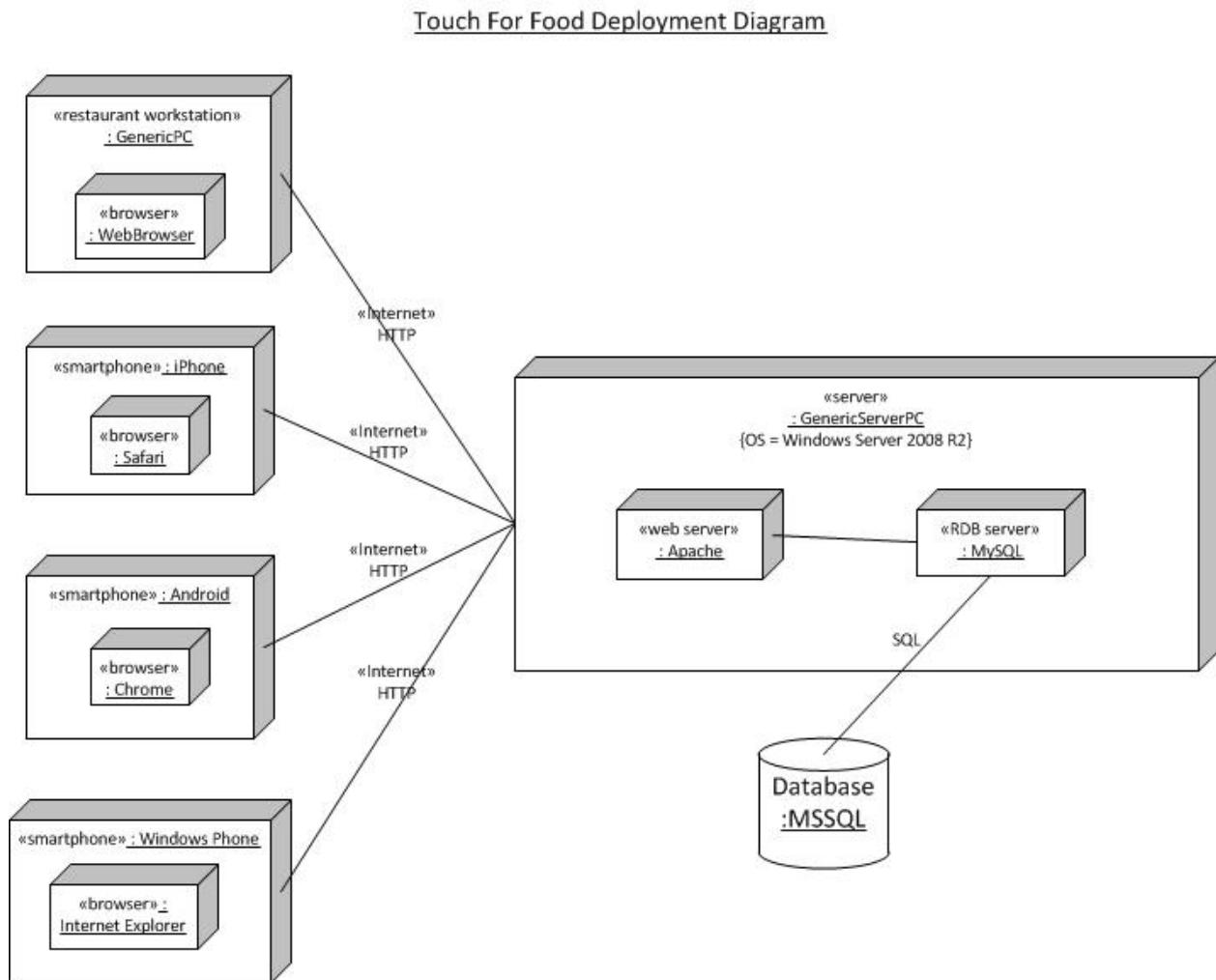


Figure 7-1 Deployment Diagram

8 Size & Performance

From a backend point of view, the element that will impact the performance the most will be the .Net framework along with the MSSQL database server. Many functions used in TFF are already included in the framework and as such are optimized to work better with the MSSQL database server, as opposed to a third party solution like MySQL. Additionally, the .Net framework includes functions and tools that can be used to form relations between different elements in the most efficient way possible.

In terms of the front end, the performance will greatly depend on the power the device accessing the mobile application (or website) actually has. In order to mitigate this potential performance hit, a JavaScript library such as JQuery or Dojo will be used. Additionally, if the user is accessing the system through the mobile application, the use of native (i.e. compiled) code will also help the overall performance of it.

The size of the system will initially be small, with one server to provide the order services. When/if the system begins to grow, then more servers will have to be deployed in order to maintain a balanced load across all of them. Windows Server 2008 R2 has an optional Network Load Balancing (NLB) component which would be used in order to configure a NLB cluster. This NLB cluster would be seen by the outside world as a single virtual server which would distribute the traffic equally between each server that makes up the cluster.

9 Quality

TFF is being developed with quality in mind. The architectural designs have been discussed and made to ensure that the quality of TFF is high.

Maintainability, scalability, security and portability are the 4 focuses of TFF. The code base needs to be flexible and easily maintained. Bugs should be easy to locate within TFF's code base. New and different components must be easy to add/modify. This will be ensured by keeping a consistent architecture as well as deploying necessary GRASP patterns.

Since TFF will be exposed to the internet, and it collects user data, security will also be a main focus. Steps will be taken to ensure that malicious users and hackers do not get access to the system and are not able to abuse it.

For more information on NFR related quality measures, please refer to the SRS document sections 3,4,5,6

Appendix A References

- [1] David Hill's WebLog. (2013) <http://blogs.msdn.com/b/dphill/archive/2009/01/31/the-viewmodel-pattern.aspx>. [Online].
<http://blogs.msdn.com/b/dphill/archive/2009/01/31/the-viewmodel-pattern.aspx>

Appendix B Prototypes

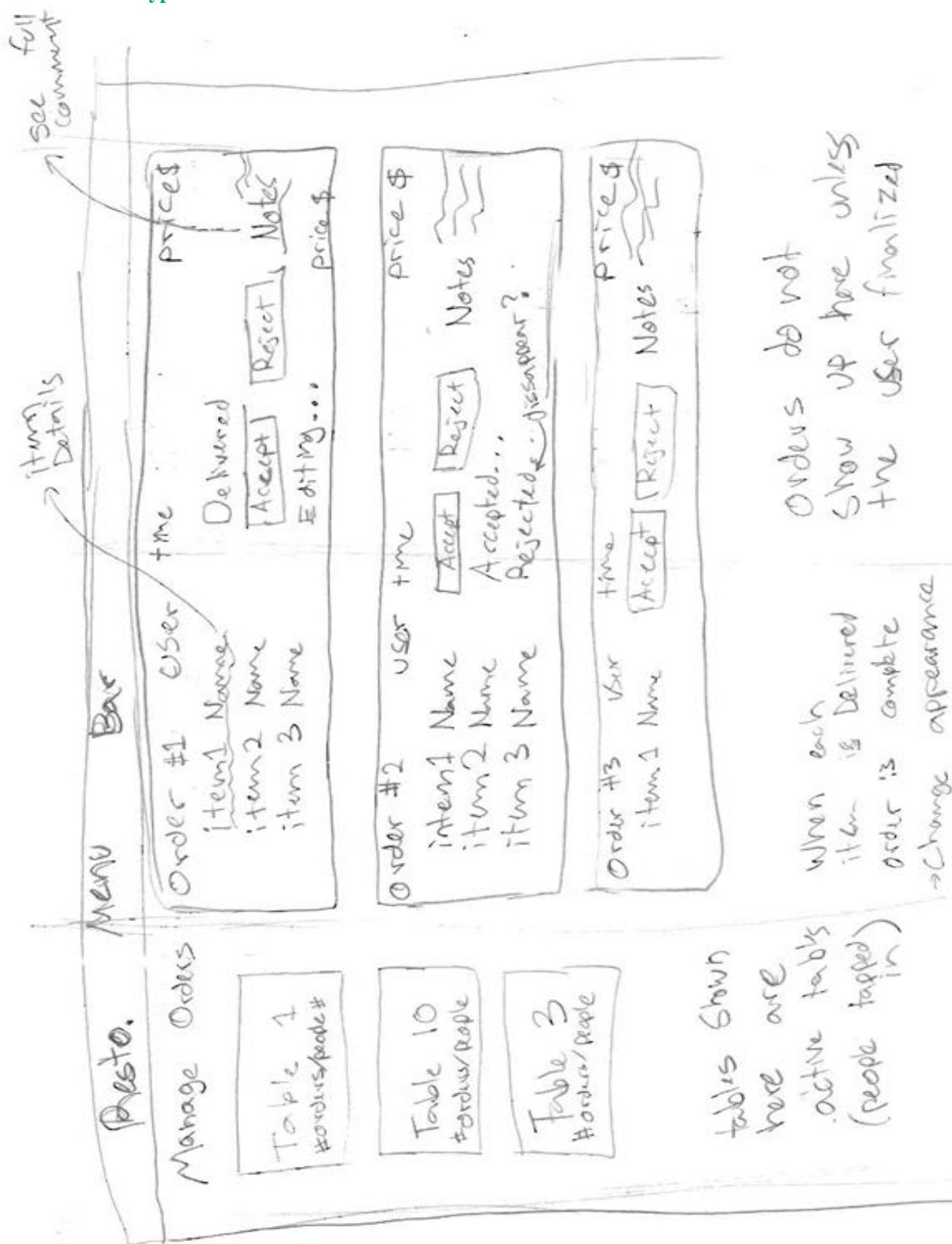


Figure B-1 Manage Order Prototype

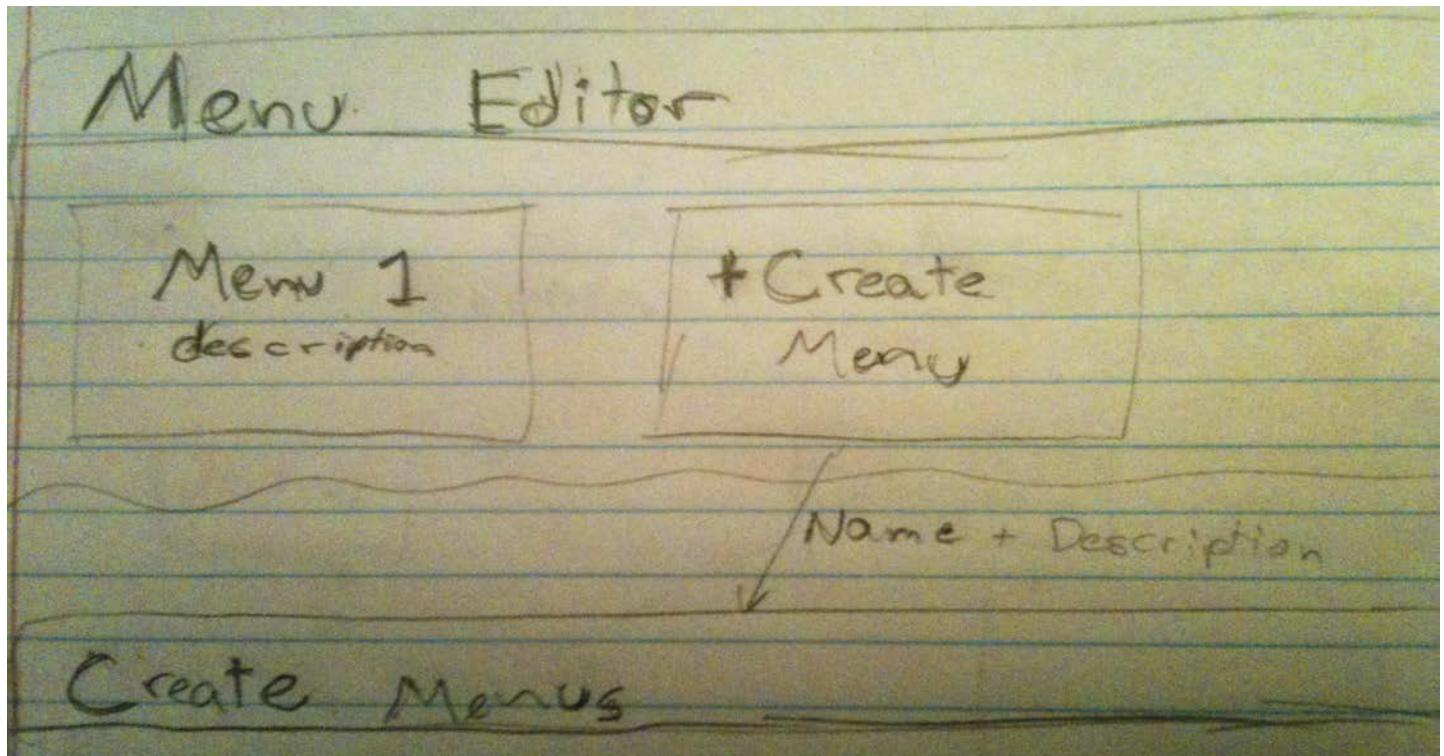


Figure B-2 Menu Editor Main Page

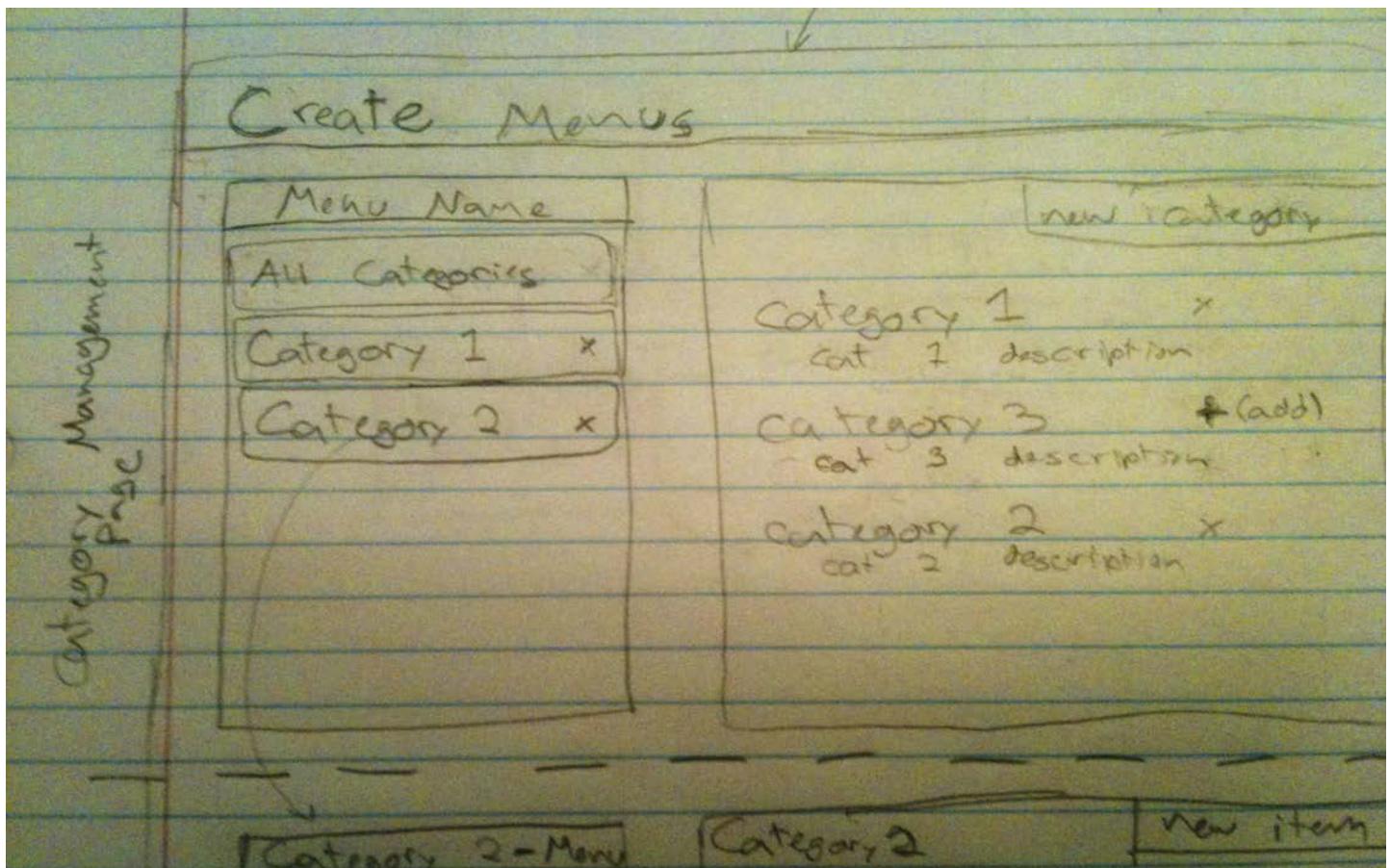


Figure B-3 Category Management Page

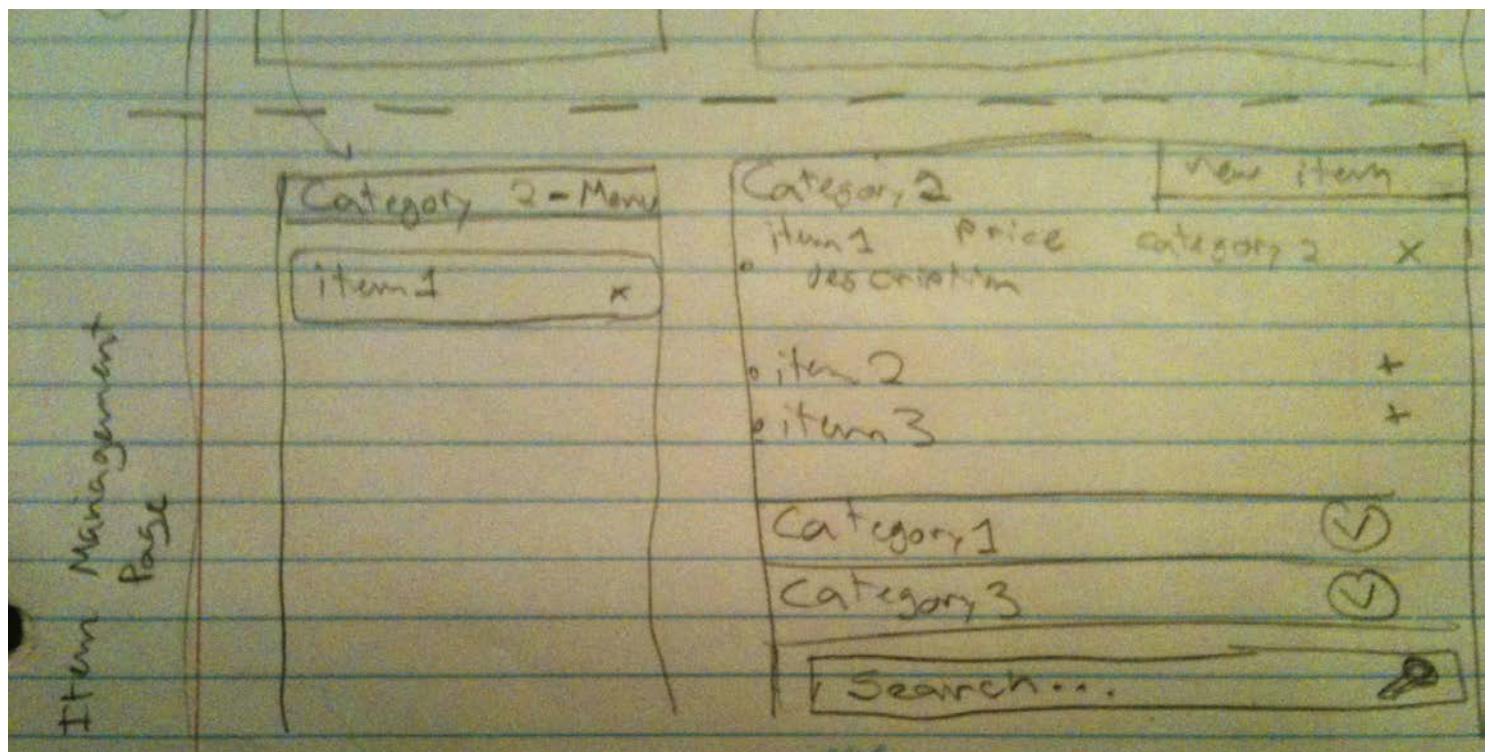


Figure B-4 Item Management Page

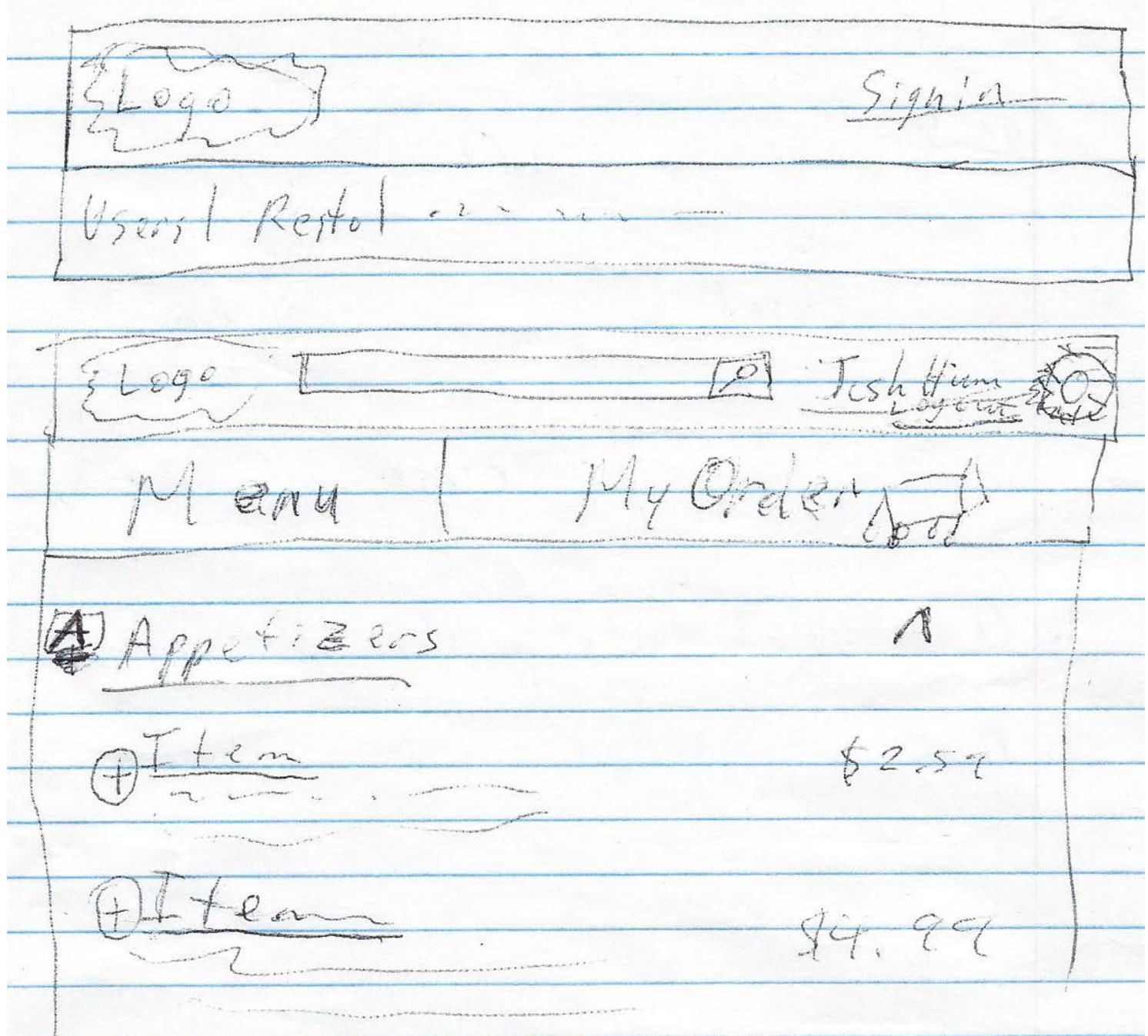


Figure B-5 Order From Menu

~~review resto-id = order-table resto~~
~~+ order-id = order-id~~
 (associations)

Populated from
Order - User DB

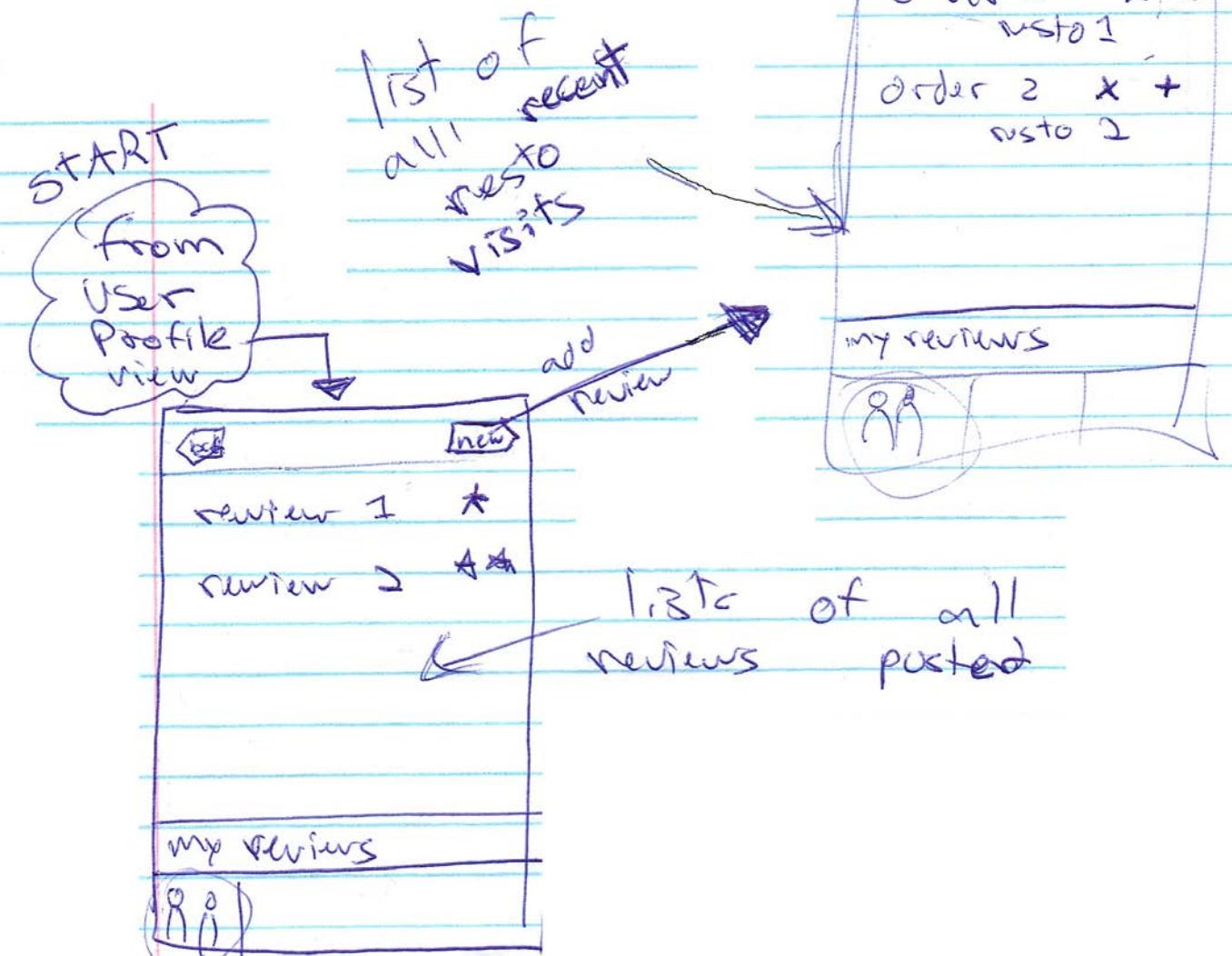


Figure B-6 Leave Review

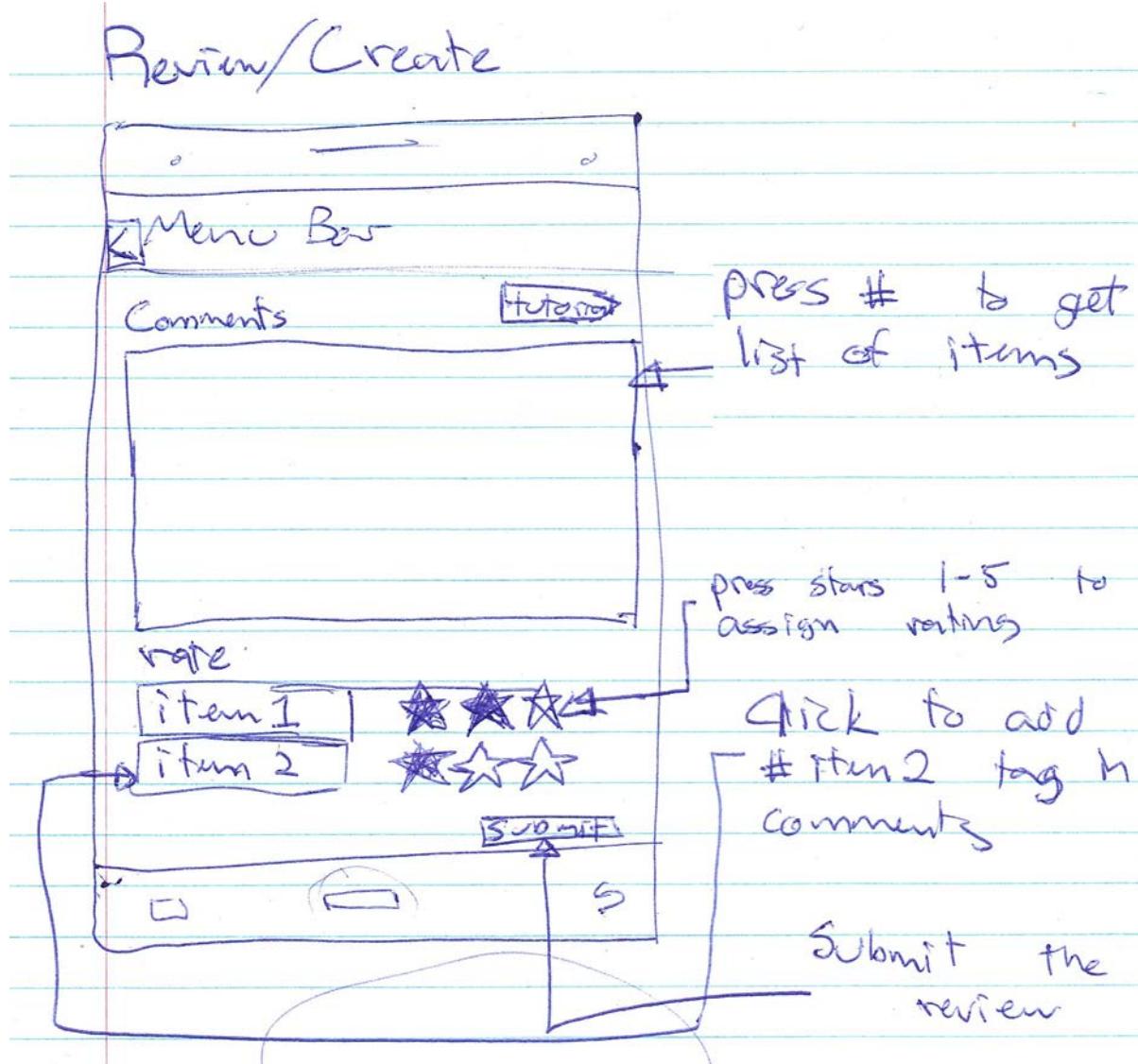


Figure B-7 Create Review

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

User Interface Requirements

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	User Centered Design	5
1.1	User Characteristics	5
1.2	Set of Tasks Performed.....	8
1.3	Context of Use	18
1.4	Stakeholder Objectives	21
2	UI evaluation.....	22
2.1	Overall site architecture	22
2.2	Navigation.....	24
2.3	Feedback	24
2.4	Screen layout.....	26
	Appendix A References.....	30
	Appendix B Glossary	31

Table of Figures

Figure 1-1 Create User Profile Hierarchical Task Model	8
Figure 1-2 Manage User Profile Hierarchical Task Model.....	9
Figure 1-3 View Previous Orders/Reviews Hierarchical Task Model	10
Figure 1-4 Deliver Order Item Hierarchical Task Model	11
Figure 1-5 View Menu Hierarchical Task Model.....	12
Figure 1-6 Populate Menu Hierarchical Task Model	13
Figure 1-7 Manage Tables Hierarchical Task Model	14
Figure 1-8 Manage Bill Hierarchical Task Model	15
Figure 1-9 Place Order Hierarchical Task Model.....	16
Figure 1-10 Submit a Review Hierarchical Task Model	17
Figure 1-11 Call Waiter Hierarchical Task Model	17
Figure 2-1 TFF Top Menu According To User Role.....	24
Figure 2-2 Category Manager Grid	27
Figure 2-3 Order from Menu	27
Figure 2-4 User Creation	28
Figure 2-5 Menu Details Overlay	29

Touch For Food

User Interface Requirements

Version 7.20

Revision History

Date	Rev.	Description	Author(s)
2012-09-17	0.0	Document Creation	Katrina Anderson
2013-01-20	6.1	Section 1.3 Context of Use	Mikhail Levkovsky
2013-01-23	6.2	Section 1.4 Stakeholder Objectives	Christian Daher
2013-01-24	6.3	Section 2.1 Overall Site Architecture	Matthew Tam
2013-01-26	6.4	Section 1.2 Set of Tasks Performed	Cristian Asenjo
2013-01-26	6.5	Reviewed Section 1.4 Stakeholder Objectives	Matthew Tam
2013-01-26	6.6	Section 1.2 updated	Cristian Asenjo
2013-01-26	6.7	Added Section 2.2 Navigation	Katrina Anderson
2013-01-26	6.8	Added section 1.1	Cynthia Donato
2013-01-26	6.9	Updated section 1.3 Context of Use	Mikhail Levkovsky
2013-01-26	6.10	Section 1.2 updated	Cristian Asenjo
2013-01-27	6.11	Reviewed section 1.1	Cristian Asenjo
2013-01-27	6.12	Updated section 1.3 Context of Use	Mikhail Levkovsky
2013-01-27	6.13	Section 2.3 Feedback	Ryan Nasr
2013-01-28	6.14	Reviewed Section 2.2 Navigation	Josh Hum
2013-01-28	6.15	Added Section 2.4 Screen Layout	Josh Hum
2013-01-28	6.16	Reviewed Section 2.1 and formatted document	Katrina Anderson
2013-01-29	6.17	Updated Section 1.3 as per review	Mikhail Levkovsky
2013-01-29	6.18	Review and updated section 1.2	Cynthia Donato
2013-02-07	7.19	Review document for submission and make necessary updates	Cynthia Donato
2013-02-12	7.20	Updated Figure 2-5 and Figure 2-3	Josh Hum

1 User Centered Design

1.1 User Characteristics

The TFF application has six basic user types; administrator, developer, customer, restaurant owner, chef and waiter/waitress. The following personas will illustrate the details of a typical type of user. Personas don't represent specific people, but are developed based on the characteristics of a real user or group of users. Every persona will have real person attributes like name, age, characteristics, goals and background. [1]

1.1.1 Persona 1 – Administrator

Name: Mat

Background:

- 30 years old, Male
- Works for Cloud9
- Friendly

Goals:

- Create user accounts
- Administer user accounts
- Troubleshoot technical issues

Frustrations:

- Unclear navigation of software
- Complicated steps to perform a task

About Mat:

Mat is an administrator for the Cloud9 software company. Mat is computer savvy and has high standards in regards to how websites look and feel should be implemented.

His mandate is to administer accounts purchased for the TFF application. His daily tasks include creating user accounts and providing them with the necessary privileges to be able to create and administer their restaurant site in TFF. Mainly Mat troubleshoots various issues which clients report in the TFF application and finds solutions for these issues.

1.1.2 Persona 2 – Developer

Name: Fred

Background:

- 27 years old, Male
- Works for Cloud9
- Friendly and talkative

Goals:

- Develop new features for the TFF application
- Maintain the TFF application
- Test the TFF application

Frustrations:

- Lack of visual representation of important information when debugging and testing

About Fred:

Fred is a web developer at Cloud9. Fred is computer savvy and enjoys spending hours coding and working out difficult issues. His goal is to develop software that is easily maintainable and functioning according to the needs of his clientele. It is also important to Fred that all deployed software be thoroughly tested to ensure optimized performance for customers using the application.

1.1.3 Persona 3 – Customer

Name: Jonathan

Background:

- 26 years old, Male
- Busy lifestyle
- Is a iPhone enthusiast

Goals:

- Order food in a timely manner
- Be able to effortlessly flag down a waiter when needed
- Be able to view billing information in real time
- Be able to leave reviews on different menu items

Frustrations:

- Waiting for waiters/waitresses to place an order wastes time on a short lunch break
- Difficult to get the attention of a waiter/waitress when a restaurant is very busy
- Difficulty to predict how much of a tab we have collected prior to receiving the bill
- Ordering a dish and not receiving what was expected

About Jonathan:

Jonathan is construction worker who loves his iPhone smartphone. With his busy lifestyle Jonathan's goal is to be able to maximize his lunch hour by decreasing the time spent waiting on waiters and waitresses in restaurants.

Jonathan is not technology savvy but can easily manipulate iPhone and mobile apps that are simple and intuitive.

1.1.4 Persona 4 – Restaurant Owner

Name: Sara

Background:

- 25 years old, Female
- Busy lifestyle
- Small business restaurant owner

Goals:

- Provide efficient service for breakfast, lunch and dinner
- Provide customers a unique experience
- Easily manager orders
- Easily manage bills
- Collect statistical information on clientele
- Create waiters/waitresses effortlessly
- Manage restaurant attributes

Frustrations:

- Difficulty finding waiters/waitresses
- Difficult to collect customer feedback and statistical information

About Sara:

Sara is a dynamic and youthful entrepreneur. Managing a restaurant takes up most of her time. Her goal is to draw in youthful clientele looking for a unique restaurant experience. At the same time she needs a system that is easy to maintain as she is not very technology savvy.

1.1.5 Persona 5 – Waiter/Waitress

Name: Jessica

Background:

- 17 years old, Female
- Student
- Part-time waitress

Goals:

- Provide efficient service for breakfast, lunch and dinner
- Easily manage orders
- Easily manage bills
- Easily identify needs of her customers

Frustrations:

- Difficulty catering to needs of all clients when restaurant is busy
- Difficulty in managing bills made by multiple individuals
- Difficulty in remembering specific requests associated to orders placed

About Jessica:

Jessica is a full-time high school student who works as a waitress part-time to make extra money for college. Her goals at work are to be able to serve customers in an efficient and polite manner. She would like to have a way to organize orders and bills so that nothing is forgotten. It is also very important to her to be able to know when a customer needs her help.

1.1.6 Persona 6 – Chef

Name: Sabrina

Background:

- 26 years old, Female
- Head chef

Goals:

- Provide efficient service for breakfast, lunch and dinner
- Easily manage orders

Frustrations:

- Difficulty managing special requests made by clients in regards to their order
- Lost order papers
- Difficulty in prioritizing orders that come in

About Sabrina:

Sabrina is a full time chef in a local Montreal restaurant. During busy peak hours it is often cumbersome to try and organize order processing by prioritization. It is also not always clear when a customer has special requests associated to their order such as allergies or dietary preferences. Often Sabrina has to deal with order slips getting lost in the busy kitchen traffic which results in unhappy customers.

1.2 Set of Tasks Performed

The set of tasks performed are based on the user types that exist within the TFF application (user and restaurant) and the use cases described in the SAD.

1.2.1 Create User Profile

Use Case Reference: UC 4.1

Before a user can access and use TFF it must first create a user profile. In order to do this the user will access the application through a mobile web browser or through the TFF application and Create an Account. On the sign up page the user will provide its information and will then tap on the Create Account button to finalize the user profile creation.

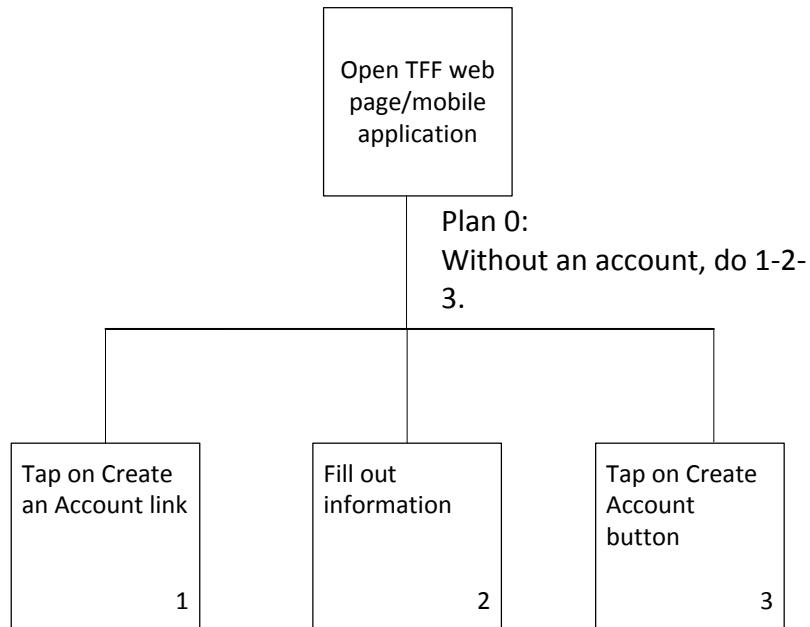


Figure 1-1 Create User Profile Hierarchical Task Model

1.2.2 Manage User Profile

Use Case Reference: UC 4.2 and 4.3

When a user has an existing account it has the ability to manage its personal profile. The user might want to change its password, update its email address, or add (or remove) a profile picture.

In order to perform these actions the user will need to have an existing account (explained in 1.2.1) and be logged in. Once logged in, the user will tap on the User link displayed in the main menu which will take it to the personal profile page. At the personal profile page the user will tap on the edit link, which will take it to the edit profile page. Once there the user may change its information and then tap on the save button in order to save the changes made.

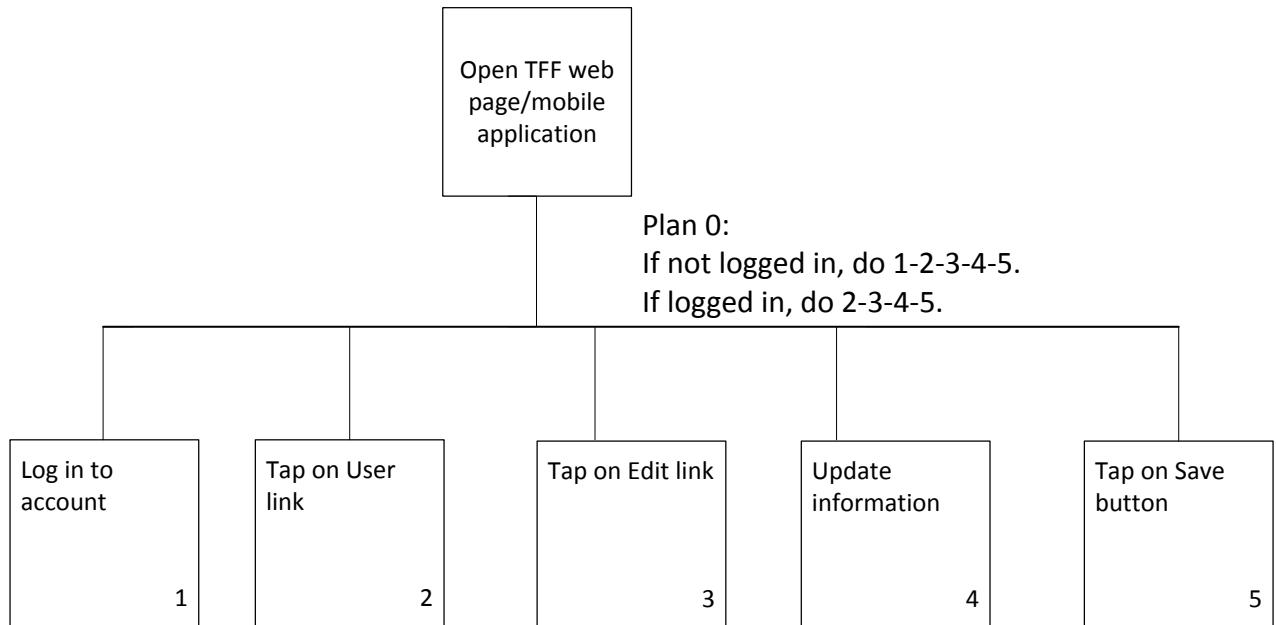


Figure 1-2 Manage User Profile Hierarchical Task Model

1.2.3 View Previous Orders/Reviews

Use Case Reference: UC 4.2, 4.4

When a user has previously used the application to successfully eat at a restaurant, this is recorded in the user's profile. This allows the user to review all previous activity and permits it to see the dishes that were consumed during that visit.

In order to achieve this the user must be logged in and have at least one complete visit at a restaurant having used TFF to place the order. On the main page the user will tap on the User link in order to view its profile. Once there the most recent past orders and past reviews can be seen.

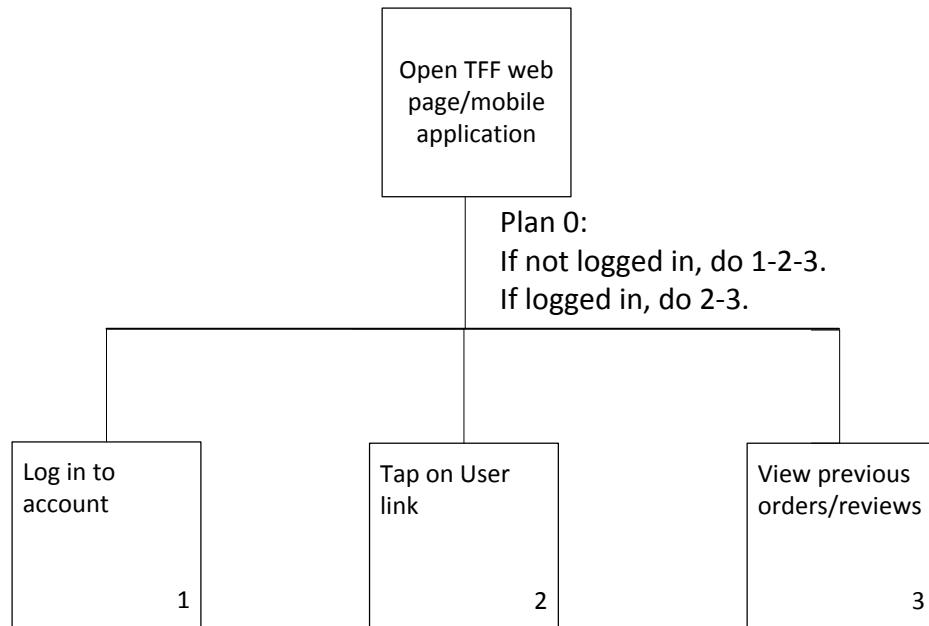


Figure 1-3 View Previous Orders/Reviews Hierarchical Task Model

1.2.4 Deliver Order Item

Use Case Reference: UC 2.2

When using TFF the restaurant will need to oversee the orders placed by the customers and apply appropriate actions to them such as declining them or accepting them. All of these actions are found when viewing a general list of order or when viewing the orders related to a specific table.

To achieve this objective the restaurant will access the TFF restaurant page and then tap on the Orders link. This will display a list of all the orders currently in place. If the restaurant wishes to view the orders of a specific table then it can tap on the table number and the orders for that table will be displayed. Once viewing the order desired, the desired action can be applied to the order.

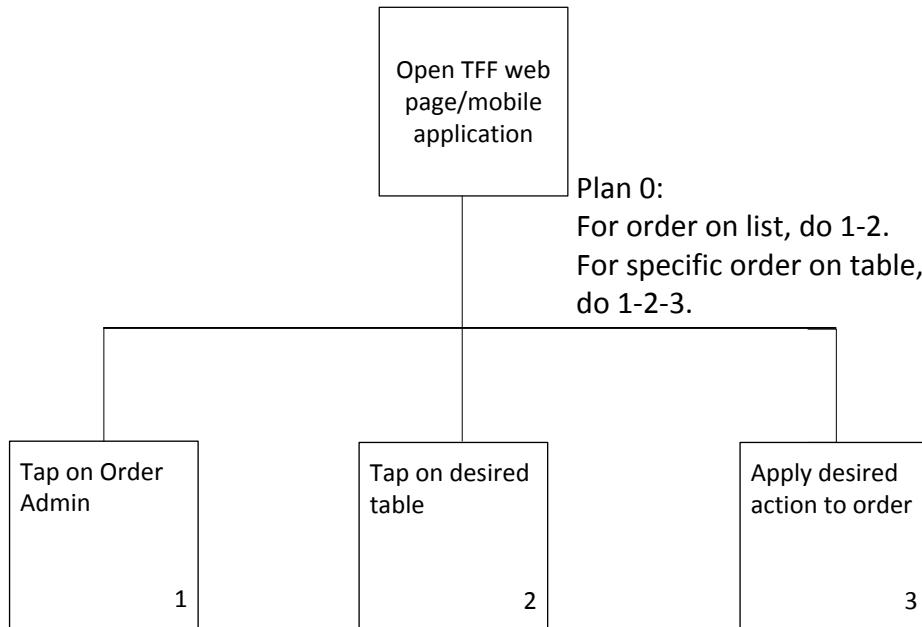


Figure 1-4 Deliver Order Item Hierarchical Task Model

1.2.5 View Menu

Use Case Reference: UC 1.1 and 4.2

Both the user and the restaurant will want to view the menus available. The user will view the menu in order to decide the items to add to its order while the restaurant will use the menus to display the items they have available for consumption.

In both situations the navigation is the same. The user/restaurant will log in to TFF (explained in 1.2.1) and click on the Menus link. From there the different menus will be shown. Tapping on a specific menu will show the items that belong to that menu.

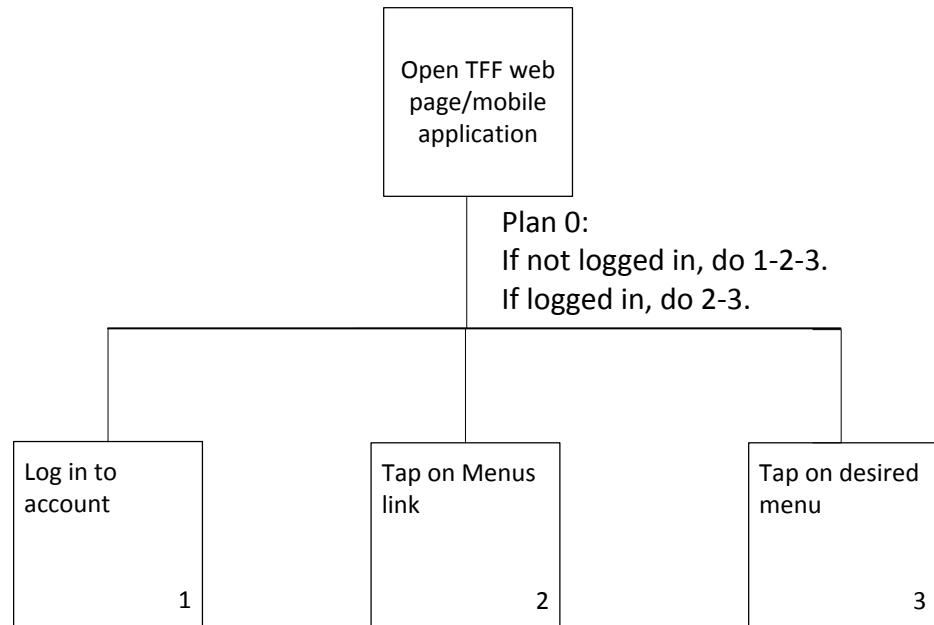


Figure 1-5 View Menu Hierarchical Task Model

1.2.6 Populate Menu

Use Case Reference: UC 3.1

When the restaurant is initially starting to use TFF it will need to populate at least one menu with items that will be offered for consumption. This area also allows the restaurant to activate or deactivate menus, such as having one available for lunch and another one for dinner.

In order to achieve this the restaurant will access the TFF main page and then tap on Menus. Once there the menus will be displayed and can be modified by clicking on one of them, or activated/deactivated by tapping on the checkmark next to it. When a menu is clicked the categories already in the menu as well as the ones available to add will be displayed. The restaurant will tap on the add button next to an available category to add it to the menu and on the remove button to remove the category from the menu. Once editing is finished the Back To List button will be tapped to return to the menu display.

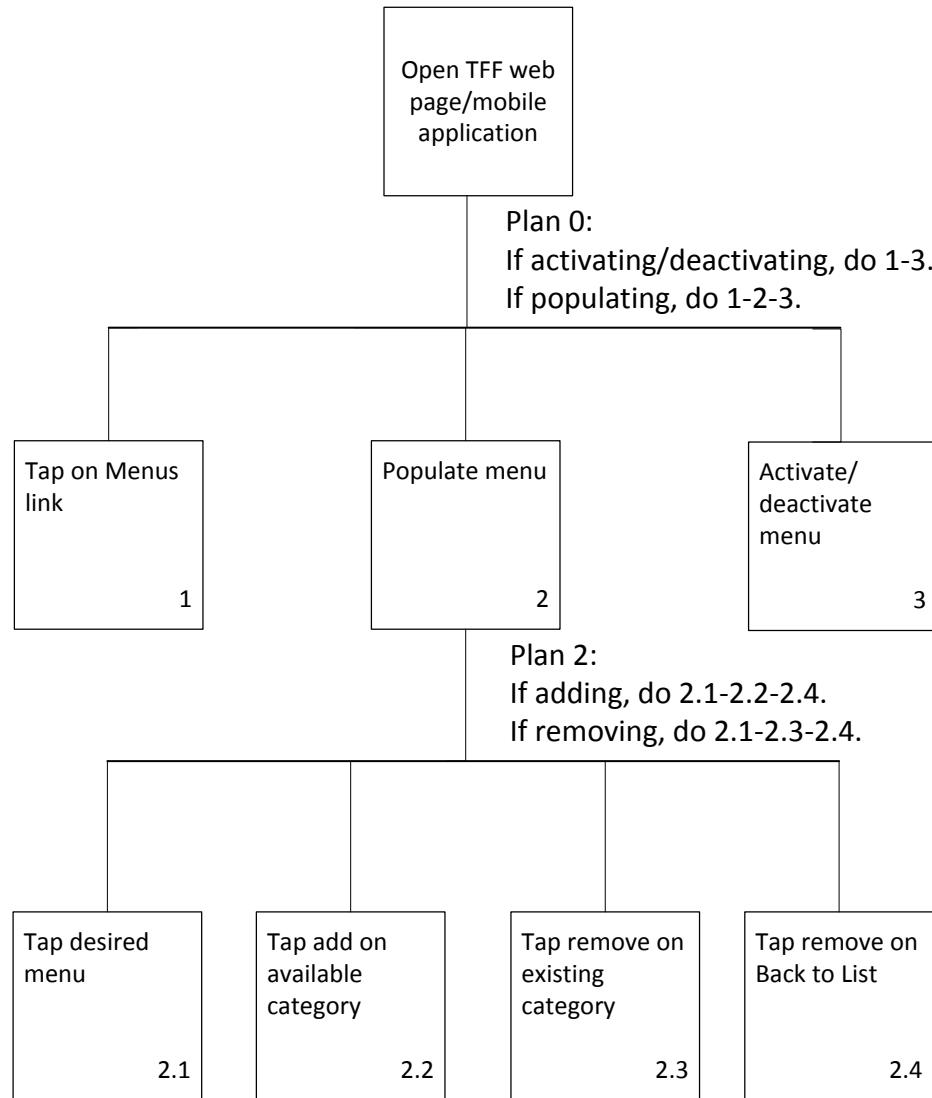


Figure 1-6 Populate Menu Hierarchical Task Model

1.2.7 Manage Tables

Use Case Reference: UC 5.1

When the restaurant is initially starting to use TFF it will need to have at least one table on its list of tables to manage. This area allows the restaurant to add or remove tables, view table details as well as viewing the NFC link associated to each table.

In order to achieve this the restaurant will access the TFF main page and then tap on Tables. Once there the tables will be displayed and the relevant information for the table can be seen by tapping on the Details button. The table may also be removed by tapping on the Delete button, and a new one may be created by tapping on the Create New button.

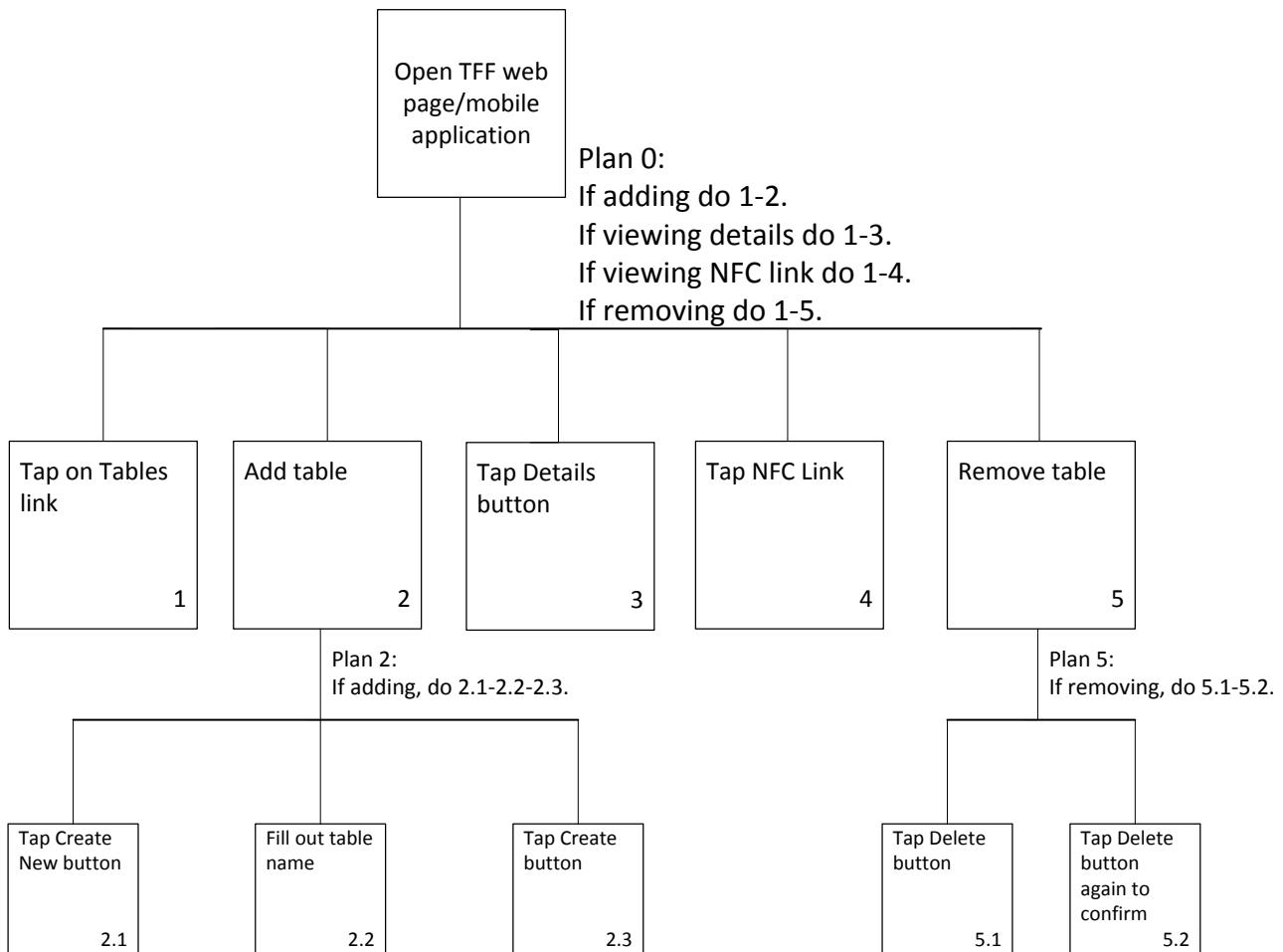


Figure 1-7 Manage Tables Hierarchical Task Model

1.2.8 Manage Bill

Use Case Reference: UC 8.1

It is a common occurrence for restaurant patrons to ask for their bills to be split up. In order to accommodate for this the functionality will be available directly from order page. The restaurant will tap on the manage bill page to access options for the bill such as splitting up or combining it with another one.

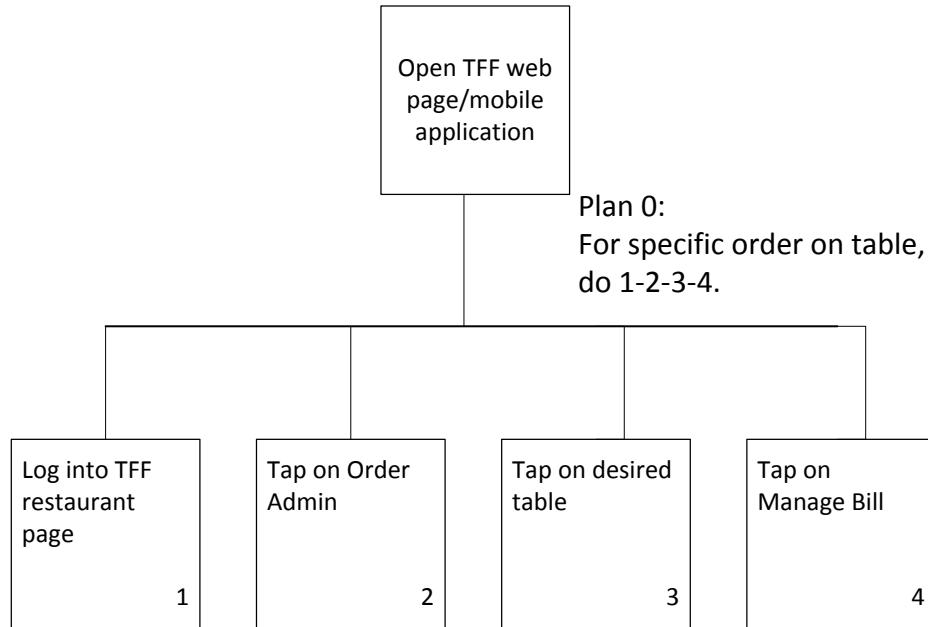


Figure 1-8 Manage Bill Hierarchical Task Model

1.2.9 Place Order

Use Case Reference: UC 2.1

The main functionality for a normal user is to place its food orders through TFF. A user will access the main TFF page and go view a specific menu. When a desired menu item is seen, the user can optionally tap on it and will be taken to the item description page. If everything is satisfactory then the user will return to the menu page and tap on the Add to Order button associated to that particular food item to add it to the current order. If more items are desired then the user may repeat the previous two steps to do so.

Once all desired items are in the order, then the user will tap on the Order button in order to view the order. The user will then tap the Finalize button in order to submit it to the restaurant for processing and delivery.

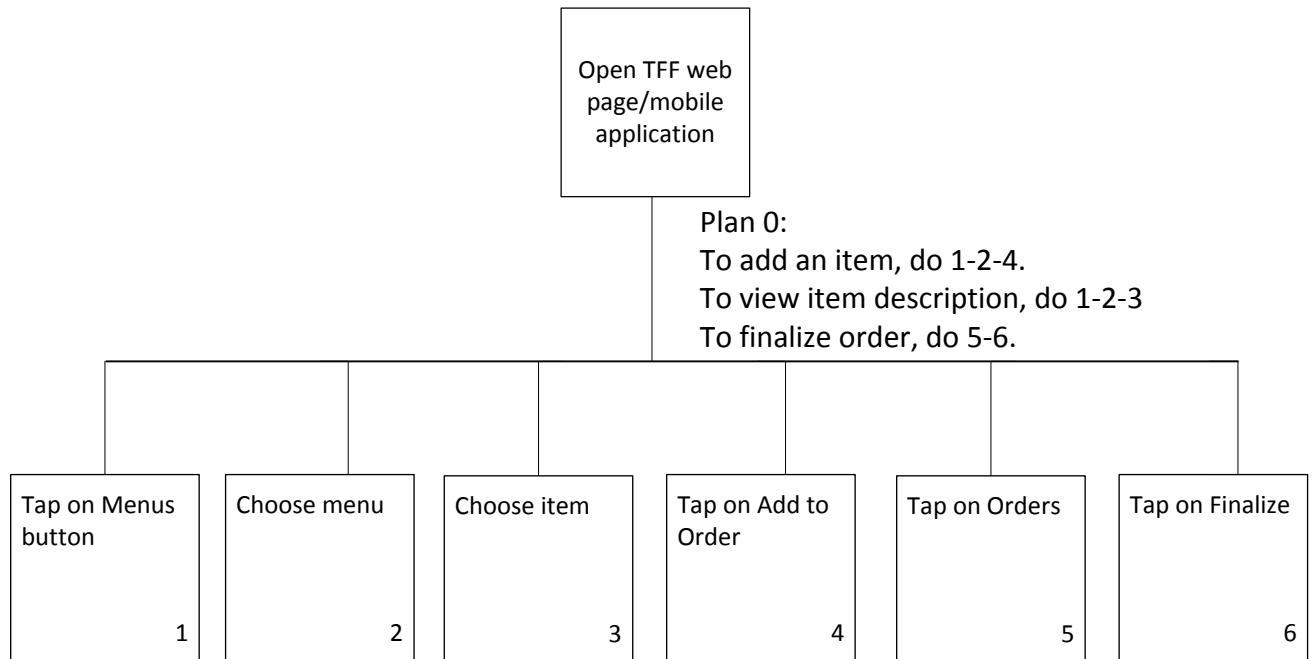


Figure 1-9 Place Order Hierarchical Task Model

1.2.10 Submit a Review

Use Case Reference: UC 6.1

A user, having used TFF to place and complete an order (explained in 1.2.9), might want to leave a review after doing so. The review page includes a rating dropdown to rate the user's experience from 1 to 5, as well as a text box for the user to leave more specifics about the experience.

In order to achieve this the user will have to place an order and successfully complete it. After completion, the user will be offered the option to leave a review related to the order that was just finished.

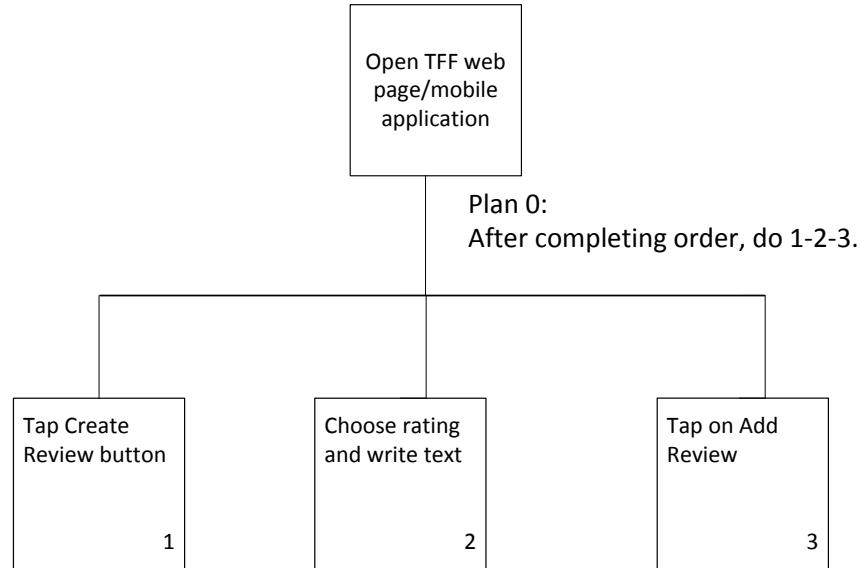


Figure 1-10 Submit a Review Hierarchical Task Model

1.2.11 Call a Waiter

Use Case Reference: UC 7.1

If the user needs a waiter then it can request one to come to the table directly from the TFF application. In order to do this the user will access TFF and tap on the Call Waiter button from the main menu.

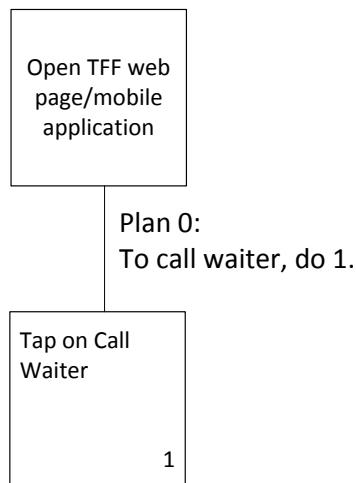


Figure 1-11 Call Waiter Hierarchical Task Model

1.3 Context of Use

The following contexts of use were derived from the goals defined in section 1.2, Set of Tasks Performed, of this document.

1.3.1 Context 1 – Create User Profile

Environmental Constraint	The user needs to have internet access on his/her mobile device.
Technical Constraint	Designed for phone and tablet
Social Constraint	The user may not want to give their personal information to create an account.
Location	At the restaurant where the user first finds out about Touch For Food (typically).
State of Mind	Relaxed and hungry
Other System	None
Tasks Performed*	4

1.3.2 Context 2 – Manage User Profile

Environmental Constraint	The user needs to have internet access on his/her mobile device.
Technical Constraint	Designed for phone and tablet
Social Constraint	NA
Location	Anywhere that conforms to the Environmental Constraint above
State of Mind	NA
Other System	None
Tasks Performed*	5

1.3.3 Context 3 – View Previous Orders/Reviews

Environmental Constraint	The user needs to have internet access on his/her mobile device.
Technical Constraint	Designed for phone and tablet
Social Constraint	NA
Location	Anywhere that conforms to the Environmental Constraint above
State of Mind	Curious
Other System	None
Tasks Performed*	3

1.3.4 Context 4 – Deliver Order Item

Environmental Constraint	The user needs to have internet access on his/her mobile device.
Technical Constraint	Designed for phone, tablet and PC
Social Constraint	NA
Location	A restaurant that uses the TFF application.
State of Mind	Rushed
Other System	None
Tasks Performed*	3

1.3.5 Context 5 – View Menu

Environmental Constraint	The user needs to have internet access on his/her mobile device.
Technical Constraint	Designed for phone
Social Constraint	Customers might not know what the items on the menu are.
Location	A restaurant that uses the TFF application.
State of Mind	Hungry
Other System	None
Tasks Performed*	3

1.3.6 Context 6 – Populate Menu

Environmental Constraint	The restaurant needs to have internet access.
Technical Constraint	Designed for tablet or PC
Social Constraint	Restaurant owners/managers might not be tech savvy.
Location	A restaurant that uses the TFF application.
State of Mind	Eager
Other System	None
Tasks Performed*	7

1.3.7 Context 7 – Manage Tables

Environmental Constraint	The restaurant needs to have internet access.
Technical Constraint	Designed for tablet or PC
Social Constraint	Restaurant owners/managers might not be tech savvy.
Location	A restaurant that uses the TFF application.
State of Mind	Eager
Other System	None
Tasks Performed*	10

1.3.8 Context 8 – Manage Bill

Environmental Constraint	The restaurant needs to have internet access.
Technical Constraint	Designed for phone and tablet
Social Constraint	Restaurant owners/managers/customers might not be tech savvy.
Location	A restaurant that uses the TFF application.
State of Mind	Impatient to leave
Other System	None
Tasks Performed*	4

1.3.9 Context 9 – Place Order

Environmental Constraint	The restaurant needs to have internet access.
Technical Constraint	Designed for phone and tablet
Social Constraint	Restaurant owners/managers/customers might not be tech savvy. The customers might not understand what the dishes are.
Location	A restaurant that uses the TFF application.
State of Mind	Hungry
Other System	None
Tasks Performed*	6

1.3.10 Context 10 – Submit a Review

Environmental Constraint	The user must have internet access on his/her mobile device.
Technical Constraint	Designed for phone, tablet and PC
Social Constraint	Restaurant owners/managers/customers might not be tech savvy. The customers might not understand what the dishes are.
Location	A restaurant that uses the TFF application.
State of Mind	Either satisfied with the service/food or upset with it.
Other System	None
Tasks Performed*	3

1.3.11 Context 11 – Call a Waiter

Environmental Constraint	The user must have internet access on his/her mobile device.
Technical Constraint	Designed for phone
Social Constraint	NA
Location	A restaurant that uses the TFF application.
State of Mind	Hungry/Impatient/Inquisitive
Other System	None
Tasks Performed*	1

1.4 Stakeholder Objectives

For information regarding stakeholder objectives, please refer to the Vision Document in Section 3.

2 UI evaluation

2.1 Overall site architecture

The overall site architecture of TFF is based on the feature groups outlined in the Vision Document. This design divides up system functions, clearly showing actions available to the user. This section of the document will detail UI specifications for coherence and comprehensibility.

2.1.1 Repeated UI Elements

The TFF web interface contains UI elements that are persistent on every page. These elements are the TFF logo, the user login status, the menu bar centered at the top and the footer at the bottom of the page, which includes the CloudNine copyright. This makes up the general look and feel of TFF.

The menu bar immediately displays the options available to the user. Features visible on the menu bar depends on the type of user that logs in. For regular users, the menu bar shows “Order” and “Menu”. Admin users have access to restaurant functionalities, which include “Menu Management”, “Order Management”, “Table Management” and “Bill Management”. For developers, they have access to all of the above including “Users”.

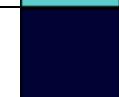
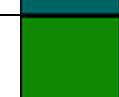
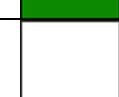
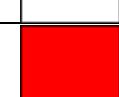
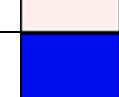
The repeated UI items are contained in the following file:

\TouchForFood\TouchForFood\Views\Shared_Layout.cshtml

This is a partial view that the MVC3 Razor framework applies to every page that a user can navigate to. If a user navigates to another page, it will have this partial view applied.

2.1.2 Colours

With the different UI aspects of TFF, a commonality shared between all pages is the colour scheme. Specifically, any text displayed uses the colour scheme. This helps the user find what they are looking for, but also allows the system to display success and error messages. The colour scheme that was picked has similar tones of green and blue [2]. The main colours of the colour scheme are:

Colour Code	Colour Sample	Usage
#269926		TFF Logo
#1D7373		Header
#5CCCCC		Management Instructions
#333		Button Hover Text
#006363		Hyperlinks
#0D8800		Logo Hover Text
#FFFFFF		Body background-color
#FF0000		Error Message Text
#FFEEEE		Error Message Background
#000EEE		Text colour on buttons

2.1.3 Similar Features

Regarding the look and feel of TFF, each section is responsible for a particular function. Many of the forms that a user fills out use the same template. For example, creating a new user and creating a new food item are two particular functions that have nothing to do with one another, but since the process is the same, the UI design for both forms are identical. This holds true for other aspects of the application, such as the buttons and font.

2.2 Navigation

Users of TFF will navigate around the system primarily with the menu located horizontally across the top of each page. The appearance of the top menu will change according to the user role; Meaning that the options or links displayed on the top menu will change according to the type of user logged into the system and their assumed end goals. The main menu will arrange itself to accommodate different screen sizes or window dimensions. The following image show an example of how the top menu will change according to who is logged in.

The figure consists of four horizontal screenshots of a web application interface, each representing a different user role:

- Client Role:** The top navigation bar includes a "Login" link on the right. Below it, a "Menus" link is underlined, indicating it is the active page.
- Restaurant Role:** The top navigation bar includes a "Welcome Client! [Log Off]" message on the right. Below it, a "Call Waiter!" link is underlined.
- Administrator Role:** The top navigation bar includes a "Welcome Restaurant! [Log Off]" message on the right. Below it, a "Service Request" link is underlined.
- Administrator Role (Detailed View):** This screenshot shows a more detailed navigation bar with links for "Users", "Restaurant", "Menus", "Categories", "Menu Category", "Item", "Order-Admin", "Order-User", "Table", and "Service Request".

Figure 2-1 TFF Top Menu According To User Role

TFF is a web application so users can also navigate around the system using the back, forward and address bar components in a web browser. Navigation may be slightly more limited when using TFF as a mobile application.

2.3 Feedback

The application provides appropriate feedback when performing various operation throughout the TFF system. Feedback is provided when adding items to an order, signing up for an account, and leaving a review for a particular restaurant.

2.3.1 Account Creation Feedback

An example of feedback received when attempting to create an account with invalid input data.

Sign Up

Please fill in the following fields to create a user account.

User Name	<input type="text" value="testuser"/>
Password	<input type="password" value="****"/>
The Password must be at least 6 characters long.	
Confirm Password	<input type="password"/>
The password and confirmation password do not match.	
First Name	<input type="text" value="test"/>
Last Name	<input type="text" value="user"/>
Email Address	<input type="text" value="test@test.com"/>
<input type="button" value="Create"/>	

Figure 2.3.1 Account Creation With Error

When an account is successfully created, the user receives feedback by seeing that he is now logged in with the newly created account.

TFF

Welcome **testuser!** [[Log Off](#)]

[Users](#) | [Restaurant](#) | [Menus](#) | [Categories](#) | [Menu Category](#) | [Item](#) | [Order-Admin](#) | [Order-User](#) | [Table](#)

Figure 2.3.2 Account Creation Successful

2.3.2 Popups

The TTF application also provides the user with information by the use of pop-ups. These pop-ups allow the user to confirm an operation. One example of such usage is when a user adds a menu item to his current order.

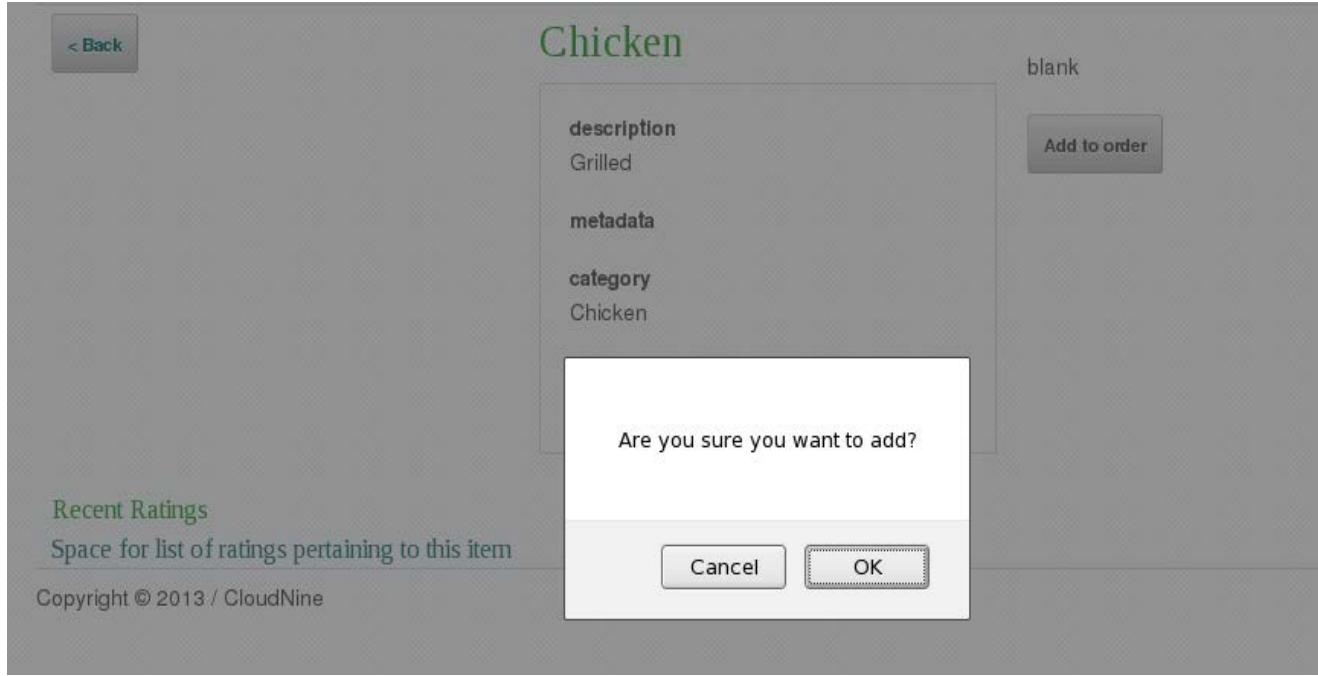


Figure 2.3.3 Adding Item To Order Confirmation

2.4 Screen layout

The screen layout for TFF is designed to be simple in order to provide easy navigation on a computer screen or a mobile device. The theme is designed to be optimized for mobile devices while still providing the same functionality on a monitor or screen. There are a few common layout schemes that are used that help provide a consistent look and feel to the user.

2.4.1 Restaurant Management Grids

Many of the pages that are available to restaurant managers contain functionality for content management system type operations. However, instead of using tables to display data, we use grids. One such example can be found in the category manager page.

Category Manager



Figure 2-2 Category Manager Grid

Like in all such grid pages, the user can update the selected item upon clicking a grid button. Creation and deletion can be done directly from the grid as seen in Figure 2-2. Grid buttons are themed with a dark blue border and font with a light green to green gradient.

2.4.2 Buttons or Icons

Buttons are themed to have a very light to light grey gradient. Button text is a dark grey. All buttons follow this color scheme except for buttons that are part of a management grid like in Figure 2-2.

Figure 2-3 shows an example of the buttons used to order an item. Figure 2-4 shows a create button which has the same theme as the order buttons except that it is slightly larger.

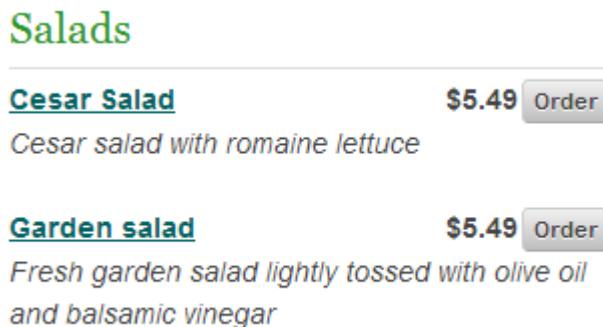


Figure 2-3 Order from Menu

2.4.3 Forms

Forms get used for creating or updating models from the content management system. They are also used for search and login. Form text inputs have a light grey border and grey text. If it is a create or update form, a grey box outlines it. Figure 2-4 shows an example of a create form.

Sign Up

Please fill in the following fields to create a user account.

User Name

Password

Confirm Password

First Name

Last Name

Email Address

Figure 2-4 User Creation

2.4.4 Overlays

Overlays are used to display information to a user without refreshing the page. One example of this is the menu item details page as seen in Figure 2-5.



Figure 2-5 Menu Details Overlay

Appendix A References

- [1] JAMPACK, "UIR," Concorida University, Montreal, Documentation 2012.
- [2] Petr Stanicek. (2010) Colr Scheme Designer. [Online]. <http://colorschemedesigner.com/#2P51Tw0w0w0w0>

Appendix B Glossary

Refer to the SRS document - Appendix B Glossary and Appendix C Acronyms for a complete list of terms and definitions.

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Traceability

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	Feature Definition	4
2	User Needs vs. Features	5
3	Use Cases vs. Features	6
4	Supplementary Requirements vs. Features	7
5	Use Cases vs. Test Cases	8

List of Tables

Table 1 - Feature Definition.....	4
Table 2 - User Needs vs. Features	5
Table 3 - Use Cases vs. Features	6
Table 4 - Supplementary Requirements vs. Features.....	7
Table 5 - Use Cases vs. Test Cases.....	8

Touch For Food

Traceability

Version 7.3

Revision History

1 Feature Definition

Feature Id	Feature Description
F1	Order Management
F2	Bill Management
F3	Food Reservation
F4	Table Reservation
F5	Assistance
F6	Review order history
F7	Manage Friends
F8	Social Networking
F9	Menu Management
F10	Rating System
F11	Customer Management
F12	Reporting System

Table 1 - Feature Definition

2 User Needs vs. Features

Needs vs. Features	Order Food (customer)	Make Reservation (customer)	Call Waiter (customer)	Manage Personal Page (customer)	View Menu (customer)	Leave Comment (customer)	View Restaurant stats & reviews	Accountability, identify priorities for customers	Manage Menu (restaurant)	Manage Tables (restaurant)	Manage bills (restaurant)	Restaurant Page	Restaurant reporting & stats
F1	X												X
F2	X											X	
F3		X											
F4		X								X			
F5			X										
F6				X									
F7				X									
F8				X		X	X						X
F9					X				X				X
F10						X	X						
F11								X					
F12													X

Table 2 - User Needs vs. Features

3 Use Cases vs. Features

Use Cases Vs. Features	UC 1.1	UC 2.1	UC 2.2	UC 3.1	UC 4.1	UC 4.2	UC 4.3	UC 4.4	UC 5.1	UC 6.1	UC 7.1	UC 8.1
F1	X	X	X									
F2												X
F3	X											
F4												
F5												X
F6												
F7												
F8					X	X	X	X				
F9				X								
F10												X
F11												
F12												

Table 3 - Use Cases vs. Features

4 Supplementary Requirements vs. Features

SRS vs. Features	SR 2.1	SR 2.2	SR 3.1	SR 3.2	SR 4.1	SR 5.1	SR 5.2	SR 5.3	SR 5.4	SR 5.5	SR 5.6	SR 5.7	SR 5.8	SR 5.9	SR 6.1	SR 6.2	SR 7.1	SR 7.2	SR 7.3	SR 7.4	SR 7.5	SR 7.6	SR 8	SR 9	SR 10.1	SR 10.2	SR 10.3	SR 10.4	SR 11.1	SR 11.2	SR 11.3	SR 11.4
F1	X	X	X		X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
F2	X	X	X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F3			X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F4		X	X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F5	X		X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F6	X		X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F7	X		X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F8	X	X	X		X				X	X	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F9		X		X	X					X			X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F10	X		X		X					X			X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F11				X	X					X			X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				
F12		X		X	X					X			X	X	X	X	X	X	X	X	X	X	X		X	X	X	X				

Table 4 - Supplementary Requirements vs. Features

5 Use Cases vs. Test Cases

Use Cases vs. Test Cases	UC 1.1	UC 2.1	UC 2.2	UC 3.1	UC 4.1	UC 4.2	UC 4.3	UC 4.4	UC C5 .1	UC 6.1	UC 7.1	UC 8.1
TC25.1		X										
TC26.1	X											
TC26.2	X											
TC26.3	X											
TC27.1						X						
TC27.2						X						
TC27.3						X						
TC27.4						X						
TC27.5						X						
TC29.1			X									
TC29.2		X										
TC29.3		X										
TC34.1										X		
TC34.2										X		
TC34.3										X		
TC34.4										X		
TC34.5										X		
TC34.6										X		
TC34.7										X		
TC40.1						X						
TC40.2				X								
TC40.3					X							
TC40.4												
TC42.1											X	
TC42.2											X	
TC42.3											X	
TC42.4											X	
TC192.1									X			
TC192.2									X			
TC192.3									X			
TC192.4									X			
TC192.5									X			
TC192.6									X			
TC192.7									X			

Table 5 - Use Cases vs. Test Cases

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Test Plan

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	Introduction.....	4
1.1	Purpose	4
1.2	Scope.....	4
1.3	Intended Audience	4
1.4	Document Terminology and Acronyms.....	4
1.5	References.....	4
2	Evaluation Mission and Test Motivation	5
2.1	Background.....	5
2.2	Evaluation Mission	5
2.3	Test Motivators	5
3	Target Test Items	6
3.1	Features to be Tested	6
3.2	Features Not to be Tested	6
4	Test Approach.....	7
4.1	Initial Test Idea Catalogues and Other Reference Sources.....	7
4.2	Testing Techniques and Types.....	7
5	Testing Workflow	11
5.1	Test Phase Table	11
5.2	Test Phase Gantt Chart	12
5.3	Test Management Tools.....	13
5.4	Test Slippage	13
5.5	Defects	13
	Appendix A References	14

Touch For Food

Test Plan

Version 7.16

Revision History

Date	Rev.	Description	Author(s)
2012-09-17	0.00	Document Creation	Katrina Anderson
2012-10-31	1.01	Added to sprint 1 and contributed section 2	Patrick Modafferri
2012-10-31	1.02	Contributed to section 4.1 and 4.2.1	Mikhail Levkovsky
2012-11-01	1.03	Contributed to Section 4.2.4	Matthew Tam
2012-11-01	1.04	Reviewed Section 4.2.1	Matthew Tam
2012-11-01	1.05	Contributed Section 1	Josh Hum
2012-11-02	1.06	Added sections 5.3, 5.4 & 5.5	Cristian Asenjo
2012-11-02	1.07	Added section 3	Ryan Nasr
2012-11-02	1.08	Added section 4.2.3 minor corrections to 4.2.4	Christian
2012-11-03	1.09	Added section 4.2.2	Cynthia Donato
2012-11-03	1.10	Added section 5.1 & 5.2	Katrina Anderson
2012-11-03	1.11	Corrected sections 5.1, 5.2, 5.3 & References. Formatted Document	Katrina Anderson
2012-11-04	1.12	Corrected section 4.2.3	Cynthia Donato
2012-12-16	3.13	Corrected Section 4.2, Section 5.5	Mikhail Levkovsky
2012-12-20	4.14	Reviewed document and reduced redundancies	Katrina Anderson
2013-02-09	7.15	Updated Section 3.1	Mikhail Levkovsky
2013-02-09	7.16	Reviewed and updated document for submission	Cynthia Donato

1 Introduction

1.1 Purpose

The purpose of this test plan is to ensure a structured and organized way of testing the TFF software. This document describes various techniques and approaches that will be used in the testing effort. This Test Plan will define both the items that must be tested as well as the criteria giving rise to a successful test.

The Test Plan for TFF will define four main concerns:

- Mission and motivation for testing
- List of features to be tested.
- List of features not to be tested
- Testing approach and techniques
- Testing workflow

The Test Plan will provide clear testing definitions and guidelines to ensure a quality product.

1.2 Scope

The TFF system will be tested using four approaches: Unit testing, user interface testing, usability testing and regression testing. For more details on each type of testing such as % coverage, success criteria, testing goals, etc., see *Section 4.2 Testing Techniques and Types*.

1.3 Intended Audience

The main audience of this test plan is the CloudNine team, who will all participate in the testing of TFF. Any supervisors and stakeholders should also be able to understand the basic plan.

1.4 Document Terminology and Acronyms

Refer to the SRS document - Appendix B Glossary and Appendix C Acronyms for a complete list of terms and definitions.

1.5 References

Please refer to Appendix A – References for a list of resources that may be referenced while preparing the Test Plan.

2 Evaluation Mission and Test Motivation

2.1 Background

In any software methodology testing is an important step. TFF is being developed using an iterative process, as such testing will be conducted each iteration. Testing frequently will increase code quality and mitigate risk. This section of the test plan will state the purpose of conducting testing activities and describe the reasoning behind the types of tests that were chosen for this project.

Refer to the Vision Document for more information regarding the history of TFF.

2.2 Evaluation Mission

The mission for the evaluation effort is to coordinate testing in a manner that reveals program defects, improves code quality and mitigates risk; specifically the risks involved with concurrent demands on the TFF software and security of sensitive client information.

2.3 Test Motivators

2.3.1 Risk Motivators

By testing early and often, bigger and more costly problems can be detected in early phase of development rather than later on in the project. The cost to repair a bug increases exponentially with respect to the lifecycle period in which it was created. Therefore, testing the functionalities iteratively, as they are being developed, will be a major factor in mitigating risk for the project.

2.3.2 Testing for Functionality and User Stories

The team will be following an agile methodology. In JIRA, an agile tracking tool, every feature is defined by a user story along with a set of subtasks. The team will add tests to these lists of tasks. By doing so, the test must be completed in order for the story to be considered closed or fixed. This will motivate our developers to test their code in order to be able to close the respective story.

2.3.3 Design and Customer Satisfaction

A major metric for success in the TFF application will be customer satisfaction and overall ease of use. Since our application will attempt to please a general audience, it will be crucial to perform usability tests on a carefully selected demographic namely restaurant employees for the version of the application that will be accessed through a web browser and the general public for the cell phone application. The motivation behind many of our tests will be to validate that the team is building a user friendly product that will appeal to both demographics described.

2.3.4 Summary

It is important to keep regular updates on the test plan in order to satisfy the previously mentioned goals and to remember why testing is necessary. Risk, traceability and customer satisfaction are factors that will motivate the team to test thoroughly in order to maintain quality.

3 Target Test Items

3.1 Features to be Tested

The high level list of features to be tested is described below. These features are all related to at least one user story. All tests associated to these features must pass successfully in order to close the user stories associated to them. In other words, a story can only be defined as ‘complete’ if its related tests all pass. These tests are not limited to a development workstation. They should also be run on TFF supported web capable mobile devices.

- Menu management
- Table management
- Order management
- Order food
- Reservations
- ~~Restaurant personal page~~
- Comment, rating and review system
- Customer management
- ~~Restaurant reporting and statistics~~
- Manage bills for restaurants
- Manage bills for customer
- The Output and Input Mappers (layer between the data store and the controllers)

3.2 Features Not to be Tested

Any auto-generated code from Visual Studio will not be tested. We will assume that these parts are already tested by the Visual Studio team; however, any changes made to the auto-generated code will be tested. In addition, there will not be any tests related to the operating system during development.

4 Test Approach

The testing for the TFF application will be done using four testing techniques: unit testing, user interface testing, usability testing and regression testing. This section of the test plan will indicate the objective, target items, testing goals, techniques, required tools, success criteria, failure contingencies and any special considerations involved with each testing methodology. For some tests, specific standards need to be followed in order to prevent false test results (i.e. different formatting in browsers) [1,2,3].

4.1 Initial Test Idea Catalogues and Other Reference Sources

Please refer to *Appendix A – References*, for the listing of existing resources that will be used to simulate the identification and selection of the specific tests to be conducted [4].

4.2 Testing Techniques and Types

4.2.1 Unit Testing

Technique Objective:	The objective of unit testing is to establish a basis for the successful completion of TFF functionalities. A user story can only be considered complete when its unit tests have successfully passed. Unit tests also ensure that new functionality does not break any existing ones.
Target Items	The features groups listed in Section 3.1 – Features to be Tested.
Testing Goal:	Unit tests must cover a minimum of 60% of the functionalities implemented by developers. Coverage will be measured by establishing the percent of statements included during the execution of unit tests.
Technique:	Unit tests will be written and executed using the Visual Studio IDE [5].
Oracles:	Visual Studio will provide a detailed report of the unit test results [5].
Required Tools:	Touch For Food – Access to the system code is required to run unit tests. Visual Studio 2010 IDE – Required for writing and running unit tests.
Success Criteria:	An individual unit test will “pass” if its programmed objective has been completed without any errors.
Failure contingencies:	A unit test will fail if the execution of its programmed objective results in an error. An error can either be a logical one (i.e. wrong data is displayed) or a runtime one (i.e. null pointer exception). A story cannot be closed until its Unit Test has passed (if applicable). The defect will be addressed in the current sprint if it is critical (i.e. inhibits progress of a task in the current sprint). If the defect is not critical it will be moved to the backlog to be re-evaluated during sprint planning.
Special Considerations:	Any auto generated code provided by Visual Studio will not be tested by the developers.

4.2.2 User Interface Testing

Technique Objective:	The objective of user interface testing is to exercise the event-driven nature of the application and log the respective results. It verifies that the user is able to execute each user story in an effective manner. UI testing will also test the access methods of TFF, such as mouse movement and tab keys to ensure that navigation is properly executed.
Target Items	The features groups listed in Section 3.1 – Features to be Tested.
Testing Goal:	100% coverage of the TFF screens
Technique:	Developers will write a minimum of one test case per user story. Each test case must then be manually executed by the developer to determine if it passes or fails.
Oracles:	After the manual execution of each test the tester will indicate if the test case has passed or failed in the test report.
Required Tools:	User Interface testing will be performed on multiple devices, such as desktops, laptops, tablets and smart phones. The restaurant management aspect of the application is meant to be viewed from a computer browser and will therefore be tested on Safari, Internet Explorer 9, Google Chrome and Mozilla Firefox browsers. Restaurant menus, order summaries, reviews and bill management are meant to be viewed from a cell phone application and will therefore be tested on Android, iOS and Windows phones.
Success Criteria:	A test will be considered as “passed” if the output of the manual test follows the flow of events indicated in the <i>Description</i> and <i>Expected Results</i> section of the respective test case being executed. No errors should appear at any of the steps.
Failure contingencies:	A test will be considered as “failed” if the output of the manual test does not follow the flow of events indicated in the <i>Description</i> and <i>Expected Results</i> section of the respective test case being executed. In addition, if any errors should appear during the execution of a test, the test is considered to have failed.
Special Considerations:	Not all properties of the TFF can be accessed or tested through UI testing on a particular device (cell phone vs. browser). It is important to distinguish which scenario is associated to each particular device in order to provide efficient testing. The test case should specify the <i>Target Platform</i> .

4.2.3 Usability Testing

Technique Objective:	The objective of usability testing is to evaluate the level of ease of learning and using the TFF interface. Usability tests evaluate whether the TFF application meets the expectations of the user and has an efficient user interface.
Target Items	The feature groups listed in Section 3.1 – Features to be tested
Testing Goal:	Usability tests are to be run by at least 6 people who are not involved in the development cycle of the application and who represent the two user groups, restaurant staff and restaurant clientele, who would use TFF. 100% Coverage of the TFF Screens
Technique:	Testers will create a number of fully dressed test cases that will be performed by the users. The users will perform the tests while being observed by the testers. Testers will record essential information (i.e. task duration), and record the results in the test report.
Oracles:	Direct observation will be used to document the performance of the usability tests.
Required Tools:	Touch For Food – Test requires system to be running Web Browser(Chrome, IE, Firefox, Safari) – Required to run usability testing of the web version Web Capable Mobile Devices (Cell phones and Tablets) – Required to run Usability testing of the mobile version
Success Criteria:	A usability test will “pass” if all users (testers and stakeholders) determine that the outcomes listed in the <i>Expected Results</i> sections of a test case are executed satisfactorily.
Failure contingencies:	An acceptance test will fail if either a tester or a stakeholder has determined that it has not completed the <i>Expected Results</i> criteria satisfactorily. Any problems found shall be added as defects in JIRA. Any critical errors found, shall be addressed in the current iteration, all other defects will be placed in the backlog.
Special Considerations:	It is important that the test cases indicate on which devices and browsers the tests should be executed.

4.2.4 Regression Testing

Technique Objective:	The objective of regression testing is to discover any bugs after implementing new features in the system. This testing verifies that any changes to the existing TFF system do not impact existing functionality.
Target Items	The feature groups listed in Section 3.1 – Features to be tested
Testing Goal:	60% for unit tests 100% for UI tests
Technique:	Unit regression testing will be executed with Visual Studio IDE [5]. UI regression testing will be done manually. Both programmed and manual tests will be executed at the end of each sprint.
Oracles:	For unit tests, Visual Studio will provide a detailed report of the unit test results [5]. For UI tests, the tester will indicate if the test has passed or failed in the test report.
Required Tools:	Touch For Food – Tests require system code to run Visual Studio 2010 IDE – Required to run unit tests Web Browser(Chrome, IE, Firefox, Safari) – Required to run UI tests Web Capable Mobile Devices (Smartphones Tablets) – Required to run UI tests
Success Criteria:	A regression test will “pass” if it successfully meets the criteria of its programmed and acceptance objectives. There should be no conflicts with the objectives. TFF can only be considered complete if it passes all regression tests.
Failure contingencies:	A regression test will “fail” if any of the criteria of its unit tests or UI tests fail. Failure of a test indicates that there is an error in the system. At the end of a sprint, all regression tests must pass for the sprint to be concluded successfully.
Special Considerations:	Some parts of the TFF may not be accessible for tests with Visual Studio

5 Testing Workflow

Section 5 of this test plan document will list the test phases of the TFF system. It will detail approximate testing time spans and illustrating them along with any dependencies in a Gantt chart. In addition, this document will also describe the management tools we plan to use during testing, as well as how we plan to handle test slippage and defects [6].

5.1 Test Phase Table

Test Phase ID	Phase Description	Phase Period
TP-01	Iteration 1: Run unit tests and fix bugs that are found. Update documents accordingly.	October 22 nd – November 2 nd
TP-02	Iteration 2: Run unit tests and fix bugs that are found. Update documents accordingly.	November 7 th – 20 th
TP-03	Iteration 3: Run unit tests and fix bugs that are found. Update documents accordingly.	December 4 th – 17 th
TP-04	Iteration 4: Run unit tests and fix bugs that are found. Update documents accordingly.	December 18 th – January 1 st
TP-05	Iteration 5: Run unit tests and fix bugs that are found. Update documents accordingly.	January 2 nd – 15 th
TP-06	Iteration 6: Run unit tests and fix bugs that are found. Update documents accordingly.	January 16 th – 27 th
TP-07	Iteration 6: Run user interface tests and fix bugs that are found. Update documents accordingly	January 28 th – 29 th
TP-08	Iteration 7: Run unit tests and fix bugs that are found. Update documents accordingly.	January 30 th – February 10 th
TP-09	Iteration 7: Run user interface tests and fix bugs that are found. Update documents accordingly	February 11 th – 12 th
TP-10	Iteration 8: Run unit tests and fix bugs that are found. Update documents accordingly.	February 13 th – 24 th
TP-11	Iteration 8: Run user interface tests and fix bugs that are found. Update documents accordingly	February 25 th – 26 th
TP-12	Iteration 8: Run usability tests for web application and fix bugs that are found. Update documents accordingly	February 25 th – 26 th
TP-13	Iteration 9: Run unit tests and fix bugs that are found. Update documents accordingly.	February 27 th – March 10 th
TP-14	Iteration 9: Run user interface tests and fix bugs that are found. Update documents accordingly	March 11 th – 12 th
TP-15	Iteration 10: Run unit tests and fix bugs that are found. Update documents accordingly.	March 13 th – 24 th
TP-16	Iteration 10: Run user interface tests and fix bugs that are found. Update documents accordingly	March 25 th – 26 th
TP-17	Iteration 10: Run usability tests for phone application and fix bugs that are found. Update documents accordingly	March 25 th – 26 th
TP-18	Iteration 11: Run unit tests and fix bugs that are found. Update documents accordingly.	March 27 th - April 7 th
TP-19	Iteration 11: Run user interface tests and fix bugs that are found. Update documents accordingly	April 8 th – 9 th

5.2 Test Phase Gantt Chart

ID	Task Name	Start	Finish	Duration	Oct 2012		Nov 2012				Dec 2012				Jan 2013				Feb 2013				Mar 2013				
						10/28	11/4	11/11	11/18	11/25	12/2	12/9	12/16	12/23	12/30	1/6	1/13	1/20	1/27	2/3	2/10	2/17	2/24	3/3	3/10	3/17	3/24
1	TP-01	22/10/2012	02/11/2012	12d																							
2	TP-02	07/11/2012	20/11/2012	14d																							
3	TP-03	04/12/2012	17/12/2012	14d																							
4	TP-04	18/12/2012	01/01/2013	15d																							
5	TP-05	02/01/2013	15/01/2013	14d																							
6	TP-06	16/01/2013	27/01/2013	12d																							
7	TP-07	28/01/2013	29/01/2013	2d																							
8	TP-08	30/01/2013	10/02/2013	12d																							
9	TP-09	11/02/2013	12/02/2013	2d																							
10	TP-10	13/02/2013	24/02/2013	12d																							
11	TP-11	25/02/2013	26/02/2013	2d																							
12	TP-12	25/02/2013	26/02/2013	2d																							
13	TP-13	27/02/2013	10/03/2013	12d																							
14	TP-14	11/03/2013	12/03/2013	2d																							
15	TP-15	13/03/2013	24/03/2013	12d																							
16	TP-16	25/03/2013	26/03/2013	2d																							
17	TP-17	25/03/2013	26/03/2013	2d																							
18	TP-18	27/03/2013	07/04/2013	12d																							
19	TP-19	08/04/2013	09/04/2013	2d																							

5.3 Test Management Tools

Three tools will be used to organize and document the testing activities for this project. The central tool we will be using to manage this project will be JIRA; with it, we will create test and directly associate them to their respective user stories. The creation and execution of the tests will be assigned to group members and can be viewed in a to-do style list format. The documentation of traceability matrices, User Stories vs. Use Cases and Test Cases vs. Use Cases, will be done using Microsoft Excel. Microsoft Excel will also be used to document the results of our test execution. Finally, Microsoft Word will be used to back up the data obtained from JIRA.

5.4 Test Slippage

Test slippage is known as the result of testers being unable to meet testing goals (usually due to time constraints). The TFF testing workflow will help see if test slippage has occurred. If slippage has occurred, then it is the responsibility of the developers and testers to report to the stakeholders, in order to establish which defects are tolerable or to reduce the set of features, so as to be able to deliver a viable system [6].

5.5 Defects

A defect is defined as improper behaviour in the application caused by either developer error or incorrect workflow. Defects may or may not terminate program execution depending on severity. Defect can cause any of the following: logical inconsistencies, crashes, unwanted side effect. Any defects encountered during the project will be documented in JIRA. Critical defects should be addressed during the testing phase of the current sprint and all other defects are placed in the backlog to be addressed in future sprints. The amount of defects will be used to establish quality metrics [6].

Appendix A References

- [1] Microsoft. (2012) ASP.NET MVC 3 Testing. [Online]. http://msdn.microsoft.com/en-us/vs2010trainingcourse_aspnetmvc3testing.aspx
- [2] The jQuery Foundation. (2012) jQuery; Write less, do more. [Online]. <http://jquery.com>
- [3] W3C. (2012) HTML & CSS. [Online]. <http://www.w3.org/standards/webdesign/htmlcss.html>
- [4] Systeme Evolutif Limited. (2012) TEST PLAN OUTLINE (IEEE 829 FORMAT). [Online]. <http://wwwcomputing.dcu.ie/~davids/courses/CA267/ieee829mtp.pdf>
- [5] Microsoft. (2012) Visual Studio 2010. [Online]. [http://msdn.microsoft.com/en-us/library/dd831853\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/dd831853(v=vs.100).aspx)
- [6] K. Anderson, C. Donato, J.Hum, M. Levkovsky, A. Lloyd, P. Modaffer, "*Testing Plan, Cases & Reports*". Montreal, QC: Concordia University, 2012.

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Test Cases & Reports

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	Test Cases	6
1.1	User Interface Test Cases	6
1.2	Usability Test Cases.....	23
2	Iteration 2 Test Results	24
2.1	Unit Tests.....	24
2.2	Defects	25
3	Iteration 3 Test Results	26
3.1	Unit Tests.....	26
3.2	Defects	28
4	Iteration 4 Test Results	29
4.1	Unit Tests.....	29
4.2	Defects	31
5	Iteration 5 Test Results	33
5.1	Unit Tests.....	33
5.2	Defects	35
6	Iteration 6 Test Results	38
6.1	Units Tests	38
6.2	Defects	41
6.3	User Interface Tests & Results	42
7	Iteration 7 Test Results	44
7.1	Units Tests	44
7.2	Defects	47
7.3	User Interface Tests & Results	51
Appendix A	References.....	53
Appendix B	Glossary	54

List of Figures

Figure 2-1 Iteration 2 Unit Test Results	24
Figure 2-2 Iteration 2 Unit Test Coverage Results	24
Figure 2-3 Iteration 2 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality	26
Figure 3-1 Iteration 3 Unit Test Results	26
Figure 3-2 Iteration 3 Unit Test Coverage Results	27
Figure 3-3 Iteration 3 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality	29
Figure 4-1 Iteration 4 Unit Test Results	29
Figure 4-2 Iteration 4 Code Coverage Results.....	30
Figure 4-3 Iteration 4 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality	32
Figure 5-1 Iteration 5 Unit Test Results	33
Figure 5-2 Iteration 5 Code Coverage Results.....	34
Figure 5-3Iteration 5 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality	37
Figure 6-1 Iteration 6 Unit Test Results	39
Figure 6-2 Iteration 6 Code Coverage Results.....	40
Figure 6-3 Iteration 6 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality	42
Figure 7-2 Iteration 7 Unit Test Results	45
Figure 7-4 Iteration 7 Code Coverage Results.....	46
Figure 7-5 Iteration 7 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality	50

List of Tables

Table 2-1 Iteration 2 List of Defects.....	25
Table 3-1 Iteration 3 List of Defects.....	28
Table 4-1 Iteration 4 List of Defects.....	31
Table 5-1 Iteration 5 List of Defects.....	35
Table 6-1 Iteration 6 List of Defects.....	41
Table 6-2 Iteration 6 UI Test Results.....	42
Table 7-1 Iteration 7 List of Defects.....	47
Table 7-2 Iteration 7 UI Test Results.....	51

Touch For Food

Test Report

Version 7.17

Revision History

Date	Rev.	Description	Author(s)
2013-01-11	5.0	Document Creation	Katrina Anderson
2013-01-13	5.1	Added unit test statistics for sprints 2-4 and added list of figures.	Katrina Anderson
2013-01-14	5.2	Added defect information for sprints 2-4 and added list of tables.	Katrina Anderson
2013-01-15	5.3	Added unit test and defect information for sprint 5	Katrina Anderson
2013-01-17	5.4	Updated table and stats	Katrina Anderson
2013-01-19	6.5	Added TC25.1	Cynthia Donato
2013-01-19	6.6	Added TC40.1-TC40.4	Katrina Anderson
2013-01-23	6.7	Added TC26.1-TC26.6	Christian Daher
2013-01-24	6.8	Added TC27.1-27.3	Mikhail Levkovsky
2013-01-26	6.9	Added TC192.1-192.8	Cristian Asenjo
2013-01-27	6.10	Added TC29.1-TC29.6	Ryan Nasr
2013-01-29	6.11	Fixed a few numbering typos in UI Test Cases	Matthew Tam
2013-01-29	6.12	Added User Interface Test & Results for sprint 6	Matthew Tam
2013-02-03	7.13	Added unit test and defect information for sprint 6	Josh Hum
2013-02-06	7.14	Added User Interface TC34 and TC42	Katrina Anderson
2013-02-07	7.15	Updated user interface tests to reflect system changes.	Katrina Anderson
2013-02-10	7.16	Added Test Report For Iteration 7	Katrina Anderson
2013-02-12	7.17	Reviewed Document	Josh Hum

1 Test Cases

1.1 User Interface Test Cases

TC25.1 Create Order

Test Case ID: TC25.1

Title: Test Create Order

Requirement: CAP-25

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Customer.
3. The user is currently on the menu page of restaurant R1.

Description:

1. The user selects a menu item and selects the “Add to Order” button.
2. The user navigates to the “Order” section of TFF.
3. The user presses the “Finalize” button to place the order.

Expected Results:

1. The system displays a message indicating that the item was added to the order.
2. The system displays the complete order. The order status is displayed as “EDITING”.
3. The system displays the order with the status of “PLACED”. The “Edit Order” and “View Order Status” buttons are visible.

TC26.1 Test View Menu (Customer)

Test Case ID: TC26.1

Title: Test View Menu (Customer)

Requirement: CAP-26 View Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Customer.
3. The menu M1 has been created.
4. The menu item I1 has been created.

Description:

1. The user selects the “Menus” option.
2. The user selects a menu M1.
3. The user selects a menu item I1.

Expected Results:

1. The system displays a list of menus for the restaurant the user is signed into.
2. The system displays various categories. Each category has at least one food item listed under it and all food items are displayed with a title, description, price and option to order.
3. The system enlarges the description for I1. The description includes title, description, price, photo and recent ratings.

TC26.2 Test View Menu (Restaurant)

Test Case ID: TC26.2

Title: Test View Menu (Restaurant)

Requirement: CAP-26 View Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. The menu M1 has been created.
4. The menu item I1 has been created.

Description:

1. The user selects the “Menus” option.
2. The user selects a menu M1.
3. The user selects a menu item I1.

Expected Results:

1. The system displays a list of menus for the restaurants associated to the user.
2. The system displays various categories. Each category has at least one food item listed under it and all food items are displayed with a title, description, and price.
3. The system enlarges the description for I1. The description includes title, description, price, photo and recent ratings.

TC26.3 Test View Menu (Administrator)

Test Case ID: TC26.3

Title: Test View Menu (Administrator)

Requirement: CAP-26 View Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Administrator.
3. The menu M1 has been created.
4. The menu item I1 has been created.

Description:

1. The user selects the “Menus” option.
2. The user selects a menu M1.
3. The user selects a menu item I1.

Expected Results:

1. The system displays a list of menus for all restaurants.
2. The system displays various categories. Each category has at least one food item listed under it and all food items are displayed with a title, description, price and option to order.
3. The system enlarges the description for I1. The description includes title, description, price, photo and recent ratings.

TC27.1 Test Menu Editor Main Page

Test Case ID: TC27.1

Title: Test Menu Editor Main Page

Requirement: CAP-27 Manage Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.

Description:

1. The user clicks on “Menus” link at the top of the page.

Expected Results:

1. The system displays a page where there is an option to create a new menu and all the current menus are displayed.

TC27.2 Test Category Management Page

Test Case ID: TC27.2

Title: Test Category Management Page

Requirement: CAP-27 Manage Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The menu M1 has been created (TC27.5)
4. The user has navigated to the menu M1 page.

Description:

1. The user clicks on “New Category”
2. The user fills in the form as follows and selects “Create”

Name: C1

Expected Results:

1. The system presents a form for the new category
2. The system displays a list of categories with the newly added category present

TC27.3 Test Item Management Page

Test Case ID: TC27.3

Title: Test Item Management Page

Requirement: CAP-27 Manage Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The category C1 has been created (TC27.3)
4. The user has navigated to the category C1 page.

Description:

1. The user clicks on the “Create Item” link.

2. The user fills in the form as follows and selects “Create”

Name: I1

Description: Yummy!

Price: 9.99

3. The user selects edit I1.

4. The user changes the fields as follows and presses “Save”

Name: I1

Description: Yummy In My Tummy!

Price: 99.99

Expected Results:

1. The system presents a form for the new item containing fields for name, description (optional), meta data (optional), a category and an image url (optional).
2. The system displays a list of items with the newly added item present
3. The system displays a form filled out with the item’s values
4. The system displays a list of items with the newly edited item present.

TC27.4 Test Menu Edit

Test Case ID: TC27.4

Title: Test Menu Edit

Requirement: CAP-27 Manage Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The menu M1 has been created (TC27.5)
4. The category C1 has been created (TC27.3)
5. The user has navigated to the “M1” page.

Description:

1. The user presses on the “+” next to C1 on the “Available Categories” list.
2. The user presses “-” next to C1 on the “On the Menu” list

Expected Results:

1. The system displays C1 on the “On the Menu” list.
2. The system displays C1 on the “Available Categories” list.

TC27.5 Test Create Menu

Test Case ID: TC27.5

Title: Test Category Management Page

Requirement: CAP-27 Manage Menu

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The restaurant R1 has been created.

Description:

1. The user clicks on “Menus” link at the top of the page.
2. The user clicks on “Create New Menu”
3. The user fills in the form as follows and selects “Create”
 Restaurant: R1
 Menu Name: M1
 Active: Checked

Expected Results:

1. The system displays a page where there will be an option to create a new menu.
2. The system displays a form to the user to put in the category name.
3. The system displays a page with an empty box entitled “On the Menu” and a second box filled with available categories and the option to create more categories.

TC29.1 Test Cancel Order

Test Case ID: TC29.1

Title: Test Cancel Order

Requirement: CAP-29 Manage Order

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The order O1 has been created.

Description:

1. The user navigates to the “Manage Order” page
2. The user clicks ‘Cancel’ on the order O1.
3. The user confirms the cancellation of O1.

Expected Results:

1. The system displays a list of orders. The status of O1 is set to “PLACED”
2. The system displays a confirmation message.
3. The system displays a list of orders. The status of O1 is set to “CANCELLED”

TC29.2 Test Decline Order Item

Test Case ID: TC29.2

Title: Test Decline Order Item

Requirement: CAP-29 Manage Order

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The order O1 has been created.

Description:

1. The user navigates to the “Manage Order” page
2. The user clicks ‘Decline’ on the order O1.
3. The user confirms the declination of O1.

Expected Results:

1. The system displays a list of orders. The status of O1 is set to “PLACED”
2. The system displays a confirmation message.
3. The system displays a list of orders that doesn’t include O1.

TC29.3 Test Accept Order Item

Test Case ID: TC29.3

Title: Test Accept Order Item

Requirement: CAP-29 Manage Order

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The order O1 has been created.

Description:

1. The user navigates to the “Manage Order” page
2. The user clicks ‘Accept’ on the order O1.

Expected Results:

1. The system displays a list of orders. The status of O1 is set to “PLACED”
2. The system displays a confirmation message.
3. The system displays a list of orders. Order items are set to processing

TC34.1 Test Create Service Request (Administrator)

Test Case ID: TC34.1

Title: Test Create Service Request (Administrator)

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Administrator.
3. An open request for service at Table 1 doesn't exist.

Description:

1. The user selects the "Service Request" link from the main menu.
2. The user selects the "Create Service Request" link.
3. The user fills in the form as follows and selects "Create"

Table: Table 1

Comment: "Water Please!"

4. The user selects the "Create Service Request" link.
5. The user fills in the form as follows and selects "Create"

Table: Table 1

Comment: "Water Please!"

Expected Results:

1. The system displays a list of all service requests (for all restaurants with all statuses).
2. The system displays the service request creation form.
3. The system displays a list of all service requests (for all restaurants with all statuses), including the new request for table 1.
4. The system displays the service request creation form.
5. The system displays an error message indicating that a request for table 1 is already open.

TC34.2 Test Create Service Request (Restaurant)

Test Case ID: TC34.2

Title: Test Create Service Request (Restaurant)

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An open request for service at Table 1 doesn't exist.

Description:

1. The user selects the "Service Request" link from the main menu.
2. The user selects the "Create Service Request" link.
3. The user fills in the form as follows and selects "Create"

Table: Table 1

Comment: "Water Please!"

4. The user selects the "Create Service Request" link.
5. The user fills in the form as follows and selects "Create"

Table: Table 1

Comment: "Water Please!"

Expected Results:

1. The system displays a list of open service requests for the user's restaurant.
2. The system displays the service request creation form.
3. The system displays a list of open service requests for the user's restaurant, including the new request for table 1.
4. The system displays the service request creation form.
5. The system displays an error message indicating that a request for table 1 is already open.

TC34.3 Test Call Waiter (Customer)

Test Case ID: TC34.3

Title: Test Call Waiter (Customer)

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Customer.
3. An open request for service at the user's table doesn't exist.

Description:

1. The user selects the "Call Waiter!" link from the main menu.
2. The user fills in the form as follows and selects "Call Waiter!"
Comment: "Water Please!"
3. The user selects the "Call Waiter!" link from the main menu
4. The user fills in the form as follows and selects "Call Waiter!"
Comment: "Water Please!"

Expected Results:

1. The system displays the call waiter form.
2. The system displays the TFF home page.
3. The system displays the call waiter form.
4. The system displays an error message indicating that a request for the user's table is already open.

TC34.4 Test Edit Service Request (Restaurant)

Test Case ID: TC34.4

Title: Test Edit Service Request (Restaurant)

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An open request for service at Table 1 already exists (TC34.1).
4. An open request for service at Table 2 doesn't exist.

Description:

1. The user selects the "Service Request" link from the main menu.
2. The user selects the "Edit" link on the service request for table 1.
3. The user changes the form as follows and selects "Save"

Table: Table 2

Comment: "Water Please! Please!"

Expected Results:

1. The system displays a list of open service requests for the user's restaurant.
2. The system displays the service request edit form.
3. The system displays a list of open service requests for the user's restaurant, including the changes made to the request for table 1.

TC34.5 Test Edit Service Request (Administrator)

Test Case ID: TC34.5

Title: Test Edit Service Request (Administrator)

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Administrator.
3. An open request for service at Table 1 already exists (TC34.1).
4. An open request for service at Table 2 doesn't exist.

Description:

1. The user selects the "Service Request" link from the main menu.
2. The user selects the "Edit" link on the service request for table 1.
3. The user changes the form as follows and selects "Save"

Create: 04/02/2013 8:44:49 PM

Status: CLOSED

Table: Table 2

Comment: "Water Please! Please!"

Expected Results:

1. The system displays a list of all service requests (for all restaurants with all statuses).
2. The system displays the service request edit form.
3. The system displays a list of all service requests (for all restaurants with all statuses), including the changes made to the request for table 1.

TC34.6 Test Cancel Service Request

Test Case ID: TC34.6

Title: Test Cancel Service Request

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An open request for service at Table 1 already exists (TC34.1).

Description:

1. The user selects the “Service Request” link from the main menu.
2. The user selects the “Cancel” link on the service request for table 1.
3. The user selects the “Cancel Request” button.

Expected Results:

1. The system displays a list of open service requests for the user’s restaurant.
2. The system requests a confirmation for cancelling the service request.
3. The system displays a list of open service requests for the user’s restaurant. A request for table 1 should not be included.

TC34.7 Test Close Service Request

Test Case ID: TC34.6

Title: Test Close Service Request

Requirement: CAP-34 Call Waiter

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An open request for service at Table 1 already exists (TC34.1).

Description:

1. The user selects the “Service Request” link from the main menu.
2. The user selects the “Close” link on the service request for table 1.
3. The user selects the “Close Request” button.

Expected Results:

1. The system displays a list of open service requests for the user’s restaurant.
2. The system requests a confirmation for closing the service request.
3. The system displays a list of open service requests for the user’s restaurant. A request for table 1 should not be included.

TC40.1 Test Login with Non-Existent User

Test Case ID: TC40.1

Title: Test Login With Non-Existent User

Requirement: CAP-40 Sign in

Type: UI

Preconditions:

1. The user is not already a member of TFF.
2. The user has navigated to the login page.

Description:

1. The user enters the following information and selects “Login”

User Name: TestingUser

Password: Testing

Expected Results:

1. The system displays the “Login” form with the error message “The user name or password provided is incorrect.”

TC40.2 Test Create New User

Test Case ID: TC40.2

Title: Test Create New User

Requirement: CAP-40 Sign in

Type: UI

Preconditions:

1. The user is not already a member of TFF.
2. The user has navigated to the login page.
3. TC40.1 has passed.

Description:

1. The user selects to “Sign Up”
2. The user enters the following information and selects “Create”

User Name: TestingUser

Password: Testing

Confirm Password: Testing

First Name: Testing

Last Name: User

Email: TestingUser@email.com

Expected Results:

1. The system displays the “Sign Up” form.
2. The system displays the TFF home page and the welcome message “Welcome TestingUser!”

TC40.3 Test Login with Existent User

Test Case ID: TC40.3

Title: Test Login With Existent User

Requirement: CAP-40 Sign in

Type: UI

Preconditions:

1. The user is already a member of TFF.
2. The user has navigated to the login page.
3. TC40.1 & TC40.2 have passed.

Description:

1. The user enters the following information and selects “Login”

User Name: TestingUsers
Password: Testing

2. The user enters the following information and selects “Login”

User Name: TestingUser
Password: testING

3. The user enters the following information and selects “Login”

User Name: TestingUser
Password: Testing

Expected Results:

1. The system displays an error message indicating that the username is invalid.
2. The system displays an error message indicating that the password is invalid.
3. The system displays the TFF home page and the welcome message “Welcome TestingUser!”

TC40.4 Test Logout

Test Case ID: TC40.4

Title: Test Logout

Requirement: CAP-40 Sign in

Type: UI

Preconditions:

1. The user is already a member of TFF.
2. The user has logged in to the system.
3. TC40.1, TC40.2 & TC40.3 have passed.

Description:

1. The user selects the “Log off” option located at the top right of the screen.

Expected Results:

1. The system displays the TFF home page without the welcome message “Welcome TestingUser!”

TC42.1 Test Create Bill (Restaurant)

Test Case ID: TC42.1

Title: Test Create Bill (Restaurant)

Requirement: CAP-42 Restaurant Bill Management

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An order, containing items, has been placed and finalized.

Description:

1. The user navigates to the order administration page.
2. The user selects the “Manage Bills” option on an order.
3. The user selects the “Create New Bill” option.

Expected Results:

1. The system displays a list of orders for the user’s restaurant.
2. The system displays a list of items associated to the order and a button to “Create New Bill”
3. The system displays a list of items associated to the order, a button to “Create New Bill” and an empty bill with a Total of 0.00\$

TC42.2 Test Add Item to Bill (Restaurant)

Test Case ID: TC42.2

Title: Test Add Item To Bill (Restaurant)

Requirement: CAP-42 Restaurant Bill Management

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An order, containing items, has been placed and finalized.
4. An empty bill has been created. (TC42.1)

Description:

1. The user selects all the items associated to the order and adds them to the bill.

Expected Results:

1. The system displays an updated version of the bill, which contains all the items associated to the order as well as an appropriate total. The system also displays the “Create New Bill” button.

TC42.3 Test Remove Item from Bill (Restaurant)

Test Case ID: TC42.3

Title: Test Remove Item From Bill (Restaurant)

Requirement: CAP-42 Restaurant Bill Management

Type: UI

Preconditions:

5. The user is a member of TFF and is signed into the system.
6. The user is assigned the role of Restaurant.
7. An order, containing items, has been placed and finalized.
8. A bill, containing items, has been created. (TC42.2)

Description:

1. The user removes an item from the bill.

Expected Results:

1. The system displays an updated version of the bill, which doesn’t contain the removed item and has the price of the removed item decremented from the total. The system also displays the removed item as un-associated.

TC42.4 Test Delete Bill (Restaurant)

Test Case ID: TC42.4

Title: Test Delete Bill (Restaurant)

Requirement: CAP-42 Restaurant Bill Management

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant.
3. An order, containing items, has been placed and finalized.
4. A bill, containing items, has been created. (TC42.2)

Description:

1. The user selects the delete bill option

Expected Results:

1. The system displays all items previously listed on the bill as un-associated.

TC192.1 Test Create Table Page

Test Case ID: TC192.1

Title: Test Create Table Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks “Create New”
3. The user fills out the form as follows and selects “Create”

Name: T1

Expected Results:

1. The system displays a list of tables.
2. The system displays a form for creating a table
3. The system displays a list of tables including the newly created table.

TC192.2 Test Creating Table Page – Error Blank Fields

Test Case ID: TC192.2

Title: Test Creating Incorrect Table Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks “Create New”
3. The user does not fill out the form and selects “Create”

Expected Results:

1. The system displays a list of tables.
2. The system displays a form for creating a table
3. The system displays an error message indicating that the table information cannot be left blank.

TC192.3 Test Edit Table Page

Test Case ID: TC192.3

Title: Test Edit Table Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The table T1 has been created.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks on Edit for table T1.
3. The user fills out the form as follows and selects “Save”
Name: T2

Expected Results:

1. The system displays a list of tables.
2. The system displays a form for editing T1
3. The system displays a list of tables including the changes made in step 3.

TC192.4 Test Edit Table Page – Error Blank Fields

Test Case ID: TC192.4

Title: Test Incorrect Edit Table Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The table T1 has been created.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks on Edit for table T1.
3. The user fills out the form as follows and selects “Save”

Name:

Expected Results:

1. The system displays a list of tables.
2. The system displays a form for editing T1
3. The system displays an error message indicating that the table information cannot be left blank.

TC192.5 Test NFC Link Page

Test Case ID: TC192.5

Title: Test NFC Link Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The table T1 has been created.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks on NFC Link for T1

Expected Results:

1. The system displays a list of tables.
2. The system displays the information of the restaurant related to T1

TC192.6 Test Table Details Page

Test Case ID: TC192.6

Title: Test Table Details Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The table T1 has been created.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks on “Details” for T1

Expected Results:

1. The system displays a list of tables.
2. The application will show the information T1

TC192.7 Test Delete Table Page

Test Case ID: TC192.7

Title: Test Delete Table Page

Requirement: CAP-192 Manage Tables

Type: UI

Preconditions:

1. The user is a member of TFF and is signed into the system.
2. The user is assigned the role of Restaurant or Administrator.
3. The table T1 has been created.

Description:

1. The user clicks on “Tables” link at the top of the page.
2. The user clicks on “Delete” for T1
3. The user confirms the deletion of T1.

Expected Results:

1. The system displays a list of tables.
2. The system displays a confirmation message.
3. The system displays a list of tables. T1 is not on the list.

1.2 Usability Test Cases

2 Iteration 2 Test Results

2.1 Unit Tests

2.1.1 Unit Test Table & Status

Number of Unit Tests: 4

Number of Passed Unit Tests: 3

Number of Failed Unit Tests: 1

Test run completing... Results (Group By: Class Name): 3/4 passed; Item(s) checked: 1				
	Result	Test Name	Project	Error Message
HomeControllerTest	<input type="checkbox"/>  Passed	Index	TouchForFood.Tests	
UserControllerTest	<input checked="" type="checkbox"/>  Failed	CreateValidUser	TouchForFood.Tests	Test method TouchForFood.Tests.UserControllerTest.CreateValidUser threw exception: ...
	<input type="checkbox"/>  Passed	CreateView	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	Index	TouchForFood.Tests	

Figure 2-1 Iteration 2 Unit Test Results

2.1.2 Unit Test Coverage

Project Coverage: 6.10%

Contoller Layer Coverage: 5.5%

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Trinaa@TOSHI 2013-01-13 17:48:56	1123	93.90 %	73	6.10 %
TouchForFood.dll	1123	93.90 %	73	6.10 %
{} TouchForFood	18	100.00 %	0	0.00 %
MvcApplication	18	100.00 %	0	0.00 %
{} TouchForFood.Attributes	7	100.00 %	0	0.00 %
AjaxAttribute	7	100.00 %	0	0.00 %
{} TouchForFood.Controllers	1014	94.50 %	59	5.50 %
CategoryController	63	100.00 %	0	0.00 %
FriendshipController	164	100.00 %	0	0.00 %
HomeController	0	0.00 %	12	100.00 %
MenuController	142	100.00 %	0	0.00 %
Menu_CategoryController	173	100.00 %	0	0.00 %
OrderController	217	100.00 %	0	0.00 %
RestaurantController	68	100.00 %	0	0.00 %
ReviewController	55	100.00 %	0	0.00 %
UserController	132	73.74 %	47	26.26 %
{} TouchForFood.Models	56	80.00 %	14	20.00 %
CategoryFilterVM	10	100.00 %	0	0.00 %
category	6	100.00 %	0	0.00 %
item	6	100.00 %	0	0.00 %
menu	4	100.00 %	0	0.00 %
menu_category	4	100.00 %	0	0.00 %
order	6	100.00 %	0	0.00 %
order_status	4	100.00 %	0	0.00 %
restaurant	10	100.00 %	0	0.00 %
touch_for_foodEntities	2	50.00 %	2	50.00 %
user	0	0.00 %	12	100.00 %
waiter	4	100.00 %	0	0.00 %
{} TouchForFood.Util.Category	28	100.00 %	0	0.00 %
CategoryUtil	28	100.00 %	0	0.00 %

Figure 2-2 Iteration 2 Unit Test Coverage Results

2.2 Defects

Table 2-1 Iteration 2 List of Defects

Key	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Created	Resolved	Affects Sprint	Fix Sprint	Time Spent (hours)
CAP-45	Bug	Resolved	Major	Fixed	Josh Hum	Cristian Asenjo	10/20/2012 12:20	11/5/2012 11:26	Sprint 2	Sprint 2	2.33
CAP-53	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	10/20/2012 18:50	10/26/2012 14:28	Sprint 2	Sprint 2	2.00
CAP-54	Bug	Open	Minor	<i>Unresolved</i>	Christian Daher	Cristian Asenjo	10/25/2012 22:20		Sprint 2	Sprint 3	
CAP-55	Bug	Open	Minor	<i>Unresolved</i>	Christian Daher	Cristian Asenjo	10/25/2012 22:36		Sprint 2	Sprint 3	
CAP-77	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	11/6/2012 12:13	11/6/2012 14:18	Sprint 2	Sprint 2	0.50
CAP-82	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 11:22	11/10/2012 15:01	Sprint 2	Sprint 2	3.00
CAP-83	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 15:02	11/10/2012 18:36	Sprint 2	Sprint 2	2.00
CAP-85	Bug	Open	Critical	<i>Unresolved</i>	Ryan Nasr	Cristian Asenjo	11/14/2012 22:09		Sprint 2	Sprint 3	
CAP-86	Bug	Open	Minor	<i>Unresolved</i>	Ryan Nasr	Cristian Asenjo	11/14/2012 22:10		Sprint 2	Sprint 3	
CAP-88	Bug	Open	Major	<i>Unresolved</i>	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:53		Sprint 2	Sprint 3	
CAP-91	Bug	Open	Trivial	<i>Unresolved</i>	Cristian Asenjo	Patrick Modafferi	11/19/2012 20:55		Sprint 2	Sprint 3	
CAP-92	Bug	Open	Trivial	<i>Unresolved</i>	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:55		Sprint 2	Sprint 3	
CAP-93	Bug	Open	Trivial	<i>Unresolved</i>	Cristian Asenjo	Patrick Modafferi	11/19/2012 20:56		Sprint 2	Sprint 3	
CAP-94	Bug	Open	Trivial	<i>Unresolved</i>	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:56		Sprint 2	Sprint 3	
CAP-95	Bug	Open	Trivial	<i>Unresolved</i>	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:58		Sprint 2	Sprint 3	
CAP-96	Bug	Open	Trivial	<i>Unresolved</i>	Cynthia Donato	Patrick Modafferi	11/19/2012 20:58		Sprint 2	Sprint 3	
CAP-97	Bug	Open	Trivial	<i>Unresolved</i>	Mikhail Levkovsky	Patrick Modafferi	11/19/2012 21:00		Sprint 2	Sprint 3	
CAP-98	Bug	Open	Trivial	<i>Unresolved</i>	Josh Hum	Patrick Modafferi	11/19/2012 21:00		Sprint 2	Sprint 3	
CAP-99	Bug	Open	Trivial	<i>Unresolved</i>	Cristian Asenjo	Patrick Modafferi	11/19/2012 21:16		Sprint 2	Sprint 3	
CAP-100	Bug	Open	Minor	<i>Unresolved</i>	Katrina Anderson	Patrick Modafferi	11/19/2012 21:28		Sprint 2	Sprint 3	

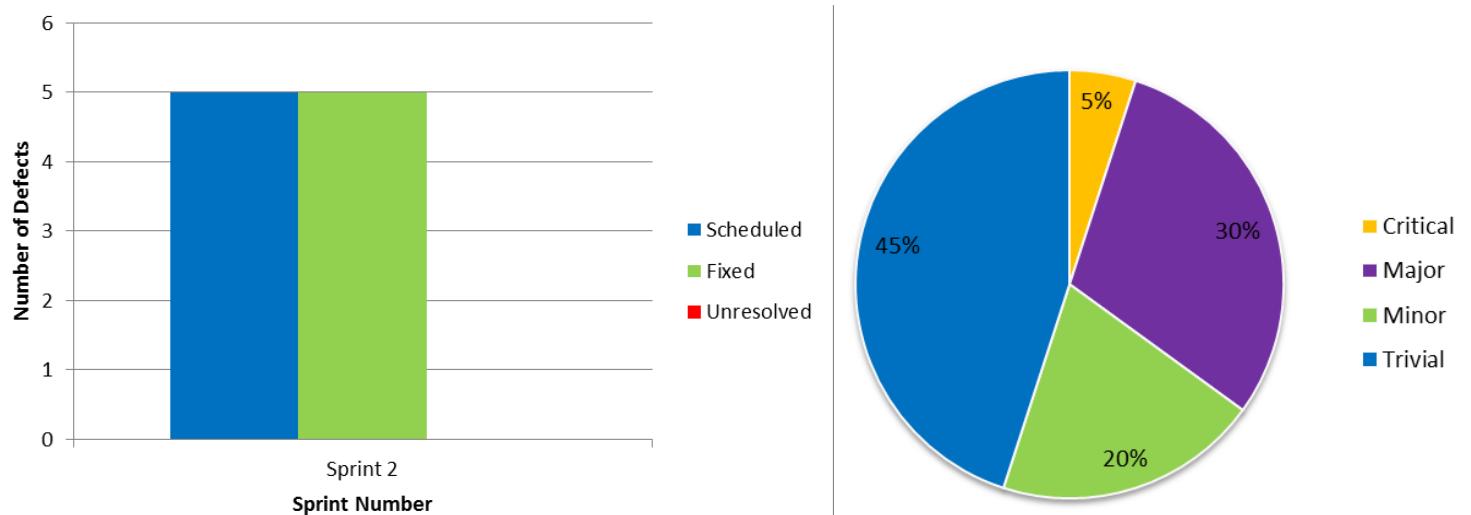


Figure 2-3 Iteration 2 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality

3 Iteration 3 Test Results

3.1 Unit Tests

3.1.1 Unit Test Table & Status

Number of Unit Tests: 4

Number of Passed Unit Tests: 3

Number of Failed Unit Tests: 1

⚠ Test run error Results (Group By: Class Name): 3/4 passed; Item(s) checked: 0			
	Result	Test Name	Project
HomeControllerTest	<input type="checkbox"/> Passed	Index	TouchForFood.Tests
UserControllerTest	<input type="checkbox"/> Failed	CreateValidUser	TouchForFood.Tests
	<input type="checkbox"/> Passed	CreateView	TouchForFood.Tests
	<input type="checkbox"/> Passed	Index	TouchForFood.Tests

Figure 3-1 Iteration 3 Unit Test Results

3.1.2 Unit Test Coverage

Project Coverage: 5.38%
Controller Layer Coverage: 4.78%

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Trinaa@TOSHI 2013-01-13 17:32:45	1284	94.62 %	73	5.38 %
TouchForFood.dll	1284	94.62 %	73	5.38 %
{} TouchForFood	18	100.00 %	0	0.00 %
MvcApplication	18	100.00 %	0	0.00 %
{} TouchForFood.Attributes	7	100.00 %	0	0.00 %
AjaxAttribute	7	100.00 %	0	0.00 %
{} TouchForFood.Controllers	1175	95.22 %	59	4.78 %
CategoryController	63	100.00 %	0	0.00 %
FriendshipController	164	100.00 %	0	0.00 %
HomeController	0	0.00 %	12	100.00 %
ItemController	116	100.00 %	0	0.00 %
MenuController	142	100.00 %	0	0.00 %
Menu_CategoryController	173	100.00 %	0	0.00 %
OrderController	217	100.00 %	0	0.00 %
RestaurantController	68	100.00 %	0	0.00 %
ReviewController	88	100.00 %	0	0.00 %
UserController	144	75.39 %	47	24.61 %
{} TouchForFood.Models	56	80.00 %	14	20.00 %
CategoryFilterVM	10	100.00 %	0	0.00 %
category	6	100.00 %	0	0.00 %
item	6	100.00 %	0	0.00 %
menu	4	100.00 %	0	0.00 %
menu_category	4	100.00 %	0	0.00 %
order	6	100.00 %	0	0.00 %
order_status	4	100.00 %	0	0.00 %
restaurant	10	100.00 %	0	0.00 %
touch_for_foodEntities	2	50.00 %	2	50.00 %
user	0	0.00 %	12	100.00 %
waiter	4	100.00 %	0	0.00 %
{} TouchForFood.Util.Category	28	100.00 %	0	0.00 %
CategoryUtil	28	100.00 %	0	0.00 %

Figure 3-2 Iteration 3 Unit Test Coverage Results

3.2 Defects

Table 3-1 Iteration 3 List of Defects

Key	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Created	Resolved	Affects Sprint	Fix Sprint	Time Spent (hours)
CAP-45	Bug	Resolved	Major	Fixed	Josh Hum	Cristian Asenjo	10/20/2012 12:20	11/5/2012 11:26	Sprint 2	Sprint 2	2.33
CAP-53	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	10/20/2012 18:50	10/26/2012 14:28	Sprint 2	Sprint 2	2.00
CAP-54	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	10/25/2012 22:20	12/17/2012 16:02	Sprint 2	Sprint 3	0.17
CAP-55	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	10/25/2012 22:36	12/17/2012 16:00	Sprint 2	Sprint 3	0.08
CAP-77	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	11/6/2012 12:13	11/6/2012 14:18	Sprint 2	Sprint 2	0.50
CAP-82	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 11:22	11/10/2012 15:01	Sprint 2	Sprint 2	3.00
CAP-83	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 15:02	11/10/2012 18:36	Sprint 2	Sprint 2	2.00
CAP-85	Bug	Open	Critical	Unresolved	Ryan Nasr	Cristian Asenjo	11/14/2012 22:09		Sprint 2	Sprint 3	
CAP-86	Bug	Resolved	Minor	Fixed	Ryan Nasr	Cristian Asenjo	11/14/2012 22:10	12/17/2012 15:52	Sprint 2	Sprint 3	1.00
CAP-88	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:53	12/17/2012 16:03	Sprint 2	Sprint 3	1.00
CAP-91	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 20:55	12/17/2012 15:33	Sprint 2	Sprint 3	1.00
CAP-92	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:55	12/11/2012 17:00	Sprint 2	Sprint 3	0.50
CAP-93	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 20:56	12/16/2012 23:12	Sprint 2	Sprint 3	1.25
CAP-94	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:56	12/16/2012 19:25	Sprint 2	Sprint 3	0.50
CAP-95	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:58	12/17/2012 17:58	Sprint 2	Sprint 3	1.00
CAP-96	Bug	Resolved	Trivial	Fixed	Cynthia Donato	Patrick Modafferi	11/19/2012 20:58	12/17/2012 15:33	Sprint 2	Sprint 3	3.52
CAP-97	Bug	Resolved	Trivial	Fixed	Mikhail Levkovsky	Patrick Modafferi	11/19/2012 21:00	12/16/2012 17:20	Sprint 2	Sprint 3	0.50
CAP-98	Bug	Open	Trivial	Unresolved	Josh Hum	Patrick Modafferi	11/19/2012 21:00		Sprint 2	Sprint 3	
CAP-99	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 21:16	12/16/2012 23:11	Sprint 2	Sprint 3	0.33
CAP-100	Bug	Open	Minor	Unresolved	Katrina Anderson	Patrick Modafferi	11/19/2012 21:28		Sprint 2	Sprint 3	
CAP-101	Bug	Open	Minor	Unresolved	Christian Daher	Katrina Anderson	12/16/2012 22:12		Sprint 3	Sprint 4	
CAP-102	Bug	Resolved	Minor	Fixed	Katrina Anderson	Katrina Anderson	12/17/2012 0:09	12/17/2012 15:44	Sprint 3	Sprint 3	0.50
CAP-108	Bug	Open	Minor	Unresolved	Christian Daher	Christian Daher	12/17/2012 16:19		Sprint 3	Sprint 4	
CAP-109	Bug	Open	Major	Unresolved	Mikhail Levkovsky	Christian Daher	12/17/2012 17:46		Sprint 3	Sprint 4	

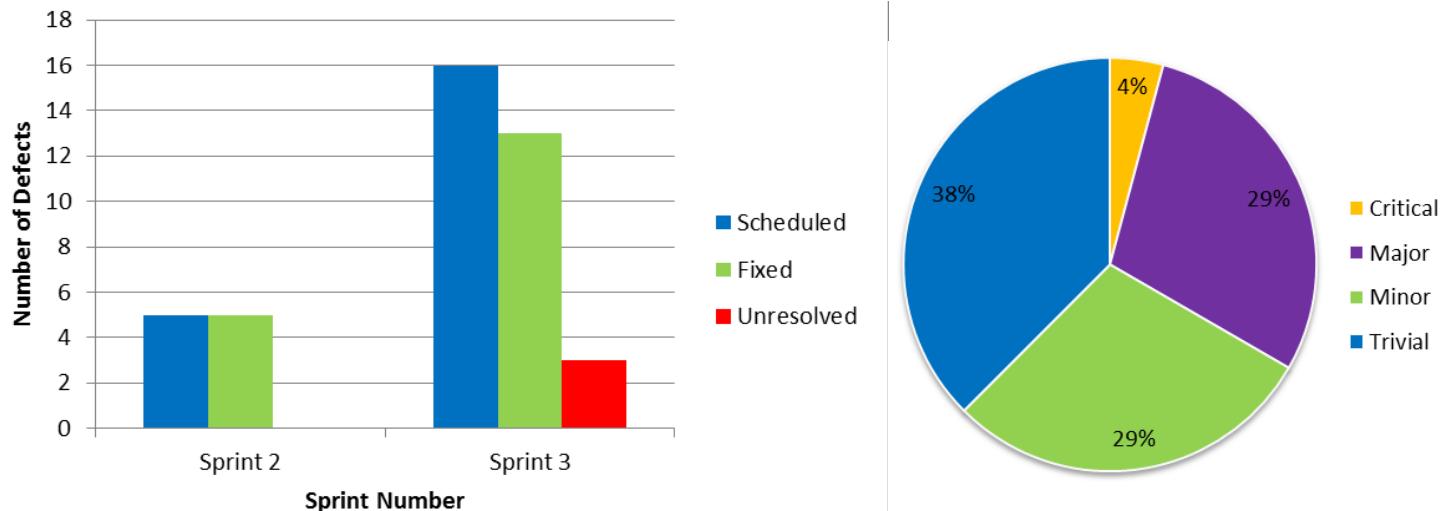


Figure 3-3 Iteration 3 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality

4 Iteration 4 Test Results

4.1 Unit Tests

4.1.1 Unit Test Table & Status

Number of Unit Tests: 11

Number of Passed Unit Tests: 4

Number of Failed Unit Tests: 7

⚠ Test run error Results (Group By: Class Name): 4/11 passed; Item(s) checked: 0			
	Result	Test Name	Project
HomeControllerTest	Passed	Index	TouchForFood.Tests
OrderControllerTest	Inconclusive	AddMenuItem	TouchForFood.Tests
	Failed	CreateTest	TouchForFood.Tests
	Failed	EditingTest	TouchForFood.Tests
	Failed	FinalizeTest	TouchForFood.Tests
	Passed	IndexTest	TouchForFood.Tests
	Failed	RetrieveOrderItemsTest	TouchForFood.Tests
	Failed	ViewFromSessionTest	TouchForFood.Tests
UserControllerTest	Failed	CreateValidUser	TouchForFood.Tests
	Passed	CreateView	TouchForFood.Tests
	Passed	Index	TouchForFood.Tests

Figure 4-1 Iteration 4 Unit Test Results

4.1.2 Unit Test Coverage

Project Coverage: 8.43%

Controller Layer Coverage: 7.93%

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Trinaa@TOSHI 2013-01-13 17:15:43	2467	91.57 %	227	8.43 %
TouchForFood.dll	2467	91.57 %	227	8.43 %
{} TouchForFood	18	100.00 %	0	0.00 %
MvcApplication	18	100.00 %	0	0.00 %
{} TouchForFood.Attributes	7	100.00 %	0	0.00 %
AjaxAttribute	7	100.00 %	0	0.00 %
{} TouchForFood.Controllers	1928	92.07 %	166	7.93 %
CategoryController	92	100.00 %	0	0.00 %
FriendshipController	164	100.00 %	0	0.00 %
HomeController	0	0.00 %	12	100.00 %
ItemController	155	100.00 %	0	0.00 %
MenuController	142	100.00 %	0	0.00 %
Menu_CategoryController	195	100.00 %	0	0.00 %
Menu_ItemController	173	100.00 %	0	0.00 %
OrderController	363	77.23 %	107	22.77 %
OrderController.<>c__DisplayClass14	4	100.00 %	0	0.00 %
OrderStatusController	58	100.00 %	0	0.00 %
Order_ItemController	282	100.00 %	0	0.00 %
RestaurantController	68	100.00 %	0	0.00 %
ReviewController	88	100.00 %	0	0.00 %
UserController	144	75.39 %	47	24.61 %
{} TouchForFood.Models	181	89.16 %	22	10.84 %
CategoryFilterVM	26	100.00 %	0	0.00 %
ItemFilterVM	26	100.00 %	0	0.00 %
OrderStatusHelper	46	100.00 %	0	0.00 %
category	6	100.00 %	0	0.00 %
item	4	100.00 %	0	0.00 %
menu	4	100.00 %	0	0.00 %
menu_category	4	100.00 %	0	0.00 %
menu_item	4	100.00 %	0	0.00 %
menu_item_status	4	100.00 %	0	0.00 %
order	33	89.19 %	4	10.81 %
order_item	4	100.00 %	0	0.00 %
order_item_status	4	100.00 %	0	0.00 %
order_status	0	0.00 %	4	100.00 %
restaurant	10	100.00 %	0	0.00 %
touch_for_foodEntities	2	50.00 %	2	50.00 %
user	0	0.00 %	12	100.00 %
waiter	4	100.00 %	0	0.00 %
{} TouchForFood.Util.Category	28	100.00 %	0	0.00 %
CategoryUtil	28	100.00 %	0	0.00 %
{} TouchForFood.Util.Item	38	100.00 %	0	0.00 %
ItemUtil	38	100.00 %	0	0.00 %
{} TouchForFood.Util.Order	159	82.81 %	33	17.19 %
OrderStatusUtil	73	68.87 %	33	31.13 %
OrderUtil	86	100.00 %	0	0.00 %
{} TouchForFood.Util.Session	61	100.00 %	0	0.00 %
SessionUtil	61	100.00 %	0	0.00 %
{} TouchForFood.Util.User	9	60.00 %	6	40.00 %
UserUtil	9	60.00 %	6	40.00 %
{} TouchForFood.ViewModels	38	100.00 %	0	0.00 %
OrderItemVM	3	100.00 %	0	0.00 %
OrderVM	35	100.00 %	0	0.00 %

Figure 4-2 Iteration 4 Code Coverage Results

4.2 Defects

Table 4-1 Iteration 4 List of Defects

Key	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Created	Resolved	Affects Sprint	Fix Sprint	Time Spent (hours)
CAP-45	Bug	Resolved	Major	Fixed	Josh Hum	Cristian Asenjo	10/20/2012 12:20	11/5/2012 11:26	Sprint 2	Sprint 2	2.33
CAP-53	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	10/20/2012 18:50	10/26/2012 14:28	Sprint 2	Sprint 2	2.00
CAP-54	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	10/25/2012 22:20	12/17/2012 16:02	Sprint 2	Sprint 3	0.17
CAP-55	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	10/25/2012 22:36	12/17/2012 16:00	Sprint 2	Sprint 3	0.08
CAP-77	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	11/6/2012 12:13	11/6/2012 14:18	Sprint 2	Sprint 2	0.50
CAP-82	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 11:22	11/10/2012 15:01	Sprint 2	Sprint 2	3.00
CAP-83	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 15:02	11/10/2012 18:36	Sprint 2	Sprint 2	2.00
CAP-85	Bug	Open	Critical	Unresolved	Ryan Nasr	Cristian Asenjo	11/14/2012 22:09		Sprint 2	Sprint 5	
CAP-86	Bug	Resolved	Minor	Fixed	Ryan Nasr	Cristian Asenjo	11/14/2012 22:10	12/17/2012 15:52	Sprint 2	Sprint 3	1.00
CAP-88	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:53	12/17/2012 16:03	Sprint 2	Sprint 3	1.00
CAP-91	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 20:55	12/17/2012 15:33	Sprint 2	Sprint 3	1.00
CAP-92	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:55	12/11/2012 17:00	Sprint 2	Sprint 3	0.50
CAP-93	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 20:56	12/16/2012 23:12	Sprint 2	Sprint 3	1.25
CAP-94	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:56	12/16/2012 19:25	Sprint 2	Sprint 3	0.50
CAP-95	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 20:58	12/17/2012 17:58	Sprint 2	Sprint 3	1.00
CAP-96	Bug	Resolved	Trivial	Fixed	Cynthia Donato	Patrick Modafferi	11/19/2012 20:58	12/17/2012 15:33	Sprint 2	Sprint 3	3.52
CAP-97	Bug	Resolved	Trivial	Fixed	Mikhail Levkovsky	Patrick Modafferi	11/19/2012 21:00	12/16/2012 17:20	Sprint 2	Sprint 3	0.50
CAP-98	Bug	Resolved	Trivial	Fixed	Josh Hum	Patrick Modafferi	11/19/2012 21:00	1/2/2013 11:14	Sprint 2	Sprint 4	0.52
CAP-99	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 21:16	12/16/2012 23:11	Sprint 2	Sprint 3	0.33
CAP-100	Bug	Resolved	Minor	Fixed	Katrina Anderson	Patrick Modafferi	11/19/2012 21:28	12/30/2012 13:35	Sprint 2	Sprint 4	13.33
CAP-101	Bug	Resolved	Minor	Fixed	Christian Daher	Katrina Anderson	12/16/2012 22:12	12/17/2012 23:01	Sprint 3	Sprint 4	0.25
CAP-102	Bug	Resolved	Minor	Fixed	Katrina Anderson	Katrina Anderson	12/17/2012 0:09	12/17/2012 15:44	Sprint 3	Sprint 3	0.50
CAP-108	Bug	Open	Minor	Unresolved	Christian Daher	Christian Daher	12/17/2012 16:19		Sprint 3	Sprint 4	
CAP-109	Bug	Open	Major	Unresolved	Mikhail Levkovsky	Christian Daher	12/17/2012 17:46		Sprint 3	Sprint 4	
CAP-111	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	12/18/2012 11:20	12/19/2012 17:00	Sprint 4	Sprint 4	8.50

CAP-125	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	12/18/2012 17:09	12/18/2012 20:44	Sprint 4	Sprint 4	6.00
CAP-128	Bug	Open	Trivial	Unresolved	Mikhail Levkovsky	Patrick Modafferi	12/18/2012 17:14		Sprint 4	Sprint 5	
CAP-156	Bug	Resolved	Blocker	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	12/19/2012 15:08	12/19/2012 15:12	Sprint 4	Sprint 4	0.27
CAP-158	Bug	Open	Minor	Unresolved	Mikhail Levkovsky	Katrina Anderson	12/21/2012 15:02		Sprint 4	Sprint 5	
CAP-159	Bug	Open	Major	Unresolved	Ryan Nasr	Matthew Tam	12/21/2012 17:27		Sprint 4	Sprint 5	
CAP-160	Bug	Resolved	Critical	Fixed	Patrick Modafferi	Mikhail Levkovsky	12/21/2012 19:19	12/21/2012 19:33	Sprint 4	Sprint 4	4.00
CAP-161	Bug	Open	Critical	Unresolved	Patrick Modafferi	Cynthia Donato	12/24/2012 13:05		Sprint 4	Sprint 5	
CAP-168	Bug	Closed	Blocker	Fixed	Christian Daher	Katrina Anderson	12/30/2012 22:14	1/2/2013 12:18	Sprint 4	Sprint 4	0.25
CAP-169	Bug	Open	Major	Unresolved	Cynthia Donato	Katrina Anderson	1/1/2013 17:42		Sprint 4	Sprint 5	
CAP-170	Bug	Open	Trivial	Unresolved	Katrina Anderson	Katrina Anderson	1/2/2013 10:49		Sprint 4	Sprint 5	
CAP-190	Bug	Open	Major	Unresolved	Ryan Nasr	Patrick Modafferi	1/2/2013 12:29		Sprint 4	Sprint 5	
CAP-191	Bug	Open	Minor	Unresolved	Cristian Asenjo	Cristian Asenjo	1/2/2013 12:29		Sprint 4	Sprint 5	
CAP-193	Bug	Open	Minor	Unresolved	Patrick Modafferi	Patrick Modafferi	1/2/2013 13:25		Sprint 4	Sprint 5	

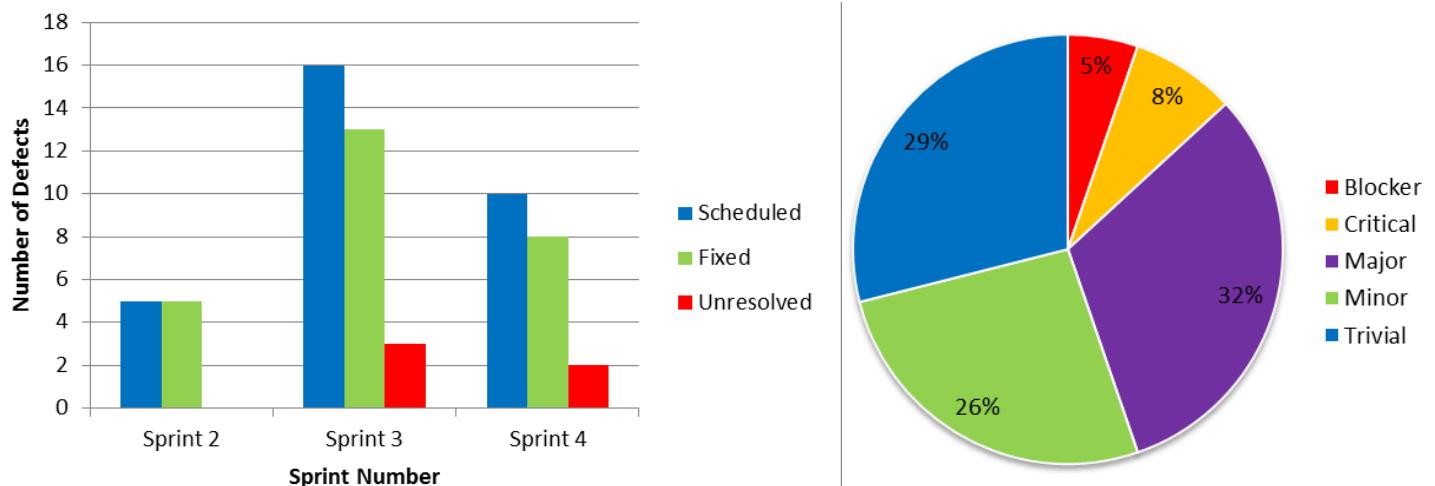


Figure 4-3 Iteration 4 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality

5 Iteration 5 Test Results

5.1 Unit Tests

5.1.1 Unit Test Table & Status

Number of Unit Tests: 34

Number of Passed Unit Tests: 29

Number of Failed Unit Tests: 5

	Result	Test Name	Project	Error Message
HomeControllerTest	<input type="checkbox"/>  Passed	Index	TouchForFood.Tests	
ItemControllerTest	<input type="checkbox"/>  Passed	addItemToMenuTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	CreatePartialTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	EditTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	FilterItemsTest	TouchForFood.Tests	
Menu_CategoryControllerTest	<input type="checkbox"/>  Passed	AddItemTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	AjaxDeleteTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	CreateTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	DeleteConfirmedTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	DeleteTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	RemoveMenuItemTest	TouchForFood.Tests	
MenuControllerTest	<input type="checkbox"/>  Passed	AddCategoryTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	DeleteConfirmedTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	DeleteTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	RemoveMenuCategoryTest	TouchForFood.Tests	
Order_ItemControllerTest	<input type="checkbox"/>  Passed	AcceptTest	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	DeleteConfirmedTest	TouchForFood.Tests	
	<input checked="" type="checkbox"/>  Inconclusive	RemoveOrderItemTest	TouchForFood.Tests	Assert.Inconclusive failed. Verify the correctness of this test method.
OrderControllerTest	<input checked="" type="checkbox"/>  Inconclusive	AddMenuItem	TouchForFood.Tests	Assert.Inconclusive failed. Unable to test with a user in the session
	<input type="checkbox"/>  Passed	CreateTest	TouchForFood.Tests	
	<input checked="" type="checkbox"/>  Failed	EditingTest	TouchForFood.Tests	Test method TouchForFood.Controllers.Tests.OrderControllerTest.EditingTest threw exception: ...
	<input checked="" type="checkbox"/>  Failed	FinalizeTest	TouchForFood.Tests	Test method TouchForFood.Controllers.Tests.OrderControllerTest.FinalizeTest threw exception: ...
	<input type="checkbox"/>  Passed	IndexTest	TouchForFood.Tests	
	<input checked="" type="checkbox"/>  Failed	ViewFromSessionTest	TouchForFood.Tests	Test method TouchForFood.Controllers.Tests.OrderControllerTest.ViewFromSessionTest threw exception: ...
RestaurantControllerTest	<input type="checkbox"/>  Passed	DetailsTest	TouchForFood.Tests	
TableControllerTest	<input type="checkbox"/>  Passed	AssignTest	TouchForFood.Tests	
UserControllerTest	<input type="checkbox"/>  Passed	CreateUserWithInvalidState	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	CreateValidUser	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	CreateView	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	Index	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	LogOnView	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	LogOnWithInvalidStateMock	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	LogOnWithInvalidUser	TouchForFood.Tests	
	<input type="checkbox"/>  Passed	LogOnWithValidUser	TouchForFood.Tests	

Figure 5-1 Iteration 5 Unit Test Results

5.1.2 Unit Test Coverage

Project Coverage: 38.23%

Controller Layer Coverage: 21.56%

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Trinaa@TOSHI 2013-01-18 20:21:37	2634	61.77 %	1630	38.23 %
TouchForFood.Tests.dll	158	15.46 %	864	84.54 %
TouchForFood.dll	2476	76.37 %	766	23.63 %
{ } TouchForFood	18	100.00 %	0	0.00 %
MvcApplication	18	100.00 %	0	0.00 %
{ } TouchForFood.Attributes	7	100.00 %	0	0.00 %
AjaxAttribute	7	100.00 %	0	0.00 %
{ } TouchForFood.Controllers	1903	78.44 %	523	21.56 %
CategoryController	114	100.00 %	0	0.00 %
FriendshipController	164	100.00 %	0	0.00 %
HomeController	0	0.00 %	12	100.00 %
ItemController	121	66.48 %	61	33.52 %
MenuController	156	67.83 %	74	32.17 %
Menu_CategoryController	157	57.93 %	114	42.07 %
Menu_ItemController	195	100.00 %	0	0.00 %
OrderController	386	84.46 %	71	15.54 %
OrderController,<>c_DisplayClass15	4	100.00 %	0	0.00 %
Order_ItemController	228	88.37 %	30	11.63 %
RestaurantController	73	81.11 %	17	18.89 %
ReviewController	91	100.00 %	0	0.00 %
TableController	129	83.23 %	26	16.77 %
UserController	85	41.87 %	118	58.13 %
{ } TouchForFood.Exceptions	2	50.00 %	2	50.00 %
ItemActiveException	2	50.00 %	2	50.00 %
{ } TouchForFood.Mappers	2	4.08 %	47	95.92 %
MenuCategoryOM	0	0.00 %	17	100.00 %
MenuOM	2	6.25 %	30	93.75 %
{ } TouchForFood.Models	198	72.00 %	77	28.00 %
CategoryFilterVM	26	100.00 %	0	0.00 %
ItemFilterVM	53	77.94 %	15	22.06 %
OrderStatusHelper	50	100.00 %	0	0.00 %
category	0	0.00 %	6	100.00 %
item	0	0.00 %	4	100.00 %
menu	0	0.00 %	4	100.00 %
menu_category	0	0.00 %	4	100.00 %
menu_item	0	0.00 %	4	100.00 %
order	57	93.44 %	4	6.56 %
order_item	0	0.00 %	4	100.00 %
restaurant	0	0.00 %	12	100.00 %
table	6	50.00 %	6	50.00 %
touch_for_foodEntities	2	50.00 %	2	50.00 %
user	0	0.00 %	12	100.00 %
waiter	4	100.00 %	0	0.00 %
{ } TouchForFood.Util.Category	28	100.00 %	0	0.00 %
CategoryUtil	28	100.00 %	0	0.00 %
{ } TouchForFood.Util.Html	51	100.00 %	0	0.00 %
HtmlDropDownExtensions	42	100.00 %	0	0.00 %
HtmlDropDownExtensions.<>c_Dis...	9	100.00 %	0	0.00 %
{ } TouchForFood.Util.Item	0	0.00 %	38	100.00 %
ItemUtil	0	0.00 %	38	100.00 %
{ } TouchForFood.Util.Order	93	100.00 %	0	0.00 %
OrderStatusUtil	2	100.00 %	0	0.00 %
OrderUtil	91	100.00 %	0	0.00 %
{ } TouchForFood.Util.Security	34	35.05 %	63	64.95 %
AES	15	19.23 %	63	80.77 %
CustomAuthorizationAttribute	19	100.00 %	0	0.00 %
{ } TouchForFood.Util.Session	61	100.00 %	0	0.00 %
SessionUtil	61	100.00 %	0	0.00 %
{ } TouchForFood.Util.User	0	0.00 %	16	100.00 %
UserUtil	0	0.00 %	16	100.00 %
{ } TouchForFood.ViewModels	79	100.00 %	0	0.00 %
OrderItemVM	3	100.00 %	0	0.00 %
OrderVM	76	100.00 %	0	0.00 %

Figure 5-2 Iteration 5 Code Coverage Results

5.2 Defects

Table 5-1 Iteration 5 List of Defects

Key	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Created	Resolved	Affects Sprint	Fix Sprint	Time Spent (hours)
CAP-45	Bug	Resolved	Major	Fixed	Josh Hum	Cristian Asenjo	10/20/2012 2 12:20	11/5/2012 11:26	Sprint 2	Sprint 2	2.33
CAP-53	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	10/20/2012 2 18:50	10/26/2012 14:28	Sprint 2	Sprint 2	2.00
CAP-54	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	10/25/2012 2 22:20	12/17/2012 16:02	Sprint 2	Sprint 3	0.17
CAP-55	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	10/25/2012 2 22:36	12/17/2012 16:00	Sprint 2	Sprint 3	0.08
CAP-77	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	11/6/2012 12:13	11/6/2012 14:18	Sprint 2	Sprint 2	0.50
CAP-82	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 2 11:22	11/10/2012 15:01	Sprint 2	Sprint 2	3.00
CAP-83	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	11/10/2012 2 15:02	11/10/2012 18:36	Sprint 2	Sprint 2	2.00
CAP-85	Bug	Resolved	Critical	Fixed	Ryan Nasr	Cristian Asenjo	11/14/2012 2 22:09	1/14/2013 21:38	Sprint 2	Sprint 5	0.83
CAP-86	Bug	Resolved	Minor	Fixed	Ryan Nasr	Cristian Asenjo	11/14/2012 2 22:10	12/17/2012 15:52	Sprint 2	Sprint 3	1.00
CAP-88	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 2 20:53	12/17/2012 16:03	Sprint 2	Sprint 3	1.00
CAP-91	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 2 20:55	12/17/2012 15:33	Sprint 2	Sprint 3	1.00
CAP-92	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 2 20:55	12/11/2012 17:00	Sprint 2	Sprint 3	0.50
CAP-93	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 2 20:56	12/16/2012 23:12	Sprint 2	Sprint 3	1.25
CAP-94	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 2 20:56	12/16/2012 19:25	Sprint 2	Sprint 3	0.50
CAP-95	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	11/19/2012 2 20:58	12/17/2012 17:58	Sprint 2	Sprint 3	1.00
CAP-96	Bug	Resolved	Trivial	Fixed	Cynthia Donato	Patrick Modafferi	11/19/2012 2 20:58	12/17/2012 15:33	Sprint 2	Sprint 3	3.52
CAP-97	Bug	Resolved	Trivial	Fixed	Mikhail Levkovsky	Patrick Modafferi	11/19/2012 2 21:00	12/16/2012 17:20	Sprint 2	Sprint 3	0.50
CAP-98	Bug	Resolved	Trivial	Fixed	Josh Hum	Patrick Modafferi	11/19/2012 2 21:00	1/2/2013 11:14	Sprint 2, Sprint 4	Sprint 3, Sprint 4	0.52
CAP-99	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	11/19/2012 2 21:16	12/16/2012 23:11	Sprint 2	Sprint 3	0.33
CAP-100	Bug	Resolved	Minor	Fixed	Katrina Anderson	Patrick Modafferi	11/19/2012 2 21:28	12/30/2012 13:35	Sprint 2	Sprint 3, Sprint 4	13.33
CAP-101	Bug	Resolved	Minor	Fixed	Christian Daher	Katrina Anderson	12/16/2012 2 22:12	12/17/2012 23:01	Sprint 3	Sprint 4	0.25
CAP-102	Bug	Resolved	Minor	Fixed	Katrina Anderson	Katrina Anderson	12/17/2012 2 0:09	12/17/2012 15:44	Sprint 3	Sprint 3	0.50
CAP-108	Bug	Closed	Minor	Fixed	Christian Daher	Christian Daher	12/17/2012 2 16:19	1/14/2013 13:24	Sprint 3	Sprint 5	2.50

CAP-109	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Christian Daher	12/17/2012 17:46	1/15/2013 1:36	Sprint 3	Sprint 5	4.93
CAP-111	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	12/18/2012 11:20	12/19/2012 17:00	Sprint 4	Sprint 4	8.50
CAP-125	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	12/18/2012 17:09	12/18/2012 20:44	Sprint 4	Sprint 4	6.00
CAP-128	Bug	Open	Trivial	<i>Unresolved</i>	Christian Daher	Patrick Modafferi	12/18/2012 17:14		Sprint 4	Sprint 5	
CAP-156	Bug	Resolved	Blocker	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	12/19/2012 15:08	12/19/2012 15:12	Sprint 4	Sprint 4	0.27
CAP-158	Bug	Resolved	Minor	Fixed	Mikhail Levkovsky	Katrina Anderson	12/21/2012 15:02	1/5/2013 14:42	Sprint 4	Sprint 5	2.00
CAP-159	Bug	Resolved	Major	Fixed	Ryan Nasr	Matthew Tam	12/21/2012 17:27	1/4/2013 14:38	Sprint 4	Sprint 5	0.50
CAP-160	Bug	Resolved	Critical	Fixed	Patrick Modafferi	Mikhail Levkovsky	12/21/2012 19:19	12/21/2012 19:33	Sprint 4	Sprint 4	4.00
CAP-161	Bug	Resolved	Critical	Fixed	Patrick Modafferi	Cynthia Donato	12/24/2012 13:05	1/13/2013 11:24	Sprint 4	Sprint 5	0.83
CAP-168	Bug	Closed	Blocker	Fixed	Christian Daher	Katrina Anderson	12/30/2012 22:14	1/2/2013 12:18	Sprint 4	Sprint 4	0.25
CAP-169	Bug	Open	Major	<i>Unresolved</i>	Cynthia Donato	Katrina Anderson	1/1/2013 17:42		Sprint 4	Sprint 5	
CAP-170	Bug	Resolved	Trivial	Fixed	Katrina Anderson	Katrina Anderson	1/2/2013 10:49	1/6/2013 19:03	Sprint 4	Sprint 5	0.42
CAP-190	Bug	Resolved	Major	Fixed	Ryan Nasr	Patrick Modafferi	1/2/2013 12:29	1/4/2013 14:05	Sprint 4	Sprint 5	4.02
CAP-191	Bug	Resolved	Minor	Fixed	Cristian Asenjo	Cristian Asenjo	1/2/2013 12:29	1/13/2013 18:52	Sprint 4	Sprint 5	1.03
CAP-193	Bug	Resolved	Minor	Fixed	Patrick Modafferi	Patrick Modafferi	1/2/2013 13:25	1/5/2013 23:00	Sprint 4	Sprint 5	6.50
CAP-229	Bug	Resolved	Minor	Fixed	Mikhail Levkovsky	Patrick Modafferi	1/6/2013 14:57	1/12/2013 13:13	Sprint 5	Sprint 5	2.52
CAP-235	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Cynthia Donato	1/8/2013 15:54		Sprint 5	Sprint 6	
CAP-236	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Cynthia Donato	1/8/2013 15:56		Sprint 5	Sprint 6	
CAP-237	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Mikhail Levkovsky	1/8/2013 15:58		Sprint 5	Sprint 6	
CAP-238	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Cynthia Donato	1/8/2013 15:58		Sprint 5	Sprint 6	
CAP-239	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Mikhail Levkovsky	1/8/2013 16:00		Sprint 5	Sprint 6	
CAP-240	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Mikhail Levkovsky	1/8/2013 16:02		Sprint 5	Sprint 6	
CAP-241	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Mikhail Levkovsky	1/8/2013 16:04		Sprint 5	Sprint 6	
CAP-242	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Cynthia Donato	1/8/2013 16:04		Sprint 5	Sprint 6	
CAP-243	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Mikhail Levkovsky	1/8/2013 16:06		Sprint 5	Sprint 6	
CAP-246	Bug	Open	Critical	<i>Unresolved</i>	Christian Daher	Mikhail Levkovsky	1/8/2013 20:29		Sprint 5	Sprint 6	
CAP-250	Bug	Open	Minor	<i>Unresolved</i>	Christian Daher	Cynthia Donato	1/8/2013 21:30		Sprint 5	Sprint 6	
CAP-252	Bug	Resolved	Blocker	Fixed	Ryan Nasr	Mikhail Levkovsky	1/10/2013 14:36	1/11/2013 15:27	Sprint 5	Sprint 5	0.52

CAP-253	Bug	Resolved	Blocker	Fixed	Ryan Nasr	Cynthia Donato	1/10/2013 15:20	1/14/2013 20:45	Sprint 5	Sprint 5	1.00
CAP-259	Bug	Closed	Blocker	Fixed	Patrick Modafferi	Matthew Tam	1/11/2013 18:25	1/14/2013 13:52	Sprint 5	Sprint 5	0.17
CAP-262	Bug	Resolved	Blocker	Fixed	Cristian Asenjo	Cynthia Donato	1/13/2013 20:51	1/15/2013 22:38	Sprint 5	Sprint 5	2.60
CAP-274	Bug	Open	Major	Unresolved	Ryan Nasr	Ryan Nasr	1/14/2013 21:39		Sprint 5	Sprint 6	
CAP-275	Bug	Resolved	Blocker	Fixed	Cynthia Donato	Cynthia Donato	1/15/2013 14:02	1/15/2013 21:54	Sprint 5	Sprint 5	0.25
CAP-276	Bug	Open	Minor	Unresolved	Cristian Asenjo	Cristian Asenjo	1/15/2013 14:03		Sprint 5	Sprint 6	
CAP-277	Bug	Open	Critical	Unresolved	Christian Daher	Cynthia Donato	1/15/2013 14:06		Sprint 5	Sprint 6	

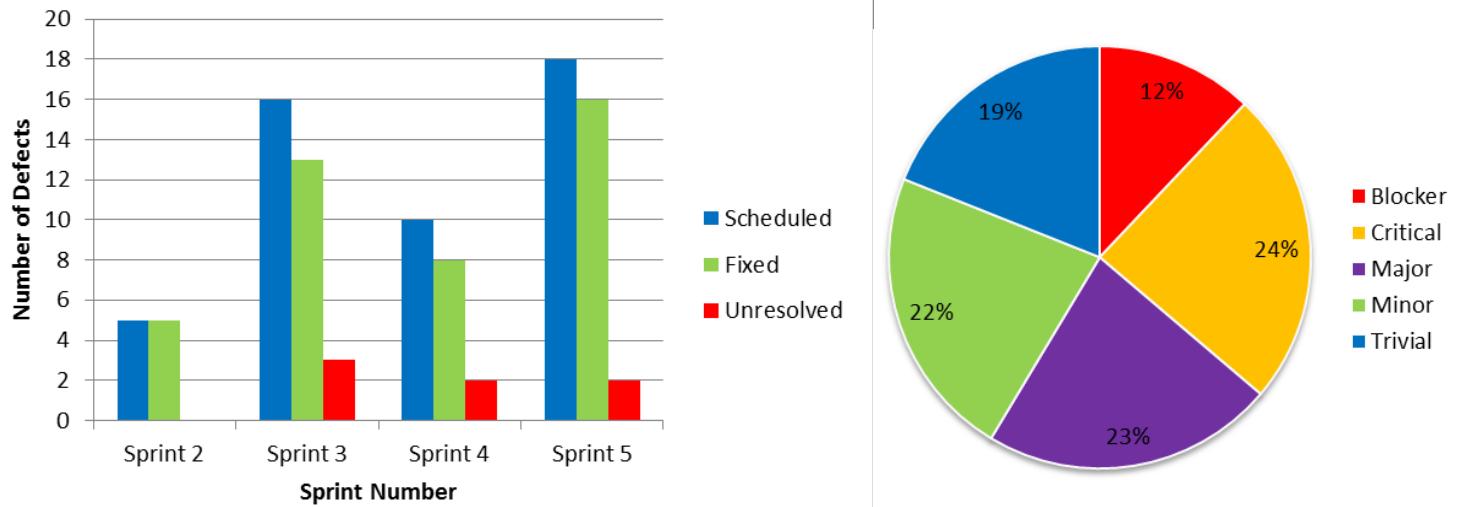


Figure 5-3 Iteration 5 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality

6 Iteration 6 Test Results

6.1 Units Tests

6.1.1 Unit Test Table & Status

Number of Unit Tests: 46

Number of Passed Unit Tests: 26

Number of Failed Unit Tests: 20

It is important to note that some of the unit tests are failing due to badly written test methods and not necessarily to the code they are testing. We are aware that there are issues with the tests and will take the appropriate measures to fix them as soon as possible.

	Result	Test Name	Project	Error Message
BillControllerTest				
<input checked="" type="checkbox"/>	 Failed	ManageBillsTest	TouchForFood.Tests	Class Initialization method TouchForFood.Tests.BillControllerTe
<input checked="" type="checkbox"/>	 Failed	CreateBillForOrderTe	TouchForFood.Tests	Class Initialization method TouchForFood.Tests.BillControllerTe
<input checked="" type="checkbox"/>	 Failed	RemoveOrderItemTe	TouchForFood.Tests	Class Initialization method TouchForFood.Tests.BillControllerTe
<input checked="" type="checkbox"/>	 Failed	DeleteBillFromOrderT	TouchForFood.Tests	Class Initialization method TouchForFood.Tests.BillControllerTe
HomeControllerTest				
<input checked="" type="checkbox"/>	 Failed	Index	TouchForFood.Tests	Test method TouchForFood.Tests.Controllers.HomeControllerT
ItemControllerTest				
<input type="checkbox"/>	 Passed	CreatePartialTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	addItemToMenuItemTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	EditTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	FilterItemsTest	TouchForFood.Tests	
Menu_CategoryControllerTest				
<input type="checkbox"/>	 Passed	DeleteConfirmedTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	DeleteTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	CreateTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	AddItemTest	TouchForFood.Tests	
<input checked="" type="checkbox"/>	 Failed	AjaxDeleteTest	TouchForFood.Tests	Assert.AreEqual failed. Expected:<0>. Actual:<245>.
<input checked="" type="checkbox"/>	 Failed	RemoveMenuItemTe	TouchForFood.Tests	Assert.AreEqual failed. Expected:<1>. Actual:<0>.
MenuCategoryOMTest				
<input checked="" type="checkbox"/>	 Failed	editTest	TouchForFood.Tests	The Web site path 'C:\Users\Ooder\Documents\Concordia\sch
<input type="checkbox"/>	 Passed	deleteTest	TouchForFood.Tests	
MenuControllerTest				
<input type="checkbox"/>	 Passed	RemoveMenuItemCatego	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	DeleteTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	AddCategoryTest	TouchForFood.Tests	
<input type="checkbox"/>	 Passed	DeleteConfirmedTest	TouchForFood.Tests	
MenuItemOMTest				
<input type="checkbox"/>	 Passed	deleteTest	TouchForFood.Tests	
<input checked="" type="checkbox"/>	 Failed	editTest	TouchForFood.Tests	The Web site path 'C:\Users\Ooder\Documents\Concordia\sch

MenuOMTest			
<input type="checkbox"/>		Passed	deleteTest
<input checked="" type="checkbox"/>		Failed	editTest
Order_ItemControllerTest			
<input checked="" type="checkbox"/>		Failed	AcceptTest
<input checked="" type="checkbox"/>		Failed	DeleteConfirmedTest
<input checked="" type="checkbox"/>		Inconclusive	RemoveOrderItemTes
OrderControllerTest			
<input checked="" type="checkbox"/>		Inconclusive	AddMenuItem
<input checked="" type="checkbox"/>		Failed	FinalizeTest
<input checked="" type="checkbox"/>		Failed	EditingTest
<input type="checkbox"/>		Passed	CreateSerServiceRequ
<input type="checkbox"/>		Passed	IndexTest
<input checked="" type="checkbox"/>		Failed	ViewFromSessionTest
OrderOMTest			
<input type="checkbox"/>		Passed	deleteTest
RestaurantControllerTest			
<input checked="" type="checkbox"/>		Failed	DetailsTest
ServiceRequestControllerTest			
<input checked="" type="checkbox"/>		Failed	CreateTest
TableControllerTest			
<input checked="" type="checkbox"/>		Failed	AssignTest
UserControllerTest			
<input type="checkbox"/>		Passed	CreateValidUser
<input type="checkbox"/>		Passed	LogOnWithInvalidSta
<input type="checkbox"/>		Passed	CreateView
<input type="checkbox"/>		Passed	Index
<input type="checkbox"/>		Passed	LogOnWithValidUser
<input type="checkbox"/>		Passed	LogOnView
<input type="checkbox"/>		Passed	CreateUserWithInvalid
<input type="checkbox"/>		Passed	LogOnWithInvalidUse

Figure 6-1 Iteration 6 Unit Test Results

6.1.2 Unit Test Coverage

Project Coverage: 14.34%
Contoller Layer Coverage: 13.95%
MapperLayerCoverage: 16.90%

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Josh@JOSH-PC 2013-02-04 18:09...	4457	85.66 %	746	14.34 %
TouchForFood.dll	4457	85.66 %	746	14.34 %
{} TouchForFood	22	100.00 %	0	0.00 %
{} TouchForFood.Attributes	7	100.00 %	0	0.00 %
{} TouchForFood.Controllers	2949	86.05 %	478	13.95 %
BillController	200	99.01 %	2	0.99 %
CategoryController	114	100.00 %	0	0.00 %
FriendshipController	164	100.00 %	0	0.00 %
HomeController	17	60.71 %	11	39.29 %
ItemController	121	66.48 %	61	33.52 %
MenuController	183	75.62 %	59	24.38 %
Menu_CategoryContr...	161	62.16 %	98	37.84 %
Menu_ItemController	200	100.00 %	0	0.00 %
OrderController	426	80.68 %	102	19.32 %
OrderController.<>c_...	4	100.00 %	0	0.00 %
Order_ItemController	202	88.99 %	25	11.01 %
RestaurantController	90	100.00 %	0	0.00 %
ReviewController	91	100.00 %	0	0.00 %
SearchController	205	100.00 %	0	0.00 %
ServiceRequestContro...	428	100.00 %	0	0.00 %
TableController	232	99.15 %	2	0.85 %
UserController	111	48.47 %	118	51.53 %
{} TouchForFood.Exceptions	6	75.00 %	2	25.00 %
{} TouchForFood.Mappers	590	83.10 %	120	16.90 %
{} TouchForFood.Mappers....	6	60.00 %	4	40.00 %
{} TouchForFood.Models	202	70.38 %	85	29.62 %
{} TouchForFood.Util.Bill	86	100.00 %	0	0.00 %
{} TouchForFood.Util.Categ...	28	100.00 %	0	0.00 %
{} TouchForFood.Util.Html	87	100.00 %	0	0.00 %
{} TouchForFood.Util.Item	0	0.00 %	38	100.00 %
{} TouchForFood.Util.Order	105	89.74 %	12	10.26 %
{} TouchForFood.Util.Search	101	100.00 %	0	0.00 %
{} TouchForFood.Util.Security	97	100.00 %	0	0.00 %

Figure 6-2 Iteration 6 Code Coverage Results

6.2 Defects

The following table of defects list the defects affecting Iteration 6 and/or to be fixed in Iteration 6.

Table 6-1 Iteration 6 List of Defects

Key	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Created	Resolved	Affects Sprint	Fix Sprint	Time Spent (hours)
CAP-128	Bug	Reopened	Trivial	Unresolved	Mikhail Levkovsky	Patrick Modafferi	18/12/2012 17:14		Sprint 4	Sprint 6, Sprint 7	0
CAP-274	Bug	Reopened	Trivial	Unresolved	Mikhail Levkovsky	Ryan Nasr	14/01/2013 21:39		Sprint 5	Sprint 6	0
CAP-251	Bug	Resolved	Minor	Duplicate	Unassigned	Josh Hum	09/01/2013 16:23	15/01/2013 14:28	Sprint 5	Sprint 6	0
CAP-276	Bug	Resolved	Minor	Fixed	Cristian Asenjo	Cristian Asenjo	15/01/2013 14:03	27/01/2013 12:21		Sprint 6	0.16667
CAP-169	Bug	Reopened	Major	Unresolved	Cynthia Donato	Katrina Anderson	01/01/2013 17:42		Sprint 4	Sprint 5, Sprint 6, Sprint 7	7.58333
CAP-261	Bug	Resolved	Major	Won't Fix	Ryan Nasr	Ryan Nasr	13/01/2013 17:26	17/01/2013 19:19	Sprint 5	Sprint 6	1
CAP-308	Bug	Closed	Major	Fixed	Ryan Nasr	Ryan Nasr	17/01/2013 19:23	29/01/2013 0:24		Sprint 6	0.08333
CAP-235	Bug	Resolved	Critical	Fixed	Christian Daher	Cynthia Donato	08/01/2013 15:54	29/01/2013 19:15	Sprint 5	Sprint 6	0.01667
CAP-236	Bug	Resolved	Critical	Duplicate	Christian Daher	Cynthia Donato	08/01/2013 15:56	29/01/2013 19:15	Sprint 5	Sprint 6	0.01667
CAP-237	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 15:58	29/01/2013 19:14	Sprint 5	Sprint 6	0.01667
CAP-238	Bug	Resolved	Critical	Duplicate	Christian Daher	Cynthia Donato	08/01/2013 15:58	29/01/2013 19:14	Sprint 5	Sprint 6	0.01667
CAP-239	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:00	29/01/2013 19:14	Sprint 5	Sprint 6	0.01667
CAP-240	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:02	29/01/2013 19:15	Sprint 5	Sprint 6	0.01667
CAP-241	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:04	29/01/2013 19:14	Sprint 5	Sprint 6	0.01667
CAP-242	Bug	Resolved	Critical	Duplicate	Christian Daher	Cynthia Donato	08/01/2013 16:04	29/01/2013 19:14	Sprint 5	Sprint 6	0.01667
CAP-243	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:06	29/01/2013 19:15	Sprint 5	Sprint 6	0.01667
CAP-246	Bug	Resolved	Critical	Fixed	Christian Daher	Mikhail Levkovsky	08/01/2013 20:29	29/01/2013 18:43	Sprint 5	Sprint 6	1
CAP-277	Bug	Resolved	Critical	Fixed	Christian Daher	Cynthia Donato	15/01/2013 14:06	29/01/2013 18:44	Sprint 5	Sprint 6	0.5
CAP-319	Bug	Resolved	Blocker	Fixed	Unassigned	Cynthia Donato	26/01/2013 13:17	29/01/2013 14:57	Sprint 6	Sprint 6	0.16667

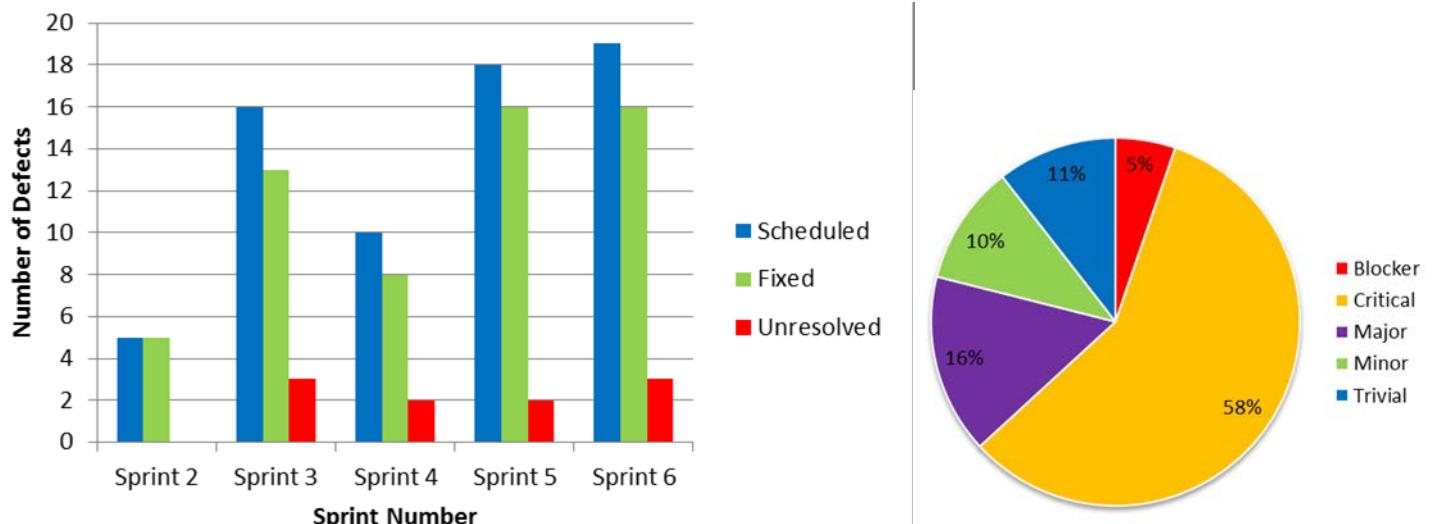


Figure 6-3 Iteration 6 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality

6.3 User Interface Tests & Results

The test cases detailed in Section 1.1 are executed by hand to verify that the system has a functional user interface and that it meets the required standards of TFF. Here are the results:

Table 6-2 Iteration 6 UI Test Results

Test ID	Title	Environment	Sprint	Test Date	Status	Notes
TC025.1	Test Create Order	Google Chrome 24.0.1312.57 m	6	1/27/2013	FAIL	- Doesn't work for a regular user, don't know if this was intended
TC026.1	Test View Menu (Customer)	Google Chrome 24.0.1312.57 m	6	1/27/2013	FAIL	- Can't finalize the order, unless that's expected
TC026.2	Test View Menu (Restaurant)	Google Chrome 24.0.1312.57 m	6	1/27/2013	FAIL	- The user should not be able to see the remove buttons.
TC026.3	Test View Menu (Administrator)	Google Chrome 24.0.1312.57 m	6	1/27/2013	PASS	
TC027.1	Test Menu Editor Main Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	
TC027.2	Test Category Management Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	- On site, it's + New Category. In test case, it specifies "Create New Category"
TC027.3	Test Item Management Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	
TC027.4	Test Menu Edit	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can't remove a category
TC027.5	Test Create Menu	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	
TC029.1	Test Cancel Order	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can't view orders

TC029.2	Test Decline Order Item	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can't view orders
TC029.3	Test Accept Order Item	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can't view orders
TC040.1	Test Login With Non-Existent User	Google Chrome 24.0.1312.57 m	6	1/27/2013	PASS	
TC040.2	Test Create New User	Google Chrome 24.0.1312.57 m	6	1/27/2013	PASS	
TC040.3	Test Login With Existent User	Google Chrome 24.0.1312.57 m	6	1/27/2013	FAIL	- Not case sensitive for user name or for password i.e. can write testinguser / testing
TC040.4	Test Logout	Google Chrome 24.0.1312.57 m	6	1/27/2013	PASS	
TC192.1	Test Create Table Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	
TC192.2	Test Creating Incorrect Table Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can create a table with no information, no error message
TC192.3	Test Edit Table Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	
TC192.4	Test Incorrect Edit Table Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can create a table with no name, no error message appears
TC192.5	Test NFC Link Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	FAIL	- Can click on NFC link to tables that are not linked to a restaurant, which results in an error
TC192.6	Test Table Details Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	
TC192.7	Test Delete Table Page	Google Chrome 24.0.1312.57 m	6	1/28/2013	PASS	

7 Iteration 7 Test Results

7.1 Units Tests

7.1.1 Unit Test Table & Status

Number of Unit Tests: 68

Number of Passed Unit Tests: 40

Number of Failed Unit Tests: 28

	Result	Test Name	Error Message
BillControllerTest			
	<input type="checkbox"/> Failed	CreateBillForOrderTest	Class Initialization method TouchForFood.Tests.BillControllerTest.MyClassIn
	<input type="checkbox"/> Failed	DeleteBillFromOrderTest	Class Initialization method TouchForFood.Tests.BillControllerTest.MyClassIn
	<input type="checkbox"/> Failed	ManageBillsTest	Class Initialization method TouchForFood.Tests.BillControllerTest.MyClassIn
	<input type="checkbox"/> Failed	RemoveOrderItemTest	Class Initialization method TouchForFood.Tests.BillControllerTest.MyClassIn
CategoryOMTest			
	<input type="checkbox"/> Passed	deleteTest	
	<input type="checkbox"/> Failed	editTest	Test method TouchForFood.Tests.CategoryOMTest.editTest threw exception:
HomeControllerTest			
	<input type="checkbox"/> Failed	Index	Test method TouchForFood.Tests.Controllers.HomeControllerTest.Index thi
ItemControllerTest			
	<input type="checkbox"/> Passed	addItemToMenuItemTest	
	<input type="checkbox"/> Passed	CreatePartialTest	
	<input type="checkbox"/> Failed	EditTest	Test method TouchForFood.Tests.ItemControllerTest.EditTest threw excepti
	<input type="checkbox"/> Failed	FilterItemsTest	Test method TouchForFood.Tests.ItemControllerTest.FilterItemsTest threw
ItemOMTest			
	<input type="checkbox"/> Passed	deleteTest	
Menu_CategoryControllerTest			
	<input type="checkbox"/> Passed	AddItemTest	
	<input type="checkbox"/> Failed	AjaxDeleteTest	Assert.AreEqual failed. Expected:<0>. Actual:<338>.
	<input type="checkbox"/> Passed	CreateTest	
	<input type="checkbox"/> Passed	DeleteConfirmedTest	
	<input type="checkbox"/> Passed	DeleteTest	
	<input type="checkbox"/> Failed	RemoveMenuItemTest	Assert.AreEqual failed. Expected:<1>. Actual:<0>.
MenuCategoryOMTest			
	<input type="checkbox"/> Passed	deleteTest	
	<input type="checkbox"/> Failed	editTest	Test method TouchForFood.Tests.MenuCategoryOMTest.editTest threw exce
MenuControllerTest			
	<input type="checkbox"/> Passed	AddCategoryTest	
	<input type="checkbox"/> Passed	DeleteConfirmedTest	
	<input type="checkbox"/> Passed	DeleteTest	
	<input type="checkbox"/> Passed	RemoveMenuCategoryTest	
MenuItemOMTest			
	<input type="checkbox"/> Passed	deleteTest	
	<input type="checkbox"/> Failed	editTest	Test method TouchForFood.Tests.MenuItemOMTest.editTest threw excepti
MenuOMTest			
	<input type="checkbox"/> Passed	deleteTest	
	<input type="checkbox"/> Failed	editTest	Test method TouchForFood.Tests.MenuOMTest.editTest threw exception: ..
Order_ItemControllerTest			
	<input type="checkbox"/> Failed	AcceptTest	Assert.AreEqual failed. Expected:<4>. Actual:<3>.
	<input type="checkbox"/> Failed	DeleteConfirmedTest	Test method TouchForFood.Tests.Order_ItemControllerTest.DeleteConfirm
	<input type="checkbox"/> Inconclusive	RemoveOrderItemTest	Assert.Inconclusive failed. Verify the correctness of this test method.

OrderControllerTest		
<input type="checkbox"/> Inconclusive	AddMenuItem	Assert.Inconclusive failed. Unable to test with a user in the session
<input type="checkbox"/> Passed	CreateSerServiceRequestTest	
<input type="checkbox"/> Failed	EditingTest	Test method TouchForFood.Controllers.Tests.OrderControllerTest.EditingTe
<input type="checkbox"/> Failed	FinalizeTest	Test method TouchForFood.Controllers.Tests.OrderControllerTest.FinalizeT
<input type="checkbox"/> Passed	IndexTest	
<input type="checkbox"/> Failed	ViewFromSessionTest	Test method TouchForFood.Controllers.Tests.OrderControllerTest.ViewFr
OrderOMTest		
<input type="checkbox"/> Passed	deleteTest	
RestaurantControllerTest		
<input type="checkbox"/> Failed	DetailsTest	Class Initialization method TouchForFood.Tests.RestaurantControllerTest.M
RestaurantOMTest		
<input type="checkbox"/> Failed	deleteTest	Test method TouchForFood.Tests.RestaurantOMTest.deleteTest threw exce
<input type="checkbox"/> Failed	editTest	Test method TouchForFood.Tests.RestaurantOMTest.editTest threw excepti
SearchControllerTest		
<input type="checkbox"/> Inconclusive	SearchControllerConstructorTest	Assert.Inconclusive failed. TODO: Implement code to verify target
<input type="checkbox"/> Failed	SearchTest	Assert.AreEqual failed. Expected:<1>. Actual:<0>.
ServiceRequestControllerTest		
<input type="checkbox"/> Passed	CancelConfirmedServiceRequestLockErrorTest	
<input type="checkbox"/> Passed	CancelConfirmedServiceRequestTest	
<input type="checkbox"/> Passed	CloseConfirmedServiceRequestLockErrorTest	
<input type="checkbox"/> Passed	CloseConfirmedServiceRequestTest	
<input type="checkbox"/> Passed	CreateServiceRequestDuplicateErrorTest	
<input type="checkbox"/> Passed	CreateServiceRequestNullTableErrorTest	
<input type="checkbox"/> Passed	CreateServiceRequestTest	
<input type="checkbox"/> Passed	DeleteConfirmedServiceRequestTest	
<input type="checkbox"/> Passed	EditServiceRequestDuplicateErrorTest	
<input type="checkbox"/> Passed	EditServiceRequestLockErrorTest	
<input type="checkbox"/> Passed	EditServiceRequestNullTableErrorTest	
<input type="checkbox"/> Passed	EditServiceRequestTest	
TableControllerTest		
<input type="checkbox"/> Failed	AssignTest	Class Initialization method TouchForFood.Tests.TableControllerTest.MyClas
TableOMTest		
<input type="checkbox"/> Passed	deleteTest	
UserControllerTest		
<input type="checkbox"/> Passed	CreateUserWithInvalidStateModel	
<input type="checkbox"/> Failed	CreateValidUser	Test method TouchForFood.Tests.UserControllerTest.CreateValidUser threw
<input type="checkbox"/> Passed	CreateView	
<input type="checkbox"/> Passed	Index	
<input type="checkbox"/> Passed	LogOnView	
<input type="checkbox"/> Passed	LogOnWithInvalidStateModel	
<input type="checkbox"/> Passed	LogOnWithInvalidUser	
<input type="checkbox"/> Passed	LogOnWithValidUser	
UserOMTest		
<input type="checkbox"/> Passed	deleteTest	
<input type="checkbox"/> Failed	editTest	Test method TouchForFood.Tests.UserOMTest.editTest threw exception: ...
WaiterOMTest		
<input type="checkbox"/> Passed	deleteTest	

Figure 7-1 Iteration 7 Unit Test Results

7.1.2 Unit Test Coverage

Project Coverage: 28.00%
Controller Layer Coverage: 21.48%
MapperLayerCoverage: 47.12%

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Trinna@TOSHI 2013-02-10 13:13:46	3904	72.00 %	1518	28.00 %
TouchForFood.dll	3904	72.00 %	1518	28.00 %
{ } TouchForFood	15	100.00 %	0	0.00 %
{ } TouchForFood.Attributes	7	100.00 %	0	0.00 %
{ } TouchForFood.Controllers	2606	78.52 %	713	21.48 %
{ } TouchForFood.Exceptions	4	50.00 %	4	50.00 %
{ } TouchForFood.Mappers	432	52.88 %	385	47.12 %
{ } TouchForFood.Mappers.Abstract	0	0.00 %	10	100.00 %
{ } TouchForFood.Models	202	70.38 %	85	29.62 %
{ } TouchForFood.Util.Bill	86	100.00 %	0	0.00 %
{ } TouchForFood.Util.Category	28	100.00 %	0	0.00 %
{ } TouchForFood.Util.Html	87	100.00 %	0	0.00 %
{ } TouchForFood.Util.Item	38	100.00 %	0	0.00 %
{ } TouchForFood.Util.Order	117	100.00 %	0	0.00 %
{ } TouchForFood.Util.Search	0	0.00 %	294	100.00 %
{ } TouchForFood.Util.Security	108	100.00 %	0	0.00 %
{ } TouchForFood.Util.Session	69	100.00 %	0	0.00 %
{ } TouchForFood.Util.User	26	61.90 %	16	38.10 %
{ } TouchForFood.ViewModels	79	87.78 %	11	12.22 %

Figure 7-2 Iteration 7 Code Coverage Results

7.2 Defects

Table 7-1 Iteration 7 List of Defects

Key	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Created	Resolved	Affects Sprint	Fix Sprint	Time Spent (hours)
CAP-45	Bug	Resolved	Major	Fixed	Josh Hum	Cristian Asenjo	20/10/2012 12:20	05/11/2012 11:26	Sprint 2	Sprint 2	2.33
CAP-53	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	20/10/2012 18:50	26/10/2012 14:28	Sprint 2	Sprint 2	2.00
CAP-54	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	25/10/2012 22:20	17/12/2012 16:02	Sprint 2	Sprint 3	0.17
CAP-55	Bug	Resolved	Minor	Fixed	Christian Daher	Cristian Asenjo	25/10/2012 22:36	17/12/2012 16:00	Sprint 2	Sprint 3	0.08
CAP-77	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	06/11/2012 12:13	06/11/2012 14:18	Sprint 2	Sprint 2	0.50
CAP-82	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	10/11/2012 11:22	10/11/2012 15:01	Sprint 2	Sprint 2	3.00
CAP-83	Bug	Resolved	Major	Fixed	Cristian Asenjo	Cristian Asenjo	10/11/2012 15:02	10/11/2012 18:36	Sprint 2	Sprint 2	2.00
CAP-85	Bug	Resolved	Critical	Fixed	Ryan Nasr	Cristian Asenjo	14/11/2012 22:09	14/01/2013 21:38	Sprint 2	Sprint 5	0.83
CAP-86	Bug	Resolved	Minor	Fixed	Ryan Nasr	Cristian Asenjo	14/11/2012 22:10	17/12/2012 15:52	Sprint 2	Sprint 3	1.00
CAP-88	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	19/11/2012 20:53	14/01/2013 13:16	Sprint 2	Sprint 3	1.00
CAP-91	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	19/11/2012 20:55	17/12/2012 15:33	Sprint 2	Sprint 3	1.00
CAP-92	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	19/11/2012 20:55	11/12/2012 17:00	Sprint 2	Sprint 3	0.50
CAP-93	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	19/11/2012 20:56	16/12/2012 23:12	Sprint 2	Sprint 3	1.25
CAP-94	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	19/11/2012 20:56	16/12/2012 19:25	Sprint 2	Sprint 3	0.50
CAP-95	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	19/11/2012 20:58	17/12/2012 17:58	Sprint 2	Sprint 3	1.00
CAP-96	Bug	Resolved	Trivial	Fixed	Cynthia Donato	Patrick Modafferi	19/11/2012 20:58	17/12/2012 15:33	Sprint 2	Sprint 3	3.52
CAP-97	Bug	Resolved	Trivial	Fixed	Mikhail Levkovsky	Patrick Modafferi	19/11/2012 21:00	16/12/2012 17:20	Sprint 2	Sprint 3	0.50
CAP-98	Bug	Resolved	Trivial	Fixed	Josh Hum	Patrick Modafferi	19/11/2012 21:00	02/01/2013 11:14	Sprint 2	Sprint 4	0.52
CAP-99	Bug	Resolved	Trivial	Fixed	Cristian Asenjo	Patrick Modafferi	19/11/2012 21:16	16/12/2012 23:11	Sprint 2	Sprint 3	0.33
CAP-100	Bug	Resolved	Minor	Fixed	Katrina Anderson	Patrick Modafferi	19/11/2012 21:28	30/12/2012 13:35	Sprint 2	Sprint 4	13.33
CAP-101	Bug	Resolved	Minor	Fixed	Christian Daher	Katrina Anderson	16/12/2012 22:12	17/12/2012 23:01	Sprint 3	Sprint 4	0.25
CAP-102	Bug	Resolved	Minor	Fixed	Katrina Anderson	Katrina Anderson	17/12/2012 0:09	14/01/2013 13:22	Sprint 3	Sprint 3	0.50
CAP-108	Bug	Closed	Minor	Fixed	Christian Daher	Christian Daher	17/12/2012 16:19	14/01/2013 13:24	Sprint 3	Sprint 5	2.50
CAP-109	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Christian Daher	17/12/2012 17:46	15/01/2013 1:36	Sprint 3	Sprint 5	4.93
CAP-111	Bug	Resolved	Major	Fixed	Patrick Modafferi	Patrick Modafferi	18/12/2012 11:20	19/12/2012 17:00	Sprint 4	Sprint 4	8.50

CAP-125	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	18/12/2012 17:09	18/12/2012 20:44	Sprint 4	Sprint 4	6.00
CAP-128	Bug	Resolved	Trivial	Fixed	Patrick Modafferi	Patrick Modafferi	18/12/2012 17:14	07/02/2013 12:18	Sprint 4	Sprint 7	0.58
CAP-156	Bug	Resolved	Blocker	Fixed	Mikhail Levkovsky	Mikhail Levkovsky	19/12/2012 15:08	19/12/2012 15:12	Sprint 4	Sprint 4	0.27
CAP-158	Bug	Resolved	Minor	Fixed	Mikhail Levkovsky	Katrina Anderson	21/12/2012 15:02	05/01/2013 14:42	Sprint 4	Sprint 5	2.00
CAP-159	Bug	Resolved	Major	Fixed	Ryan Nasr	Matthew Tam	21/12/2012 17:27	04/01/2013 14:38	Sprint 4	Sprint 5	0.50
CAP-160	Bug	Resolved	Critical	Fixed	Patrick Modafferi	Mikhail Levkovsky	21/12/2012 19:19	21/12/2012 19:33	Sprint 4	Sprint 4	4.00
CAP-161	Bug	Resolved	Critical	Fixed	Patrick Modafferi	Cynthia Donato	24/12/2012 13:05	13/01/2013 11:24	Sprint 4	Sprint 5	0.83
CAP-168	Bug	Closed	Blocker	Fixed	Christian Daher	Katrina Anderson	30/12/2012 22:14	02/01/2013 12:18	Sprint 4	Sprint 4	0.25
CAP-169	Bug	In Progress	Major	Unresolved	Cynthia Donato	Katrina Anderson	01/01/2013 17:42		Sprint 4	Sprint 7	8.25
CAP-170	Bug	Resolved	Trivial	Fixed	Katrina Anderson	Katrina Anderson	02/01/2013 10:49	06/01/2013 19:03	Sprint 4	Sprint 5	0.42
CAP-190	Bug	Resolved	Major	Fixed	Ryan Nasr	Patrick Modafferi	02/01/2013 12:29	04/01/2013 14:05	Sprint 4	Sprint 5	4.02
CAP-191	Bug	Resolved	Minor	Fixed	Cristian Asenjo	Cristian Asenjo	02/01/2013 12:29	13/01/2013 18:52	Sprint 4	Sprint 5	1.03
CAP-193	Bug	Resolved	Minor	Fixed	Patrick Modafferi	Patrick Modafferi	02/01/2013 13:25	05/01/2013 23:00	Sprint 4	Sprint 5	6.50
CAP-229	Bug	Resolved	Minor	Fixed	Mikhail Levkovsky	Patrick Modafferi	06/01/2013 14:57	12/01/2013 13:13	Sprint 5	Sprint 5	2.52
CAP-235	Bug	Resolved	Critical	Fixed	Christian Daher	Cynthia Donato	08/01/2013 15:54	29/01/2013 19:15	Sprint 5	Sprint 6	0.02
CAP-236	Bug	Resolved	Critical	Duplicate	Christian Daher	Cynthia Donato	08/01/2013 15:56	29/01/2013 19:15	Sprint 5	Sprint 6	0.02
CAP-237	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 15:58	29/01/2013 19:14	Sprint 5	Sprint 6	0.02
CAP-238	Bug	Resolved	Critical	Duplicate	Christian Daher	Cynthia Donato	08/01/2013 15:58	29/01/2013 19:14	Sprint 5	Sprint 6	0.02
CAP-239	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:00	29/01/2013 19:14	Sprint 5	Sprint 6	0.02
CAP-240	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:02	29/01/2013 19:15	Sprint 5	Sprint 6	0.02
CAP-241	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:04	29/01/2013 19:14	Sprint 5	Sprint 6	0.02
CAP-242	Bug	Resolved	Critical	Duplicate	Christian Daher	Cynthia Donato	08/01/2013 16:04	29/01/2013 19:14	Sprint 5	Sprint 6	0.02
CAP-243	Bug	Resolved	Critical	Duplicate	Christian Daher	Mikhail Levkovsky	08/01/2013 16:06	29/01/2013 19:15	Sprint 5	Sprint 6	0.02
CAP-246	Bug	Resolved	Critical	Fixed	Christian Daher	Mikhail Levkovsky	08/01/2013 20:29	29/01/2013 18:43	Sprint 5	Sprint 6	1.00
CAP-250	Bug	Open	Trivial	Unresolved	Mikhail Levkovsky	Cynthia Donato	08/01/2013 21:30		Sprint 5	Sprint 7	0.00
CAP-251	Bug	Resolved	Minor	Duplicate	Unassigned	Josh Hum	09/01/2013 16:23	15/01/2013 14:28	Sprint 5	Sprint 6	0.00
CAP-252	Bug	Resolved	Blocker	Fixed	Ryan Nasr	Mikhail Levkovsky	10/01/2013 14:36	11/01/2013 15:27	Sprint 5	Sprint 5	0.52
CAP-253	Bug	Resolved	Blocker	Fixed	Ryan Nasr	Cynthia Donato	10/01/2013 15:20	14/01/2013 20:45	Sprint 5	Sprint 5	1.00

CAP-254	Bug	Resolved	Critical	Duplicate	Unassigned	Matthew Tam	11/01/2013 14:15	11/01/2013 14:30	Sprint 5	Sprint 5	0.00
CAP-259	Bug	Closed	Blocker	Fixed	Patrick Modafferi	Matthew Tam	11/01/2013 18:25	14/01/2013 13:52	Sprint 5	Sprint 5	0.17
CAP-260	Bug	Closed	Blocker	Duplicate	Unassigned	Matthew Tam	11/01/2013 18:48	14/01/2013 13:45			0.00
CAP-261	Bug	Resolved	Major	Won't Fix	Ryan Nasr	Ryan Nasr	13/01/2013 17:26	17/01/2013 19:19	Sprint 5	Sprint 6	1.00
CAP-262	Bug	Resolved	Blocker	Fixed	Cristian Asenjo	Cynthia Donato	13/01/2013 20:51	15/01/2013 22:38	Sprint 5	Sprint 5	2.60
CAP-274	Bug	Reopened	Trivial	Unresolved	Mikhail Levkovsky	Ryan Nasr	14/01/2013 21:39		Sprint 5	Sprint 6	0.00
CAP-275	Bug	Resolved	Blocker	Fixed	Cynthia Donato	Cynthia Donato	15/01/2013 14:02	15/01/2013 21:54	Sprint 5	Sprint 5	0.25
CAP-276	Bug	Resolved	Minor	Fixed	Cristian Asenjo	Cristian Asenjo	15/01/2013 14:03	27/01/2013 12:21	Sprint 5	Sprint 6	0.17
CAP-277	Bug	Resolved	Critical	Fixed	Christian Daher	Cynthia Donato	15/01/2013 14:06	29/01/2013 18:44	Sprint 5	Sprint 6	0.50
CAP-308	Bug	Closed	Major	Fixed	Ryan Nasr	Ryan Nasr	17/01/2013 19:23	29/01/2013 0:24	Sprint 5	Sprint 6	0.08
CAP-313	Bug	Resolved	Critical	Fixed	Mikhail Levkovsky	Ryan Nasr	19/01/2013 17:00	08/02/2013 13:33	Sprint 7	Sprint 7	1.02
CAP-315	Bug	Resolved	Major	Fixed	Cristian Asenjo	Mikhail Levkovsky	25/01/2013 14:35	27/01/2013 13:18	Sprint 6	Sprint 6	0.80
CAP-316	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Ryan Nasr	26/01/2013 12:43	08/02/2013 13:32	Sprint 7	Sprint 7	0.52
CAP-317	Bug	Resolved	Major	Fixed	Mikhail Levkovsky	Ryan Nasr	26/01/2013 12:47	08/02/2013 18:02	Sprint 6	Sprint 7	0.02
CAP-319	Bug	Resolved	Blocker	Fixed	Unassigned	Cynthia Donato	26/01/2013 13:17	29/01/2013 14:57	Sprint 6	Sprint 6	0.17
CAP-321	Bug	Open	Major	Unresolved	Cristian Asenjo	Cristian Asenjo	28/01/2013 15:42		Sprint 6	Sprint 7	0.00
CAP-322	Bug	Open	Major	Unresolved	Cristian Asenjo	Cristian Asenjo	28/01/2013 20:00		Sprint 6	Sprint 7	0.00
CAP-323	Bug	Open	Minor	Unresolved	Cristian Asenjo	Cristian Asenjo	28/01/2013 20:01		Sprint 6	Sprint 7	0.00
CAP-324	Bug	Open	Minor	Unresolved	Cristian Asenjo	Cristian Asenjo	28/01/2013 20:02		Sprint 7	Sprint 7	0.00
CAP-325	Bug	Open	Trivial	Unresolved	Mikhail Levkovsky	Josh Hum	28/01/2013 23:50		Sprint 6	Sprint 8	0.00
CAP-386	Bug	Open	Minor	Unresolved	Unassigned	Patrick Modafferi	01/02/2013 23:27		Sprint 7	Sprint 8	0.00
CAP-387	Bug	Open	Major	Unresolved	Unassigned	Katrina Anderson	04/02/2013 20:57		Sprint 7	Sprint 8	0.00
CAP-388	Bug	Resolved	Critical	Fixed	Katrina Anderson	Katrina Anderson	05/02/2013 11:16	07/02/2013 2:28	Sprint 7	Sprint 7	1.90
CAP-390	Bug	Open	Trivial	Unresolved	Unassigned	Katrina Anderson	06/02/2013 23:07		Sprint 7	Sprint 8	0.00
CAP-391	Bug	Open	Minor	Unresolved	Unassigned	Katrina Anderson	06/02/2013 23:19		Sprint 7	Sprint 8	0.00
CAP-392	Bug	Open	Minor	Unresolved	Unassigned	Katrina Anderson	06/02/2013 23:30		Sprint 7	Sprint 8	0.00
CAP-393	Bug	Open	Minor	Unresolved	Unassigned	Katrina Anderson	06/02/2013 23:31		Sprint 7	Sprint 8	0.00
CAP-394	Bug	Open	Trivial	Unresolved	Unassigned	Cynthia Donato	07/02/2013 14:17		Sprint 7	Sprint 8	0.00

CAP-400	Bug	Open	Trivial	Unresolved	Unassigned	Cynthia Donato	07/02/2013 14:28		Sprint 7	Sprint 8	0.00
CAP-402	Bug	Open	Major	Unresolved	Unassigned	Cynthia Donato	09/02/2013 9:25		Sprint 7	Sprint 8	0.00
CAP-403	Bug	Open	Major	Unresolved	Unassigned	Cynthia Donato	09/02/2013 9:26		Sprint 7	Sprint 8	0.00
CAP-404	Bug	Open	Major	Unresolved	Unassigned	Cynthia Donato	09/02/2013 14:55		Sprint 7	Sprint 8	0.00

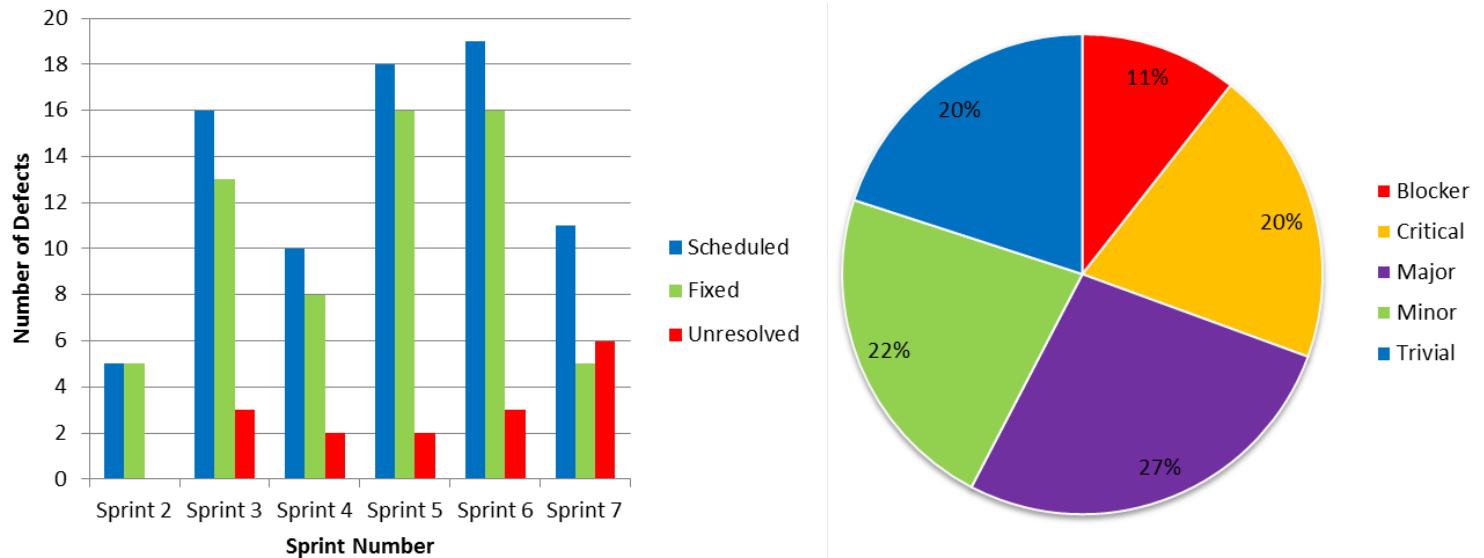


Figure 7-3 Iteration 7 Defect Breakdown vs. Sprint & Number of Defects vs. Criticality

7.3 User Interface Tests & Results

Number of UI Tests: 34

Number of Passed UI Tests: 22

Number of Failed UI Tests: 12

Table 7-2 Iteration 7 UI Test Results

Test ID	Title	Environment	Sprint	Test Date	Status	Notes
TC025.1	Test Create Order	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Doesn't work for a user with role customer.
TC026.1	Test View Menu (Customer)	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Menu list is not exclusive to the restaurant where the user is signed in. - Intervention in the database is required to make menus categories and items visible.
TC026.2	Test View Menu (Restaurant)	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Menu list is not exclusive to the restaurants associated to the user. - The option to order food should not be available to restaurant users.
TC026.3	Test View Menu (Administrator)	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC027.1	Test Menu Editor Main Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- All the current menus are not displayed.
TC027.2	Test Category Management Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC027.3	Test Item Management Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Option to edit an item is unavailable.
TC027.4	Test Menu Edit	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Category cannot be removed.
TC027.5	Test Create Menu	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC029.1	Test Cancel Order	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Option to cancel order is not visible
TC029.2	Test Decline Order Item	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC029.3	Test Accept Order Item	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Order item status not visible - No indication that the order was successfully accepted provided.
TC034.1	Test Create Service Request (Administrator)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC034.2	Test Create Service Request (Restaurant)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	

TC034.3	Test Call Waiter (Customer)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC034.4	Test Edit Service Request (Restaurant)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC034.5	Test Edit Service Request (Administrator)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC034.6	Test Cancel Service Request	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC034.7	Test Close Service Request	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC040.1	Test Login With Non-Existent User	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC040.2	Test Create New User	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC040.3	Test Login With Existing User	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Password is not case sensitive
TC040.4	Test Logout	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC42.1	Test Create Bill (Restaurant)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC42.2	Test Add Item to Bill (Restaurant)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC42.3	Test Remove Item from Bill (Restaurant)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC42.4	Test Delete Bill (Restaurant)	Google Chrome 24.0.1312.57 m	7	10/02/2013	PASS	
TC192.1	Test Create Table Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC192.2	Test Creating Incorrect Table Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- No error message is displayed.
TC192.3	Test Edit Table Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC192.4	Test Incorrect Edit Table Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- No error message is displayed.
TC192.5	Test NFC Link Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	FAIL	- Error is thrown when table isn't associated to a restaurant
TC192.6	Test Table Details Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	
TC192.7	Test Delete Table Page	Google Chrome 24.0.1312.57 m	6	10/02/2013	PASS	

Appendix A References

Appendix B Glossary

Refer to the SRS document - Appendix B Glossary and Appendix C Acronyms for a complete list of terms and definitions.

**Concordia University
Department of Computer Science
and Software Engineering**

Touch For Food

Management

**SOEN 490
Capstone Project
Fall 2012 – Winter 2013**

CloudNine

Josh Hum (Team Leader)	9583157
Katrina Anderson	9106251
Cristian Asenjo	9280014
Christian Daher	9599673
Cynthia Donato	9353852
Mikhail Levkovsky	9583165
Patrick Modafferi	9401377
Ryan Nasr	9605614
Matthew Tam	9675701

Table of Contents

1	List of User Stories	9
2	AHP.....	10
3	Code Quality Goals	10
4	Iteration 0 Report	11
4.1	Person-Hour Work Log	11
5	Iteration 1 Plan	12
5.1	Planned Activities	12
5.2	Person-Hour Estimation.....	12
6	Iteration 1 Report	13
6.1	Person-Hour Work Log	13
6.2	Cumulative Velocities vs. Time.....	14
6.3	Cumulative Flow Diagram.....	14
6.4	Measurement Report.....	15
6.5	Retrospective	16
7	Iteration 2 Plan	17
7.1	Planned Activities	17
7.2	Person-Hour Estimation.....	17
8	Iteration 2 Report	18
8.1	Person-Hour Work Log	18
8.2	Hour Burndown Chart	19
8.3	Issue Burndown Chart	21
8.4	Cumulative Velocities vs. Time.....	22
8.5	Cumulative Flow Diagram.....	22
8.6	Measurement Report.....	23
8.7	Retrospective	26
9	Iteration 3 Plan	28
9.1	Planned Activities	28
9.2	Person-Hour Estimation.....	28
10	Iteration 3 Report	29
10.1	Person-Hour Work Log	29
10.2	Hour Burndown Chart	30
10.3	Issue Burndown Chart	31
10.4	Cumulative Flow Diagram.....	32

10.5	Measurement Report.....	33
10.6	Retrospective	35
11	Iteration 4 Plan	36
11.1	Planned Activities.....	36
11.2	Person-Hour Estimation.....	36
12	Iteration 4 Report	37
12.1	Person-Hour Work Log	37
12.2	Hour Burndown Chart	38
12.3	Issue Burndown Chart	39
12.4	Cumulative Flow Diagram.....	40
12.5	Measurement Report.....	41
12.6	Retrospective	44
13	Iteration 5 Plan	45
13.1	Planned Activities	45
13.2	Person-Hour Estimation.....	47
14	Iteration 5 Report	48
14.1	Person-Hour Work Log	48
14.2	Hour Burndown Chart	49
14.3	Issue Burndown Chart	50
14.4	Cumulative Flow Diagram.....	51
14.5	Measurement Report.....	52
14.6	Retrospective	55
15	Iteration 6 Plan	56
15.1	Planned Activities.....	56
15.2	Person-Hour Estimation.....	58
16	Iteration 6 Report	59
16.1	Person-Hour Work Log	59
16.2	Hour Burndown Chart	61
16.3	Issue Burndown Chart	62
16.4	Cumulative Flow Diagram.....	63
16.5	Measurement Report.....	64
16.6	Retrospective	68
17	Iteration 7 Plan	69
17.1	Planned Activities.....	69
17.2	Person-Hour Estimation.....	71
17.3	Activity-on-Node Planning.....	72

18 Iteration 7 Report	73
18.1 Person-Hour Work Log	73
18.2 Hour Burndown Chart	75
18.3 Issue Burndown Chart	76
18.4 Cumulative Flow Diagram.....	77
18.5 Measurement Report.....	78
18.6 Retrospective	82
Appendix A References.....	84
Appendix B Glossary	85

List of Figures

Figure 1-1 Backlog Stories	9
Figure 2-1 AHP Graph.....	10
Figure 4-1 Person-Hour Work Log.....	11
Figure 6-1 Person-Hour Work Log.....	13
Figure 6-2 Cumulative Velocities vs. Time.....	14
Figure 6-3 Cumulative Flow Diagram.....	14
Figure 6-4 Code Analysis report.....	15
Figure 8-1 Person-Hour Work Log.....	18
Figure 8-2 Hour Burndown Chart.....	19
Figure 8-3 Overestimation of CAP-73 and Underestimation of CAP-48	20
Figure 8-4 Issue Burndown Chart.....	21
Figure 8-5 Cumulative Velocities vs. Time.....	22
Figure 8-6 Cumulative Flow Diagram.....	23
Figure 10-1 Person-Hour Work Log.....	29
Figure 10-2 Hour Burndown Chart.....	30
Figure 10-3 Issue Burndown Chart.....	31
Figure 10-4 Cumulative Flow Diagram.....	32
Figure 10-5 Code Quality Report	34
Figure 12-1 Person-Hour Work Log.....	37
Figure 12-2 Hour Burndown Chart.....	38
Figure 12-3 Issue Burndown Chart.....	39
Figure 12-4 Cumulative Flow Diagram.....	40
Figure 12-5 Code Quality Report	43
Figure 14-1 Person-Hour Work Log.....	48
Figure 14-2 Hour Burndown Chart.....	49
Figure 14-3 Issue Burndown Chart.....	50
Figure 14-4 Cumulative Flow Diagram.....	51
Figure 14-5 Code Quality Report	54
Figure 16-1 Person-Hour Work Log.....	60
Figure 16-2 Hour Burndown Chart.....	61
Figure 16-3 Issue Burndown Chart.....	62
Figure 16-4 Cumulative Flow Diagram.....	63
Figure 16-5 Code Quality Report	67
Figure 17-1 Activity-on-Node Diagram	72

Figure 18-1 Person-Hour Work Log.....	73
Figure 18-2 Hour Burndown Chart.....	75
Figure 18-3 Issue Burndown Chart.....	76
Figure 18-4 Cumulative Flow Diagram.....	77
Figure 18-5 Code Quality Report	81

List of Tables

Table 3-1 Code Quality Goals	10
Table 5-1 Planned Activities.....	12
Table 5-2 Person-Hour Estimation	12
Table 6-1 Completed User Stories.....	16
Table 5-7-1 Planned Activities	17
Table 5-7-2 Person-Hour Estimation	17
Table 9-1 Planned Stories	28
Table 9-2 Planned Defects.....	28
Table 9-3 Person-Hour Estimation	28
Table 11-1 Planned Stories	36
Table 11-2 Planned Defects.....	36
Table 11-3 Person-Hour Estimation	36
Table 13-1 Iteration 5 Planned Tasks	45
Table 13-2 Iteration 5 Planned Stories	46
Table 13-3 Iteration 5 Planned Defects	47
Table 13-4 Iteration 5 Person-Hour Estimation.....	47
Table 15-1 Planned Activities.....	56
Table 17-1 Planned Activities.....	69
Table 17-2 Person-Hour Estimation	71

Touch For Food

Management

Version 7.14

Revision History

Date	Rev.	Description	Author(s)
2012-11-03	1.0	Document Creation	Josh Hum
2012-11-04	1.1	Update doc	Josh Hum
2012-12-03	2.2	Added Iteration 2 retrospective	Josh Hum
2012-12-17	3.3	Made correction to Section 3.3	Josh Hum
2012-12-30	4.4	Reducing redundancy	Katrina Anderson
2013-01-08	5.5	Iteration 3 Plan and Report	Josh Hum
2013-01-10	5.6	Iteration 4 Plan and Report	Josh Hum
2013-01-11	5.7	Iteration 5 Plan added and formatting fixing	Katrina Anderson
2013-01-15	5.8	Fixed user story number error	Josh Hum
2013-01-15	5.9	Iteration 0 Report	Josh Hum
2013-01-18	6.10	Iteration 6 Plan	Josh Hum
2013-02-01	6.11	Iteration 6 Report, Analysis, and Retrospective	Josh Hum
2013-02-01	7.12	Iteration 7 Plan	Josh Hum
2013-02-12	7.13	Reviewed Document for submission	Cynthia Donato
2013-02-12	7.14	Iteration 7 Report, Analysis, and Retrospective	Josh Hum

1 List of User Stories

The following is the list of user stories in the backlog. Stories will be chosen to work on at the beginning of each sprint based on the results of the AHP graph as well as if they are blockers or not.

Backlog		Create Sprint
	CAP-26 View Menu	3
	CAP-25 Order Food	13
	CAP-29 Manage Order	5
	CAP-33 Make Reservation	13
	CAP-34 Call Waiter	1
	CAP-36 Restaurant Statistics and Reports	21
	CAP-37 View Restaurant Statistics and Reviews	8
	CAP-38 Customer Management and Accountability	3
	CAP-39 Leave Review	3
	CAP-40 Sign in/Manage Tables	3
	CAP-41 Customer Bill Management	8
	CAP-42 Restaurant Bill Management	8
	CAP-43 Restaurant Page	13
	CAP-27 Manage Menu	8
	CAP-35 Manage Personal Profile	34
	CAP-28 Set Up DB	2

Figure 1-1 Backlog Stories

2 AHP

The AHP method was used to determine priority of user stories. The following graph shows the results of building AHP cost-value matrixes. High priority stories are between the y-axis and the first diagonal line. Medium priority stories fall between the two diagonal lines and low priority stories come between the second diagonal line and the x-axis.

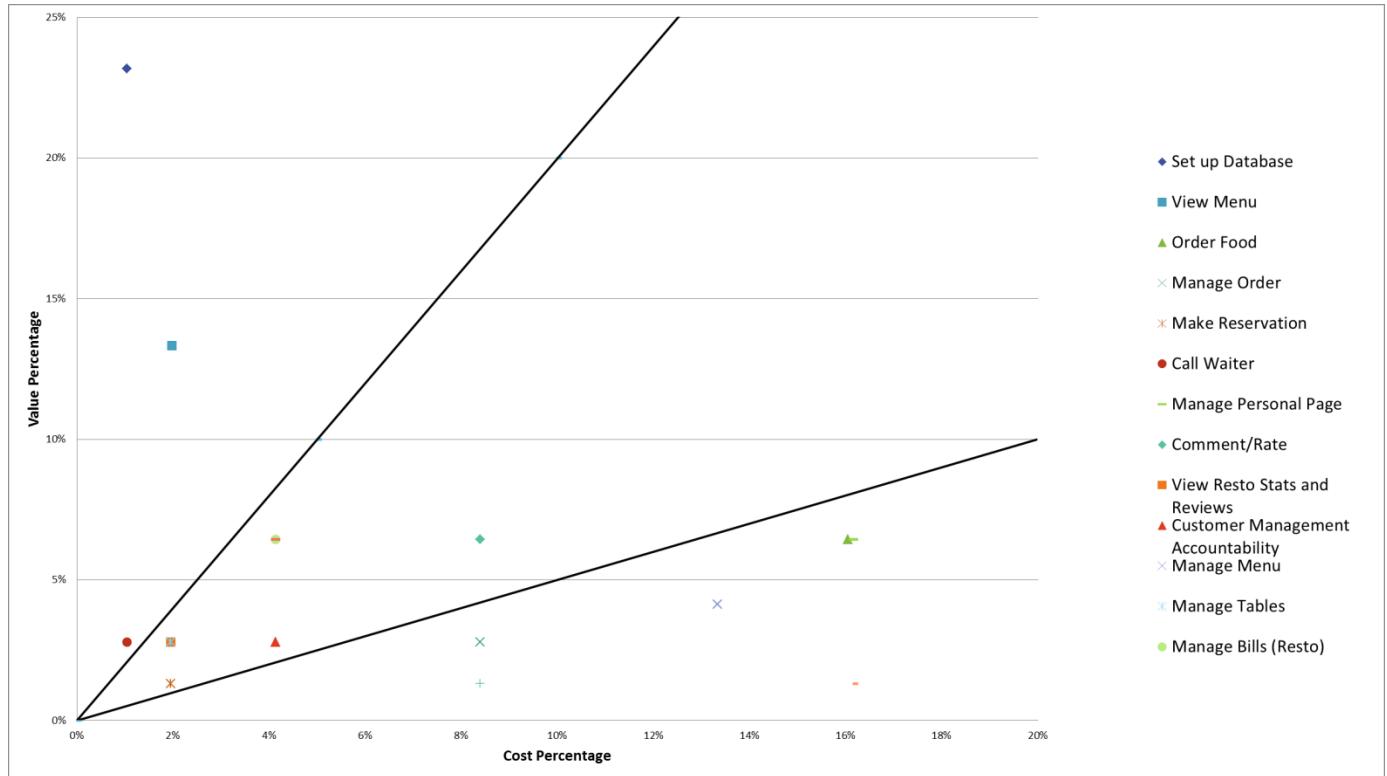


Figure 2-1 AHP Graph

3 Code Quality Goals

We will attempt to meet the following goals with the combination of our code and the Code Metrics Viewer Visual Studio plugin [1]. The plugin outputs a report with the following metrics displayed. Average is defined on the report with a yellow background, while an optimal result is shown with a green background. Below average is shown with a red background. Our goals are shown in the table below.

Table 3-1 Code Quality Goals

Metric	Level
Cyclomatic Complexity (CC)	Average (or lower)
Maintainability Index (MI)	Average (or higher)
Class Coupling (CCP)	Average (or lower)
Lines of Code (LOC)	Average (or lower)
Depth of Inheritance (DOI)	Average (or lower)

4 Iteration 0 Report

Iteration 0 was spent doing research, setup, and preparation for development. Research was conducted for configuration management, code hosting and versioning, server and client side technologies, mobile technologies, and databases. We also made decisions on local machine standards for both documenting and code development. Time was also spent setting up our local machines.

4.1 Person-Hour Work Log

The following work log shows hours worked during Iteration 0.

Start Date: 3/Sep/12 End Date: 21/Oct/12 [Change] (UNREGISTERED)		Total	Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total
Issue	Description	Total	15.583h	27.133h	23.25h	31.917h	23.333h	15h	19.217h	37.083h	7h	199.517h
CAP-8	Documentation - Proposal	3h	0h	0h	0.75h	0h	0.75h	0h	0h	1.5h	0h	3h
CAP-9	Documentation - Vision Document	6.25h	1.5h	0h	1.083h	0h	1.917h	0h	1.25h	0.5h	0h	6.25h
CAP-10	Documentation - SRS	10h	1h	3.667h	0h	2h	0.583h	0.75h	0h	2h	0h	10h
CAP-11	Documentation - SAD	16.417h	2.5h	2.5h	2h	0h	1.667h	0.75h	3h	4h	0h	16.417h
CAP-12	Documentation - User Manual	0.583h	0h	0h	0h	0h	0.583h	0h	0h	0h	0h	0.583h
CAP-14	Correspondence	25.3h	1.083h	2.05h	1.917h	10.917h	3.583h	1.5h	0.167h	4.083h	0h	25.3h
CAP-18	Meetings	102.55h	9h	15.25h	14h	17h	11.5h	12h	8.8h	8h	7h	102.55h
CAP-23	Setup	19.333h	0.5h	1.333h	0.5h	1h	2h	0h	1h	13h	0h	19.333h
CAP-27	Manage Menu	8h	0h	0h	0h	0h	0h	4h	4h	0h	0h	8h
CAP-28	Set Up DB	5h	0h	0h	3h	1h	0h	0h	1h	0h	0h	5h
CAP-44	Documentation - Test Plan	0.75h	0h	0h	0h	0h	0.75h	0h	0h	0h	0h	0.75h
CAP-45	Set up Visual Studio skeleton project	2.333h	0h	2.333h	0h	0h	0h	0h	0h	0h	0h	2.333h

Figure 4-1 Person-Hour Work Log

5 Iteration 1 Plan

Iteration 1 is the first iteration where coding development began. The period of time before Iteration 1 was used for planning, meetings, setup and configuration of each team member's development environment.

5.1 Planned Activities

The two stories planned for Iteration 1 are Cap-35 – Manage Personal Profile and Cap-27 – Manage Menu. We chose to start with them because they provide a foundation for other user stories to be developed.

Table 5-1 Planned Activities

User Story ID	Total Story Points	Related Use Cases
CAP-35, CAP-27, CAP-28	44.00	UC1.1, UC2.5

5.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations. Estimations were made for the worst case, most likely case, and best case. The worst case and best case are +30% and -30% The expected case was then calculated with the formula:

$$\text{Expected case} = 1/6([\text{worst case}] + 4[\text{most likely case}] + [\text{best case}])$$

Table 5-2 Person-Hour Estimation

User Story ID	Worst Case	Most Likely Case	Best Case	Expected Case
US35	110	90	75	90.8
US27	30	20	15	20.8
US28	10	7	3	6.8
Total (ph)	150	117	93	118.4
Velocity (ph/day)	10.7	8.4	6.6	8.5
Velocity (ph/team member/day)	1.2	0.9	0.7	0.9

6 Iteration 1 Report

6.1 Person-Hour Work Log

The following table shows the person-hour work log for each team member and activity during this iteration. This table was generated from the JIRA management system.

Start Date: 22/Oct/12 End Date: 4/Nov/12 [Change] (UNREGISTERED)		Total	Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nas	Total
Issue												
CAP-8	Documentation	0.5h	0h	0h	0h	0h	0h	0h	0.5h	0h	0h	0.5h
CAP-11	CAP-8 SAD	3.5h	0h	0h	0h	0h	0h	0h	1h	2.5h	0h	3.5h
CAP-15	CAP-14 Email	13.333h	0h	0.083h	1.25h	6.5h	4h	0.5h	0h	1h	0h	13.333h
CAP-16	CAP-14 Phone	3.417h	0h	0h	0h	2.5h	0.917h	0h	0h	0h	0h	3.417h
CAP-17	CAP-14 Google Groups	0.667h	0h	0.083h	0h	0h	0h	0h	0.583h	0h	0h	0.667h
CAP-18	Meetings	1h	0h	0h	1h	0h	0h	0h	0h	0h	0h	1h
CAP-19	CAP-18 Weekly Group Meeting	6.75h	0h	0h	1h	1h	0h	2.75h	0h	2h	0h	6.75h
CAP-23	Setup	7.5h	0h	0.5h	3h	0h	0h	0h	0h	4h	0h	7.5h
CAP-24	CAP-8 Proposal	0.5h	0h	0h	0h	0.5h	0h	0h	0h	0h	0h	0.5h
CAP-26	View Menu	4.867h	0h	4.867h	0h	0h	0h	0h	0h	0h	0h	4.867h
CAP-27	Manage Menu	0.05h	0h	0h	0h	0h	0h	0h	0h	0h	0h	0.05h
CAP-44	CAP-8 Test Plan	14.042h	4.5h	0h	1h	1.75h	3h	1h	0.75h	2h	0.042h	14.042h
CAP-47	CAP-27 Menu Editor Main Page	2h	0h	0h	0h	0h	0h	0h	2h	0h	0h	2h
CAP-53	DB structure for menu, item and category is faulty.	2h	0h	0h	0h	0h	0h	0h	2h	0h	0h	2h
CAP-56	CAP-18 Other (GoogleHangouts/Mini Team Meetings/Emergency Meetings/SCRUM)	9.333h	2h	1.75h	1h	1.75h	2.833h	0h	0h	0h	0h	9.333h
CAP-57	CAP-8 Management Docs	18h	0h	0h	0h	17h	1h	0h	0h	0h	0h	18h
CAP-59	CAP-35 SAD 4.3 Login Page Use Case	0.5h	0h	0h	0h	0h	0h	0.5h	0h	0h	0h	0.5h
CAP-60	CAP-35 Customer Settings Page Programming	7.5h	0h	0h	7.5h	0h	0h	0h	0h	0h	0h	7.5h
CAP-61	CAP-35 SAD 4.3 Customer Settings Page Use Case	0.417h	0h	0h	0.417h	0h	0h	0h	0h	0h	0h	0.417h
CAP-62	CAP-35 Customer Sign-up Page Programming	5h	0h	5h	0h	0h	0h	0h	0h	0h	0h	5h
CAP-63	CAP-35 SAD 4.3 Customer Sign-up Use Case	0.5h	0h	0.5h	0h	0h	0h	0h	0h	0h	0h	0.5h
CAP-64	CAP-35 Customer Profile Page View Programming	0.5h	0h	0h	0.5h	0h	0h	0h	0h	0h	0h	0.5h
CAP-65	CAP-35 SAD 5.3 Class Diagram	3h	0h	0h	3h	0h	0h	0h	0h	0h	0h	3h
CAP-66	CAP-35 SAD 4.1 & 4.2 Update	1h	0h	0h	0h	1h	0h	0h	0h	0h	0h	1h

Figure 6-1 Person-Hour Work Log

6.2 Cumulative Velocities vs. Time

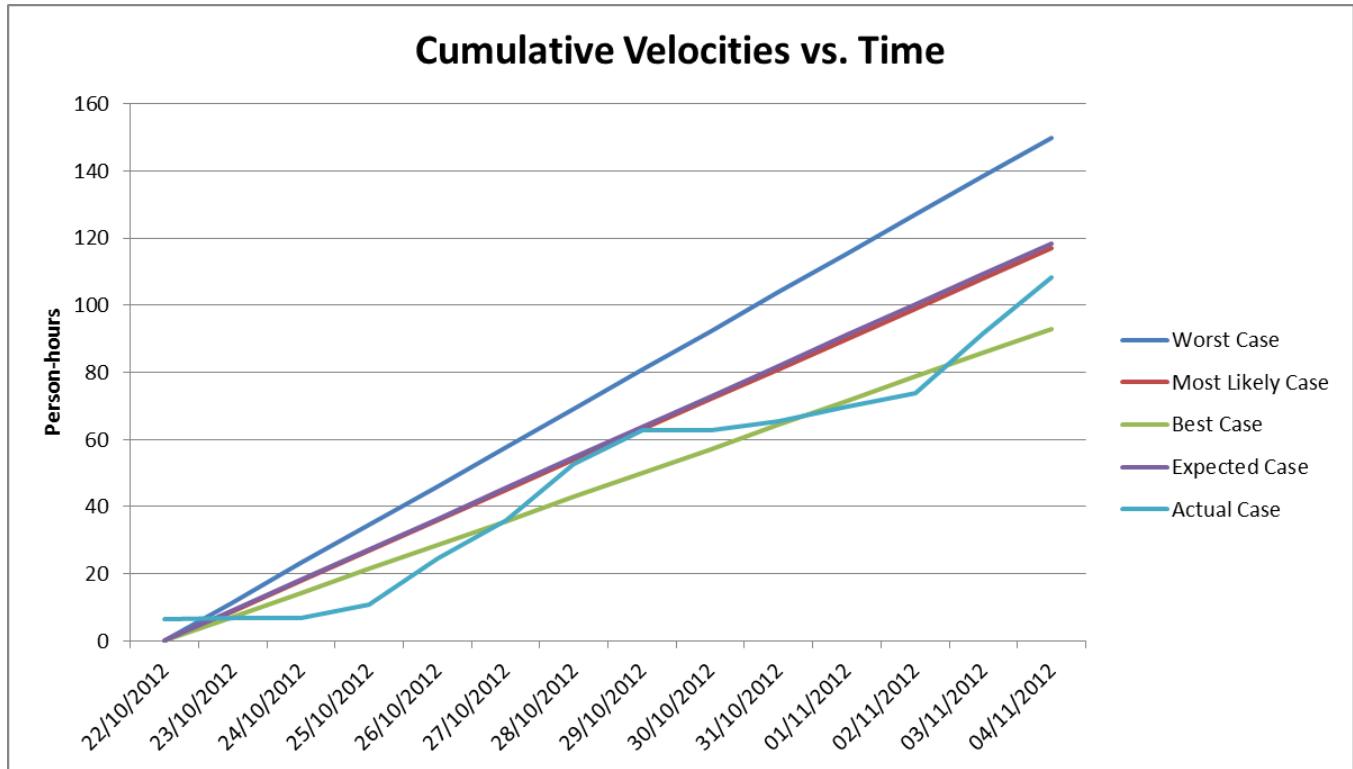


Figure 6-2 Cumulative Velocities vs. Time

6.3 Cumulative Flow Diagram

The following diagram shows the three stories and how the work progressed. Blue tasks have not been started, purple tasks are in progress and green represents a completed story. This diagram was generated in JIRA.

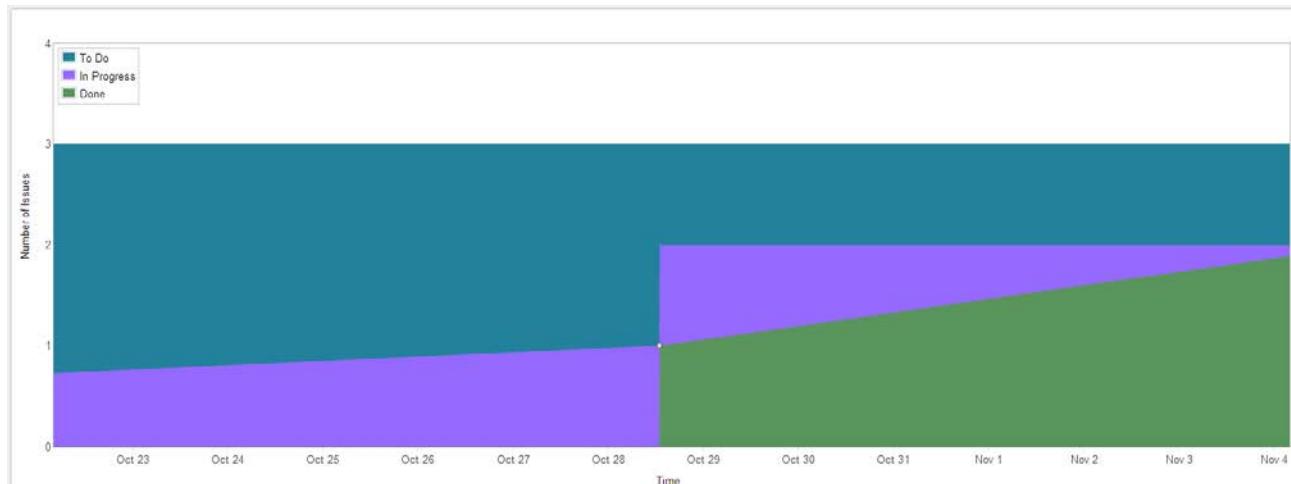


Figure 6-3 Cumulative Flow Diagram

6.4 Measurement Report

6.4.1 Code Quality Analysis

The following report was generated after analyzing the code. All our minimum goals were met while some were exceeded.

Analysis tool used: Code Metrics Viewer

Found at:

<http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>

Version: 1.5.3

Last updated: 2/5/2012

1	Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
2	ModuleScope	TouchForFood.dll	88	286	58	3	413
3	NamespaceScope	TouchForFood	90	4	9	2	7
4	TypeScope	MvcApplication	90	4	9	2	7
5	MemberScope	Application_Start() : void	80	1	3		3
6	MemberScope	MvcApplication()	100	1	1		1
7	MemberScope	RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
8	MemberScope	RegisterRoutes(RouteCollection) : void	82	1	3		2
9	NamespaceScope	TouchForFood.Controllers	78	76	36	3	187
10	TypeScope	CategoryController	76	12	12	3	33
21	TypeScope	HomeController	96	2	2	3	3
24	TypeScope	Menu_CategoryController	69	20	29	3	42
35	TypeScope	MenuController	73	16	30	3	38
46	TypeScope	RestaurantController	76	12	12	3	33
57	TypeScope	UserController	75	14	15	3	38
58	MemberScope	Create() : ActionResult	89	1	2		2
59	MemberScope	Create(user) : ActionResult	69	2	8		6
60	MemberScope	Delete(int) : ActionResult	77	1	5		3
61	MemberScope	DeleteConfirmed(int) : ActionResult	71	1	8		5
62	MemberScope	Details(int) : ViewResult	77	1	5		3
63	MemberScope	Dispose(bool) : void	87	1	3		2
64	MemberScope	Edit(int) : ActionResult	77	1	5		3
65	MemberScope	Edit(user, HttpPostedFileBase) : ActionResult	59	4	11		11
66	MemberScope	Index() : ViewResult	83	1	5		2
67	MemberScope	UserController()	92	1	2		1
68	NamespaceScope	TouchForFood.Models	92	206	21	2	219
69	TypeScope	category	92	9	4	1	11
79	TypeScope	item	92	19	6	1	21
99	TypeScope	menu	93	11	5	1	12
111	TypeScope	menu_category	93	13	5	1	14
125	TypeScope	menu_item	93	13	4	1	13
139	TypeScope	order	92	25	8	1	27

Figure 6-4 Code Analysis report

6.5 Retrospective

In Iteration 1, we completed US27 and US28. This amounts to 10 out of the planned 44 story points. Therefore, it was not a successful iteration. The following table shows the completed user stories:

Table 6-1 Completed User Stories

User Story ID	User Story	Story Points
CAP-27	As a restaurant manager, I would like to create, update, delete and view customized menus.	8
CAP-28	As a developer, I would like to set up the database in order to be able to begin building the system.	2

Although the story US35 could not be completed, it was partially done and progress was made. However, for the next sprint, we will adjust the amount of story points we take on based on our current velocity.

Positives:

- Finished the critical story of setting up the database

Negatives:

- Did not complete all the user stories planned
- Could have better communication
- Missed out on certain management metrics

105.9 person-hours were spent working in Iteration 1. This was close to the 118 person-hour estimate. However, it is below half of the initial budgeted person-hours scheduled. We attribute this to two causes. First, this iteration was started while the developers were right in the middle of midterms. We expect to accomplish much more and spend more time working in the following iteration due to less time constraints. Secondly, the team is not yet fully accustomed to the use of the management system. Thus the work times logged are much lower than the actual work times. There seem to be quite a few data entry errors as well. We will look to correct this in the following iteration.

7 Iteration 2 Plan

7.1 Planned Activities

The stories CAP-26 – View Menu and CAP-35 – Manage Personal Profile were initially planned for Sprint 2. However, early in the sprint, bugs were found in CAP-27 – Manage Menu that were blockers, so the story was reopened and added to the sprint.

Table 5-7-1 Planned Activities

User Story ID	Total Story Points	Related Use Cases
CAP-35, CAP-27, CAP-26	45.00	UC1.1, UC3.1, UC4.1, UC4.2, UC4.3

7.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations.

Table 5-7-2 Person-Hour Estimation

User Story ID	Worst Case	Most Likely Case	Best Case	Expected Case
CAP-35	110	90	75	90.8
CAP-27	30	20	15	20.8
CAP-26	30	20	15	20.8
Total (ph)	170	130	105	132.4
Velocity (ph/day)	12.1	15.5	7.5	9.5
Velocity (ph/team member/day)	1.3	1.7	0.8	1.1

8 Iteration 2 Report

8.1 Person-Hour Work Log

The following table shows the person-hour work log for each team member and activity during this iteration. This table was generated from the JIRA management system. Individual schedules varied so there is a large deviation between hours worked. These hours reflect time spent on all tasks.

Start Date: 6/Nov/12 End Date: 21/Nov/12 [Change] (UNREGISTERED)		Total	Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total
Issue		Total	10h	16.917h	7.75h	9.417h	18.917h	7.417h	16.583h	23.25h	8.5h	118.75h
 CAP-8	Documentation	1h	0h	0h	0h	1h	0h	0h	0h	0h	0h	1h
 CAP-14	Correspondence	10.417h	0h	0.083h	3.25h	0.75h	2.25h	1h	0.083h	3h	0h	10.417h
 CAP-18	Meetings	22h	6h	1h	0.5h	2.667h	0.667h	2.167h	2h	1.5h	5.5h	22h
 CAP-23	Setup	33h	2h	0h	0h	6h	15h	0h	0h	9h	1h	33h
 CAP-26	View Menu	4h	2h	0h	0h	0h	0h	0h	0h	0h	2h	4h
 CAP-27	Manage Menu	24h	0h	0h	0h	0h	0h	0.25h	14h	9.75h	0h	24h
 CAP-35	Manage Personal Profile	13.5h	0h	5.5h	4h	0h	0h	4h	0h	0h	0h	13.5h
 CAP-37	View Restaurant Statistics and Reviews	2.333h	0h	2.333h	0h	0h	0h	0h	0h	0h	0h	2.333h
 CAP-39	Leave Review	3h	0h	3h	0h	0h	0h	0h	0h	0h	0h	3h
 CAP-77	Restaurants are not connected to any user	0.5h	0h	0h	0h	0h	0h	0h	0.5h	0h	0h	0.5h
 CAP-82	Duplicate usernames can be created. Database has to be changed to restrict usernames to be unique.	3h	0h	3h	0h	0h	0h	0h	0h	0h	0h	3h
 CAP-83	Confirm password entry isn't working after moving to a well-partial'd class	2h	0h	2h	0h	0h	0h	0h	0h	0h	0h	2h

Figure 8-1 Person-Hour Work Log

8.2 Hour Burndown Chart

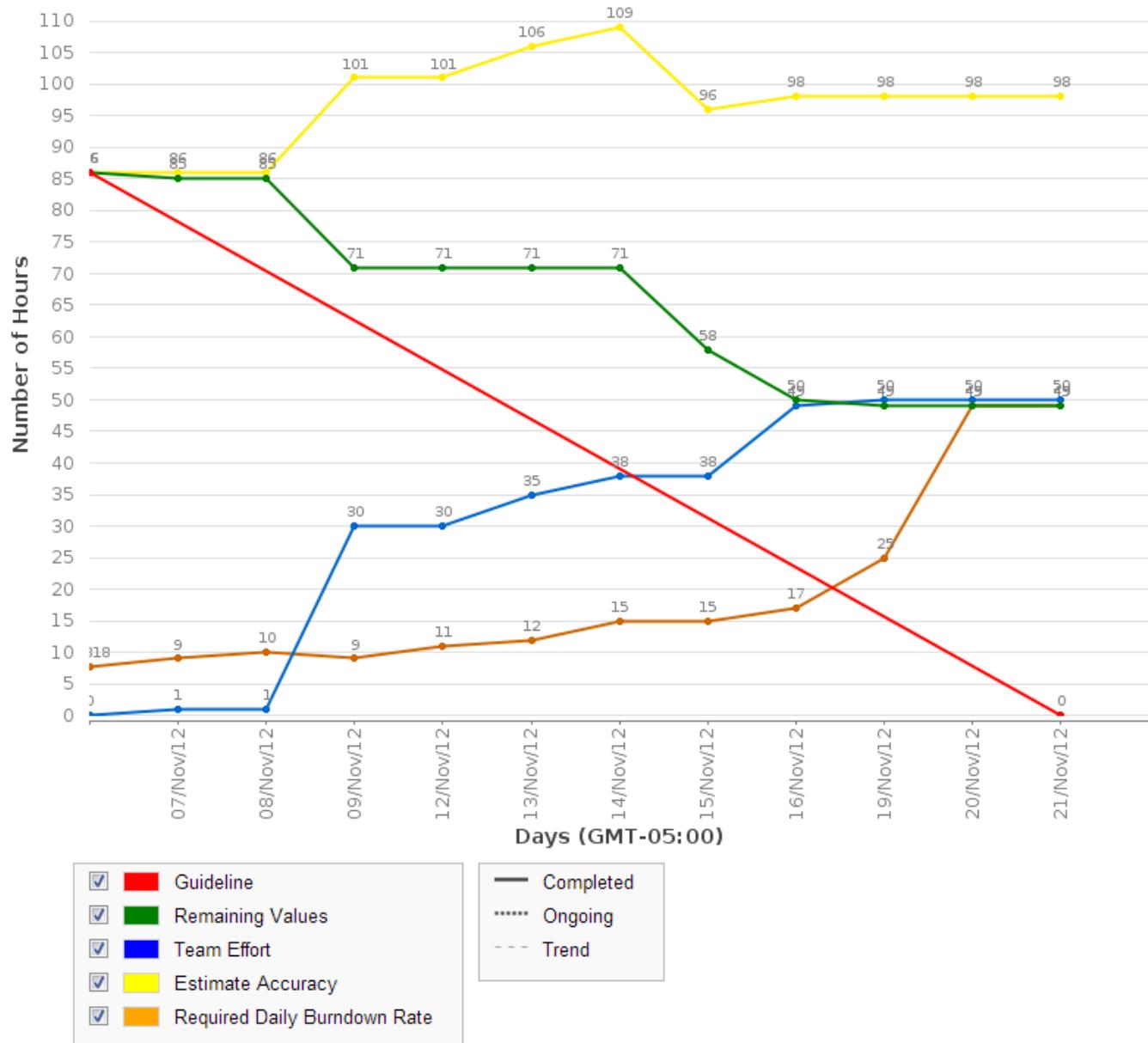


Figure 8-2 Hour Burndown Chart

As seen in the previous chart, the team was not able to burndown everything that was assigned initially in this sprint. We initially estimated 85 hours were needed to complete all the programming tasks. Although the team put 118 hours of work into this iteration, only 50 were spent working on coding tasks as seen by the blue line. The remaining tasks represented by the green line show that there are 49 hours left before all stories can be burned down. We attribute this discrepancy between our total original estimate (85 hours) and the actual work needed ($50 + 49 = 99$ hours) to overly optimistic estimates. For example, some tasks took much longer than planned so the burndown was much slower than desired. The following image is an example of one such task.

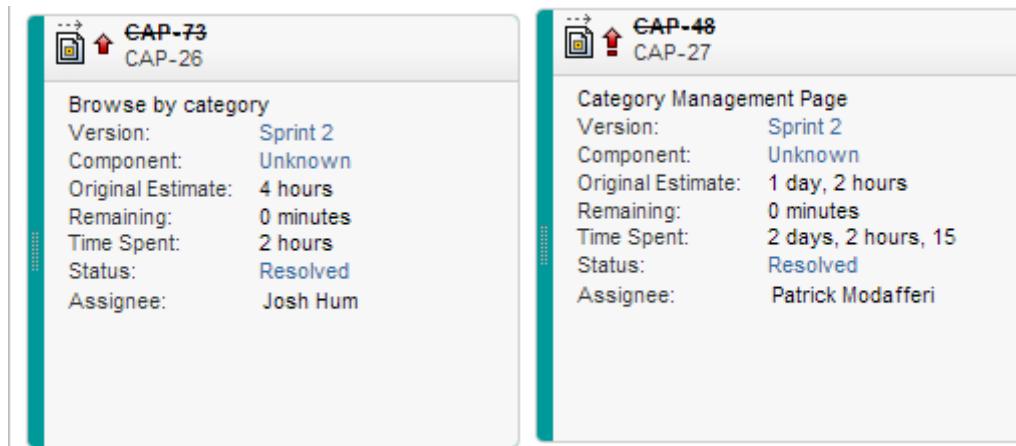


Figure 8-3 Overestimation of CAP-73 and Underestimation of CAP-48

As shown in the previous two figures, there were both overestimations and underestimations made during our sprint planning. However, the discrepancy between the underestimations and the actual time needed were greater than the overestimations which caused the team to fall behind schedule. There were multiple tasks which were underestimated and only two that were overestimated.

8.3 Issue Burndown Chart

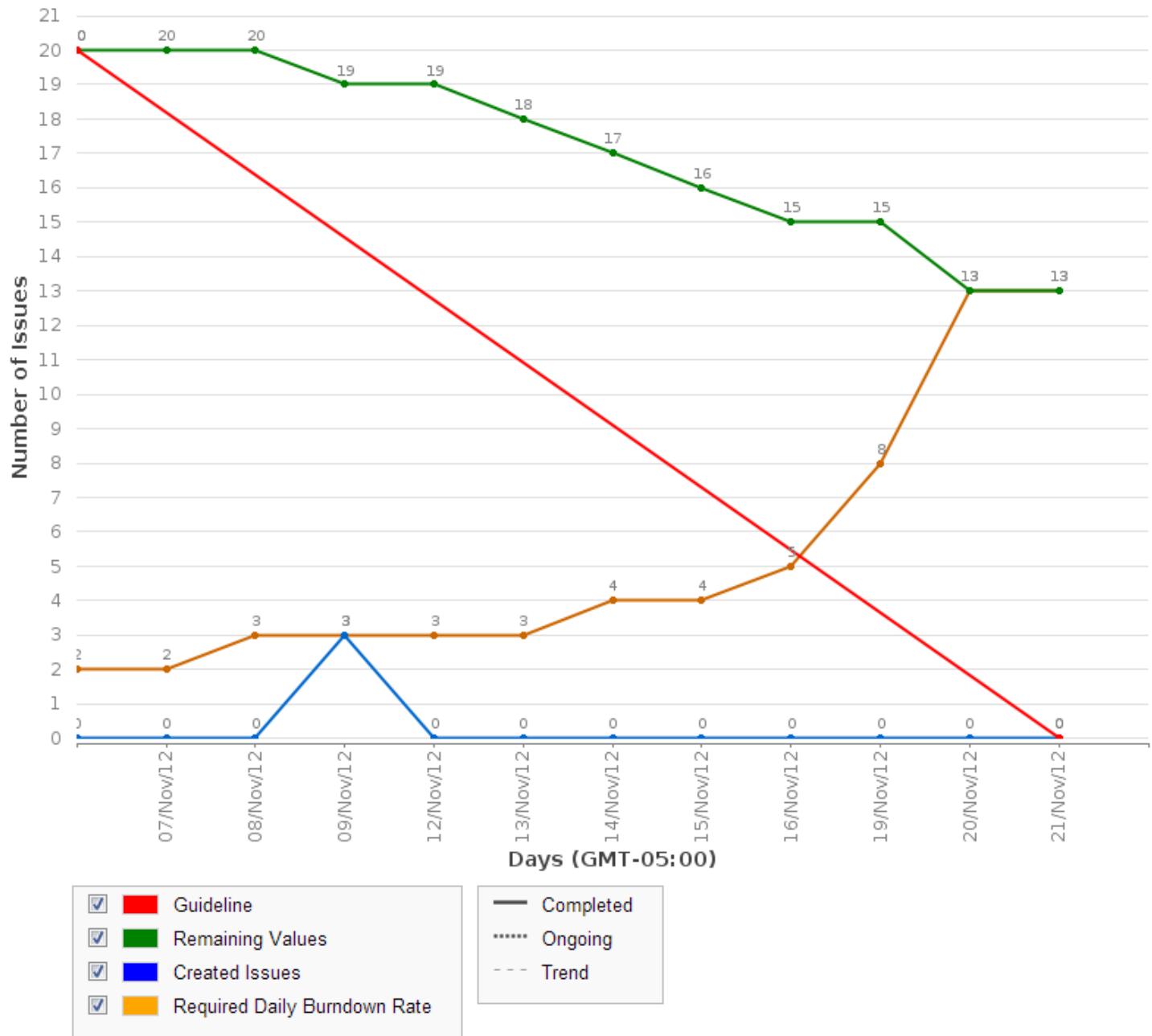


Figure 8-4 Issue Burndown Chart

As seen in the issue burndown chart, 3 tasks were created during the iteration. This was because we overlooked some sub-tasks that were part of the sprint and needed to be added immediately. We will be working to eliminate such mistakes in the future with more detailed planning.

8.4 Cumulative Velocities vs. Time

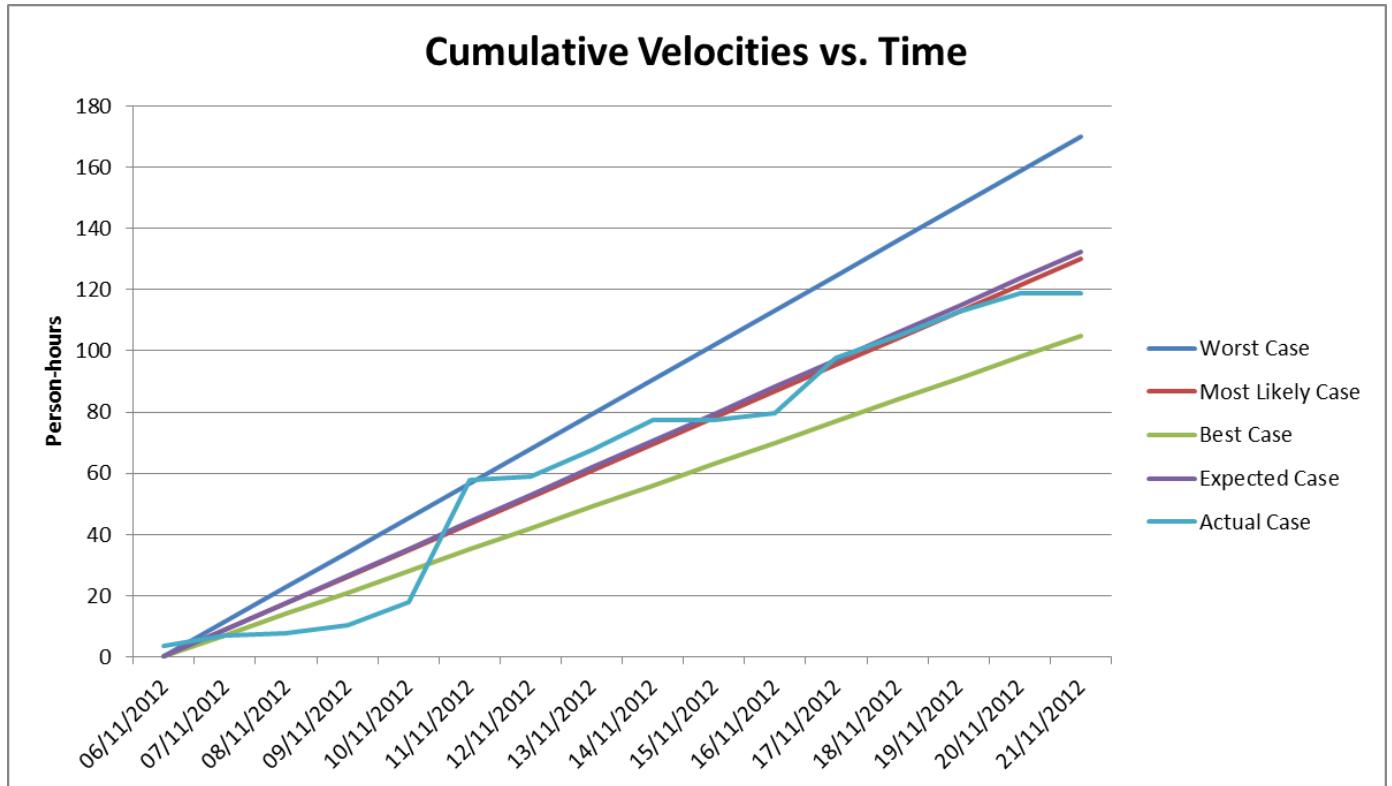
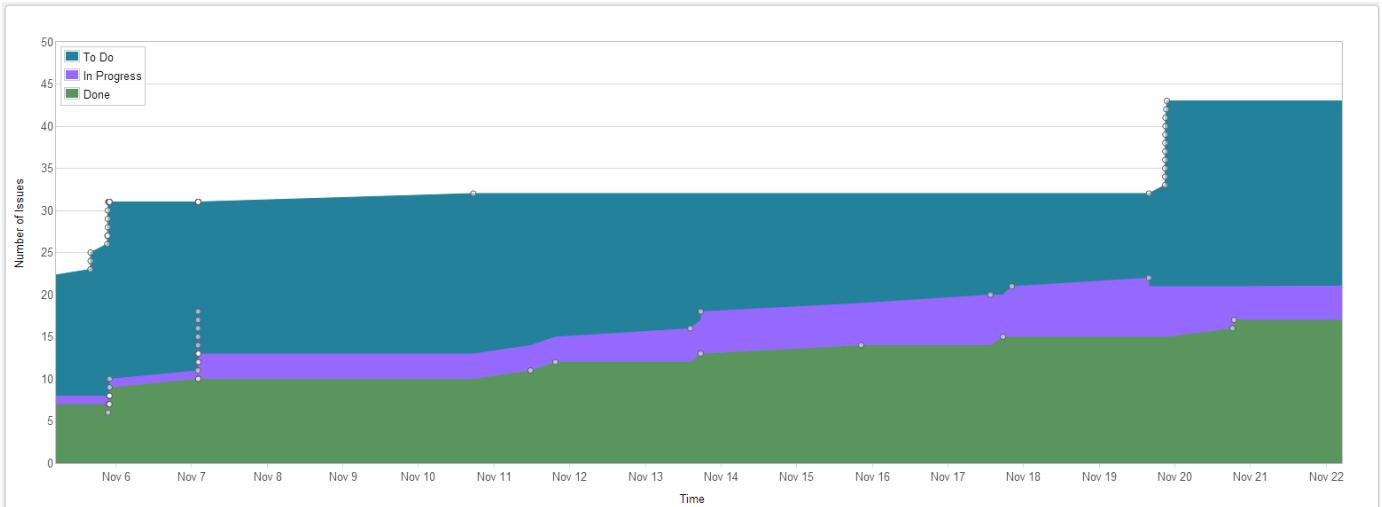


Figure 8-5 Cumulative Velocities vs. Time

8.5 Cumulative Flow Diagram

The following diagram shows the three stories and how the work progressed. Blue tasks have not been started, purple tasks are in progress and green represents a completed story. This diagram was generated in JIRA.

We attribute the increase in blue areas to inaccurate initial time estimations. Therefore, as we increased our time estimations to complete a story, the blue area also increased. The purple area stays relatively the same over the sprint showing that the work in progress stayed relatively constant.

**Figure 8-6 Cumulative Flow Diagram**

8.6 Measurement Report

8.6.1 Code Quality Analysis

The following report was generated after analyzing the code. Only one method did not meet code quality expectations by exceeding the expected number of lines of code. This will be corrected in future iterations.

Analysis tool used:	Code Metrics Viewer
Found at:	http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3
Version:	1.5.3
Last updated:	2/5/2012

1	Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
2	ModuleScope	TouchForFood.Tests.dll	72	13	20	1	29
14	ModuleScope	TouchForFood.dll	85	470	95	3	728
15	NamespaceScope	TouchForFood	81	4	9	2	9
16	TypeScope	MvcApplication	81	4	9	2	9
17	MemberScope	Application_Start() : void	80	1	3		3
18	MemberScope	MvcApplication()	100	1	1		1
19	MemberScope	RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
20	MemberScope	RegisterRoutes(RouteCollection) : void	71	1	3		4
21	NamespaceScope	TouchForFood.Attributes	84	2	5	3	4
22	TypeScope	AjaxAttribute	84	2	5	3	4
23	MemberScope	AjaxAttribute(bool)	87	1	1		2
24	MemberScope	IsValidForRequest(ControllerContext, MethodInfo) : bo	83	1	4		2
25	NamespaceScope	TouchForFood.Controllers	71	148	60	3	369
26	TypeScope	CategoryController	76	13	16	3	36
38	TypeScope	FriendshipController	69	20	29	3	42
49	TypeScope	HomeController	80	3	9	3	4
52	TypeScope	Menu_CategoryController	69	22	30	3	50
64	TypeScope	MenuController	70	20	35	3	53
77	TypeScope	OrderController	68	24	31	3	46
88	TypeScope	RestaurantController	76	14	15	3	35
99	TypeScope	ReviewController	68	8	24	3	21
104	TypeScope	UserController	66	24	38	3	82
105	MemberScope	Create() : ActionResult	74	2	9		3
106	MemberScope	Create(user) : ActionResult	52	5	16		19
107	MemberScope	Delete(int) : ActionResult	77	1	5		3
108	MemberScope	DeleteConfirmed(int) : ActionResult	71	1	8		5
109	MemberScope	Details(int) : ViewResult	77	1	5		3
110	MemberScope	Dispose(bool) : void	87	1	3		2
111	MemberScope	Edit(int) : ActionResult	77	1	5		3
112	MemberScope	Edit(user, HttpPostedFileBase) : ActionResult	59	4	11		11
113	MemberScope	Index() : ViewResult	73	2	12		3
113	MemberScope	Index() : ViewResult	73	2	12		3
114	MemberScope	LogOff() : ActionResult	82	1	3		3
115	MemberScope	LogOn() : ViewResult	87	1	2		2
116	MemberScope	LogOn(string, string) : ActionResult	50	3	16		24
117	MemberScope	UserController()	92	1	2		1
118	NamespaceScope	TouchForFood.Models	93	308	32	2	331
119	TypeScope	category	92	9	4	1	11
129	TypeScope	CategoryFilterVM	92	6	4	1	10
136	TypeScope	friendship	93	11	2	1	11
148	TypeScope	item	92	19	6	1	21
168	TypeScope	menu	93	11	5	1	12
180	TypeScope	menu_category	93	13	5	1	14
194	TypeScope	menu_item	93	13	4	1	13
208	TypeScope	order	92	25	8	1	27
234	TypeScope	order_item	93	13	3	1	13
248	TypeScope	order_status	92	17	5	1	18
266	TypeScope	restaurant	92	21	9	1	25
288	TypeScope	restaurant_user	93	11	3	1	11
300	TypeScope	RestaurantMetadata	94	9	1	1	9
310	TypeScope	review	93	21	6	1	21
332	TypeScope	ReviewMetadata	93	11	5	1	11
344	TypeScope	sysdiagram	93	11	1	1	11
356	TypeScope	touch_for_foodEntities	92	32	19	2	32
389	TypeScope	user	91	27	7	1	32
417	TypeScope	UserMetadata	93	15	5	1	15
433	TypeScope	waiter	93	13	5	1	14
434	MemberScope	first_name.get() : string	98	1	0		1
435	MemberScope	first_name.set(string) : void	95	1	0		1
436	MemberScope	id.get() : int	98	1	0		1
437	MemberScope	id.set(int) : void	95	1	0		1
438	MemberScope	last_name.get() : string	98	1	0		1
439	MemberScope	last_name.set(string) : void	95	1	0		1

439	MemberScope	last_name.set(string) : void	95	1	0	1	
440	MemberScope	orders.get() : ICollection<order>	98	1	2	1	
441	MemberScope	orders.set(ICollection<order>) : void	95	1	2	1	
442	MemberScope	restaurant.get() : restaurant	98	1	1	1	
443	MemberScope	restaurant.set(restaurant) : void	95	1	1	1	
444	MemberScope	resto_id.get() : int?	98	1	1	1	
445	MemberScope	resto_id.set(int?) : void	95	1	1	1	
446	MemberScope	waiter()	87	1	2	2	
447	NamespaceScope	TouchForFood.Util.Category	70	8	12	1	15
448	TypeScope	CategoryUtil	70	8	12	1	15
449	MemberScope	CategoryUtil()	100	1	0	1	1
450	MemberScope	CategoryUtil()	94	1	1	1	1
451	MemberScope	filterListByMenu(menu) : IList<category>	58	6	12	13	

8.7 Retrospective

In Iteration 2, we found that we were overly ambitious. We found that our time estimates were too optimistic which caused us to burn down tasks slower than planned. There were also still issues in learning how to use JIRA and setting up the coding environment. Being in the second sprint, some of these issues are understandable and accounted for in our best case-worst case styled estimates. However, corrective action must be taken immediately to avoid escalating these problems or repeating them further in the process. In this section we will see what the velocity was, what the team did well and did less well, along with what corrective measures we will take to improve our workflow and efficiency.

Our velocity in person-hours for this sprint was 50 person-hours. In terms of story points, our velocity was 0. If we look at cumulative velocity, we must add the velocity from Sprint 1 as well which was 2 story points, giving an average velocity of 1 story point per sprint. Although this figure seems alarming, we must consider that the Manage Personal Profile system is a story worth 34 points which we have started working on and spent over 24 person hours on already. Furthermore, the View Menu and Manage Menu user story are very close to completion. Ultimately, due to the fact that fractions of a user story can't be considered when measure this type of velocity, the actual story point number don't reflect perfectly the work that is done and what is left to be done.

8.7.1 What was learned

The following positive and negative points were gathered from this sprint.

Positives:

- We committed more person-hours than previously
- Our processes and procedures became more defined and clear to the team
- Team members slowly becoming more familiar with the technologies in place
- The technologies chosen for the coding environment and framework are powerful and useful

Negatives:

- Did not complete any user stories planned
- The learning curve on setup, JIRA, and coding slowed down general progress
- The times logged are still inaccurate
- The estimates for the time required for each task was generally incorrect
- Many charts and graphs based on story points are misleading

8.7.2 Changes in methodology

From this sprint a couple of changes in methodology were made. These changes come from the process of refining and perfecting our workflow to match the agile methodology and adapt it to our current project.

First of all, tasks were reorganized in JIRA, and tools were made available to all developers so they may access the specific task and log work under the correct sections. This effort is an attempt to reduce time needed to log ones time in JIRA and encourage a more involved and accurate timesheet in the end.

Another flaw that was exposed was the lack of planning in the start of the sprint. Many of the tasks in the sprint had their estimates completely wrong and some issues and tasks were not accounted for from the start. This is an unacceptable occurrence and in order to reduce the chances of this occurring again, more thorough sprint planning meetings will be conducted where design for the user stories is discussed in more depth, thus giving the team a better idea of what needs to be done and how complex each task really is. As a result of this meeting, all the tasks should be clearly labelled, estimated and accompanied by a description of what is expected to be done for this task to be resolved.

Finally, there is a more technical adjustment made on the JIRA side. It was noticed that the new Greenhopper tools only displayed graphs based on story points which, as discussed earlier, were not completely representative. The scrum leader then switched the sprint planning and reporting to work with JIRA's "Classic" functionality which allows more flexibility such as issues burndown and more importantly an hourly burndown chart to track the teams progress against an estimated guideline.

9 Iteration 3 Plan

9.1 Planned Activities

This sprint was scheduled to span four weeks instead of the usual two weeks because the CloudNine team members had responsibilities to complete final projects and final exams. Documentation defects were assigned in this sprint as were the stories: CAP-26 – View Menu and CAP-27 – Manage Menu. There were no expectations to get much work done within the first two weeks as many team members were overwhelmed with other responsibilities. After the first two weeks, CAP-35 – Manage Personal Profile was removed from the sprint as we realized it shouldn't have been there in the first place. Thus, the following two stories were planned for this sprint.

Table 9-1 Planned Stories

User Story ID	Total Story Points	Related Use Cases
CAP-27, CAP-26	11.00	UC1.1, UC3.1

The following defects were also planned. There are no story points associated to these defects.

Table 9-2 Planned Defects

Defect ID	Hours Estimated
CAP-88, CAP-91, CAP-92, CAP-93, CAP-94, CAP-95, CAP-96, CAP-97, CAP-98, CAP-99, CAP-100	15.25 hours

9.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations. Estimations were made for the worst case, most likely case, and best case. The worst case and best case are +30% and -30% of the most likely case. The expected case was then calculated with the formula:

$$\text{Expected case} = 1/6([\text{worst case}] + 4[\text{most likely case}] + [\text{best case}])$$

The numbers are extremely low because these hours only take planned stories into account. Overall however, there were many defects assigned in this sprint that were estimated to take 15.25 hours to complete. Although in time the sprint took place over four weeks, it was planned to be done in the final two weeks so all estimates are based on a fourteen day time period.

Table 9-3 Person-Hour Estimation

User Story ID	Worst Case	Most Likely Case	Best Case	Expected Case
CAP-27	6.5	5	3.5	5
CAP-26	8.8	6.75	4.7	6.75
Total (ph)	15.3	11.75	8.2	11.75
Velocity (ph/day)	1.09	0.84	0.59	0.84
Velocity (ph/team member/day)	0.12	0.09	0.07	0.09

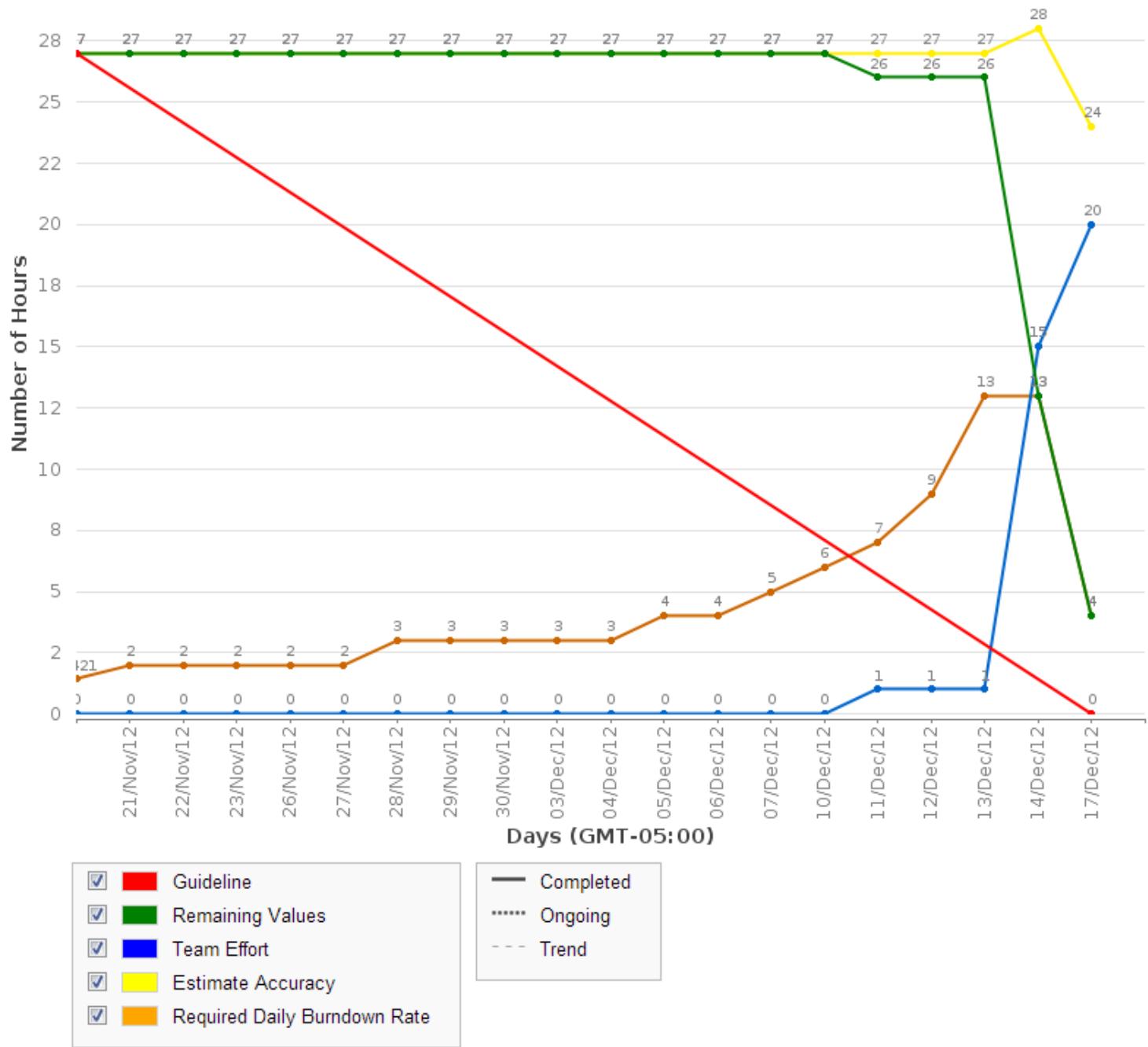
10 Iteration 3 Report

10.1 Person-Hour Work Log

The following table shows the person-hour work log for each team member and activity during this iteration. This table was generated from the Jira management system. As expected, there were very few hours worked this sprint due to other academic and work responsibilities. These hours reflect time spent on all tasks.

Start Date: 21/Nov/12 End Date: 17/Dec/12 [Change] (UNREGISTERED)				Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total	
Issue				Total	11h	7.083h	7.517h	10.75h	11.25h	7.25h	5.5h	9.25h	12.5h	82.1h
 CAP-8	Documentation - Proposal			 1h	0h	0h	0h	0h	0h	0h	1h	0h	1h	
 CAP-10	Documentation - SRS			 3h	0h	2h	0h	0h	0h	1h	0h	0h	3h	
 CAP-14	Correspondence			 7h	2h	0h	0h	0.75h	3.25h	0h	0.25h	0.75h	0h	7h
 CAP-18	Meetings			 36h	3h	3h	4h	6.5h	4.5h	5.25h	4.75h	5h	0h	36h
 CAP-23	Setup			 5.5h	0h	0h	0h	0h	0h	0h	0h	0h	5.5h	5.5h
 CAP-26	View Menu			 8h	4h	0h	0h	0h	0h	0h	0h	0h	4h	8h
 CAP-27	Manage Menu			 3h	1.5h	0h	0h	0h	0h	0h	0h	1.5h	3h	
 CAP-54	Decimals can't be used when creating a price for an item			 0.167h	0.167h	0h	0h	0h	0h	0h	0h	0h	0.167h	
 CAP-55	Database IDs need to be configured with auto-increment			 0.083h	0.083h	0h	0h	0h	0h	0h	0h	0h	0.083h	
 CAP-57	Documentation - Management			 3h	0h	0h	0h	3h	0h	0h	0h	0h	3h	
 CAP-85	Creating a review should use the last visit (i.e. the order) you created			 0.5h	0h	0h	0h	0h	0h	0h	0h	0.5h	0.5h	
 CAP-86	Latest reviews still don't display properly after adding new restaurant reviews			 1h	0h	0h	0h	0h	0h	0h	0h	1h	1h	
 CAP-88	Page 6-7-8-16			 1h	0h	0h	0h	0h	0h	1h	0h	0h	1h	
 CAP-91	Page 17			 1h	0h	1h	0h	0h	0h	0h	0h	0h	1h	
 CAP-92	Page 18			 0.5h	0h	0h	0h	0h	0h	0h	0.5h	0h	0.5h	
 CAP-93	Page 9			 1.25h	0h	0.75h	0h	0h	0h	0h	0.5h	0h	1.25h	
 CAP-94	Page 20			 0.5h	0h	0h	0h	0h	0h	0h	0.5h	0h	0.5h	
 CAP-95	24-27-28-32			 1h	0h	0h	0h	0h	0h	0h	1h	0h	1h	
 CAP-96	Traceability Page 40 something			 3.517h	0h	0h	3.517h	0h	0h	0h	0h	0h	3.517h	
 CAP-97	Page 57-61-67-73			 0.5h	0h	0h	0h	0h	0h	0h	0.5h	0h	0.5h	
 CAP-98	Page 76-80			 0.5h	0h	0h	0h	0.5h	0h	0h	0h	0h	0.5h	
 CAP-99	Page 28			 0.333h	0h	0.333h	0h	0h	0h	0h	0h	0h	0.333h	
 CAP-100	Eliminate Redundancy			 3h	0h	0h	0h	0h	3h	0h	0h	0h	3h	
 CAP-101	Vision Doc Section 4.3 Missing Information			 0.25h	0.25h	0h	0h	0h	0h	0h	0h	0h	0.25h	
 CAP-102	Test Plan Incorrect Dates			 0.5h	0h	0h	0h	0h	0.5h	0h	0h	0h	0.5h	

Figure 10-1 Person-Hour Work Log

10.2 Hour Burndown Chart**Figure 10-2 Hour Burndown Chart**

In total, between planned stories and defects, there were 27 hours estimated to working. As seen in the above graph, most of the work was either done last minute or logged last minute. Both stories were completed and all defects were completed except for two. Overall, we met our goals for our stories and for most of our defects. However, we were not satisfied with our overall time management because we felt that we did not plan enough and therefore felt no pressure to get more work done.

10.3 Issue Burndown Chart

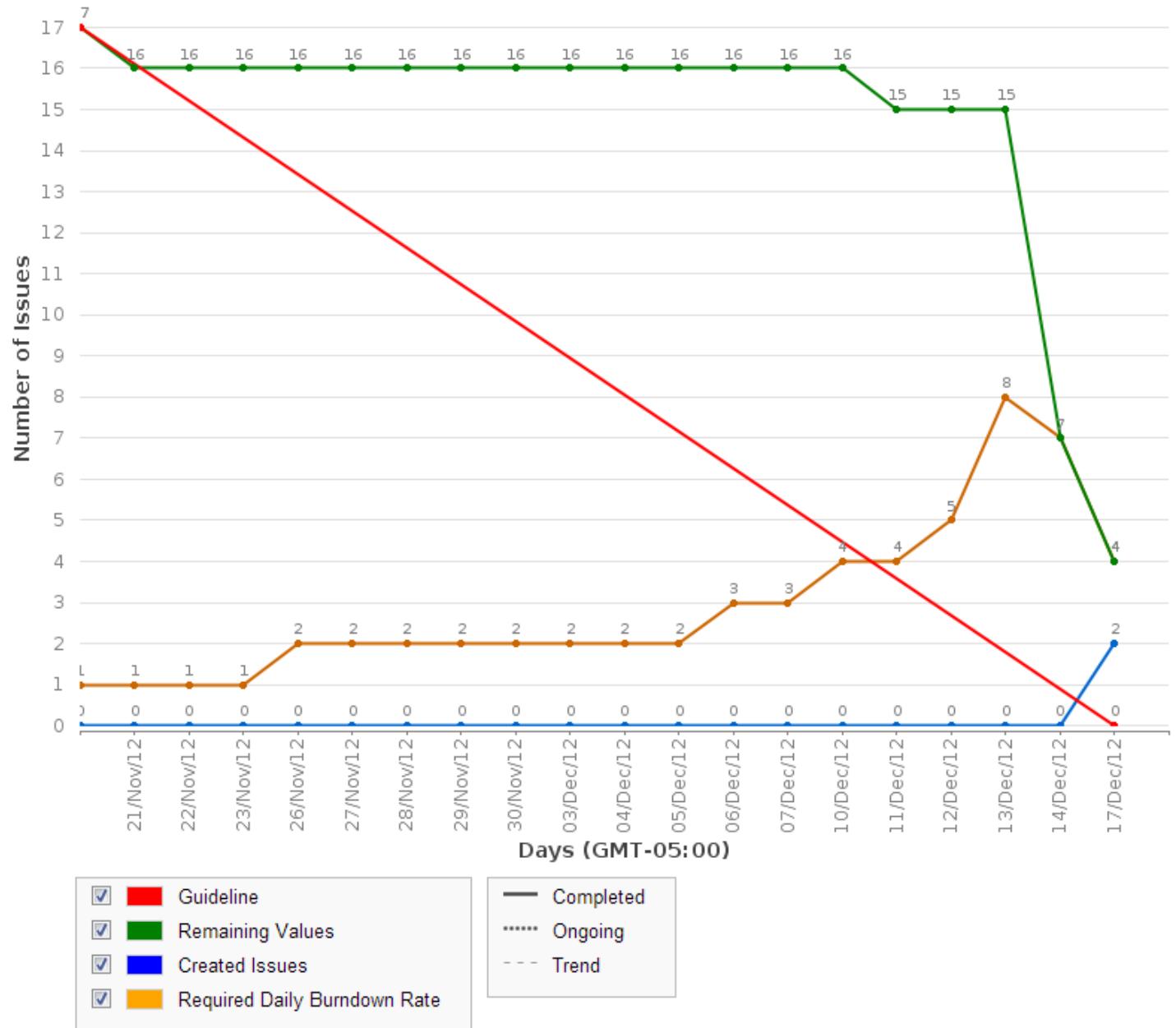


Figure 10-3 Issue Burndown Chart

As seen in the issue burndown chart, most of the issues (both stories and defects) were completed. The two defects that were not completed were CAP-98 and CAP-100 which we realized would take more time and planning than expected.

10.4 Cumulative Flow Diagram

The following diagram shows the cumulative flow of the project so far based on the number of issues completed, in progress, and to do. Although we are completing issues, we are also finding bugs and adding them to the backlog which explains the increase of issues to do at the end of the sprint.

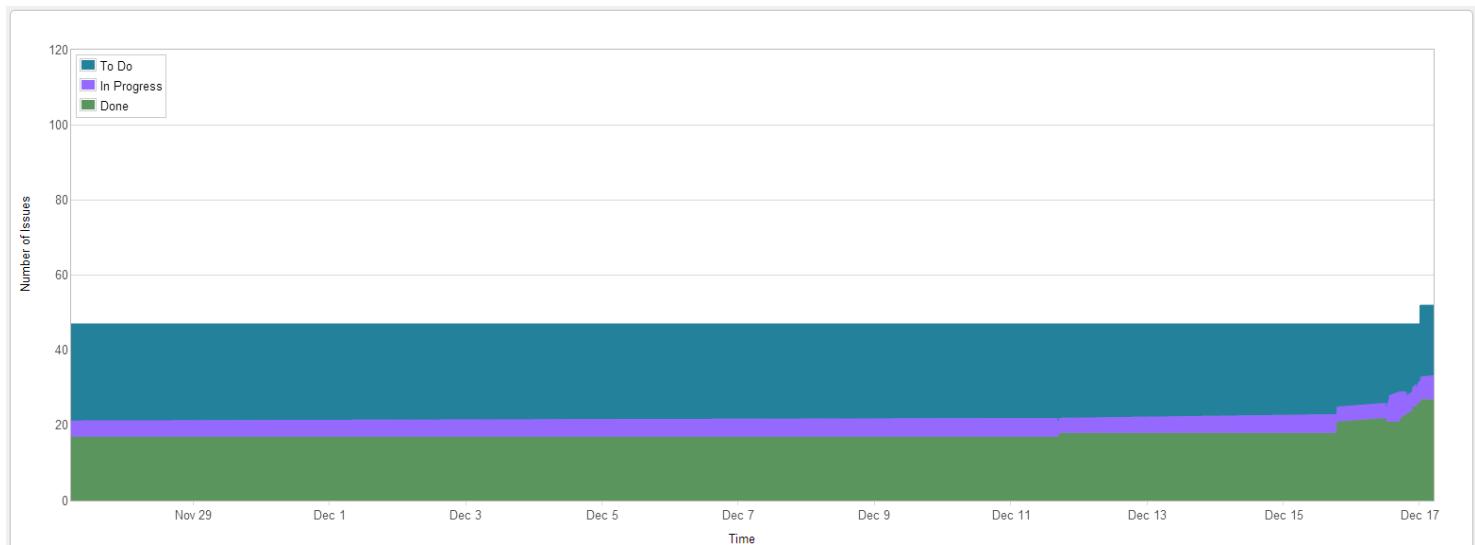


Figure 10-4 Cumulative Flow Diagram

10.5 Measurement Report

10.5.1 Code Quality Analysis

The following report was generated after analyzing the code. Only one method did not meet code quality expectations by exceeding the expected number of lines of code. This was found in an earlier iteration and has not been yet corrected.

Analysis tool used: Code Metrics Viewer

Found at:

<http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>

Version: 1.5.3

Last updated: 2/5/2012

1	Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
2	ModuleScope	TouchForFood.Tests.dll	72	13	20	1	29
14	ModuleScope	TouchForFood.dll	85	494	95	3	787
15	NamespaceScope	TouchForFood	81	4	9	2	9
16	TypeScope	MvcApplication	81	4	9	2	9
17	MemberScope	Application_Start() : void	80	1	3		3
18	MemberScope	MvcApplication()	100	1	1		1
19	MemberScope	RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
20	MemberScope	RegisterRoutes(RouteCollection) : void	71	1	3		4
21	NamespaceScope	TouchForFood.Attributes	84	2	5	3	4
22	TypeScope	AjaxAttribute	84	2	5	3	4
23	MemberScope	AjaxAttribute(bool)	87	1	1		2
24	MemberScope	IsValidForRequest(ControllerContext, MethodInfo) : bool	83	1	4		2
25	NamespaceScope	TouchForFood.Controllers	71	170	66	3	426
26	TypeScope	CategoryController	76	13	16	3	36
38	TypeScope	FriendshipController	69	20	29	3	42
49	TypeScope	HomeController	80	3	9	3	4
52	TypeScope	ItemController	74	17	29	3	41
64	TypeScope	Menu_CategoryController	69	22	30	3	50
76	TypeScope	MenuController	70	20	35	3	53
89	TypeScope	OrderController	68	24	31	3	46
100	TypeScope	RestaurantController	76	14	15	3	35
111	TypeScope	ReviewController	62	13	30	3	32
116	TypeScope	UserController	66	24	38	3	87
117	MemberScope	Create() : ActionResult	74	2	9		3
118	MemberScope	Create(user) : ActionResult	52	5	16		19
119	MemberScope	Delete(int) : ActionResult	77	1	5		3
120	MemberScope	DeleteConfirmed(int) : ActionResult	71	1	8		5
121	MemberScope	Details(int) : ViewResult	77	1	5		3
122	MemberScope	Dispose(bool) : void	87	1	3		2
123	MemberScope	Edit(int) : ActionResult	77	1	5		3
124	MemberScope	Edit(user, HttpPostedFileBase) : ActionResult	59	4	11		11

125	MemberScope	Index() : ViewResult	73	2	12	3
126	MemberScope	LogOff() : ActionResult	82	1	3	3
127	MemberScope	LogOn() : ViewResult	87	1	2	2
128	MemberScope	LogOn(string, string) : ActionResult	47	3	21	29
129	MemberScope	UserController()	92	1	2	1
130	NamespaceScope	TouchForFood.Models	93	310	32	2
131	TypeScope	category	92	9	4	11
141	TypeScope	CategoryFilterVM	92	6	4	10
148	TypeScope	friendship	93	11	2	11
160	TypeScope	item	92	19	6	1
180	TypeScope	menu	93	13	5	14
194	TypeScope	menu_category	93	13	5	14
208	TypeScope	menu_item	93	13	4	13
222	TypeScope	order	92	25	8	27
248	TypeScope	order_item	93	13	3	13
262	TypeScope	order_status	92	17	5	18
280	TypeScope	restaurant	92	21	9	25
302	TypeScope	restaurant_user	93	11	3	11
314	TypeScope	RestaurantMetadata	94	9	1	9
324	TypeScope	review	93	21	6	21
346	TypeScope	ReviewMetadata	93	11	5	11
358	TypeScope	sysdiagram	93	11	1	11
370	TypeScope	touch_for_foodEntities	92	32	19	32
403	TypeScope	user	91	27	7	32
431	TypeScope	UserMetadata	93	15	5	15
447	TypeScope	waiter	93	13	5	14
448	MemberScope	first_name.get() : string	98	1	0	1
449	MemberScope	first_name.set(string) : void	95	1	0	1
450	MemberScope	id.get() : int	98	1	0	1
451	MemberScope	id.set(int) : void	95	1	0	1
452	MemberScope	last_name.get() : string	98	1	0	1
453	MemberScope	last_name.set(string) : void	95	1	0	1
454	MemberScope	orders.get() : ICollection<order>	98	1	2	1
455	MemberScope	orders.set(IICollection<order>) : void	95	1	2	1
456	MemberScope	restaurant.get() : restaurant	98	1	1	1
457	MemberScope	restaurant.set(restaurant) : void	95	1	1	1
458	MemberScope	resto_id.get() : int?	98	1	1	1
459	MemberScope	resto_id.set(int?) : void	95	1	1	1
460	MemberScope	waiter()	87	1	2	2
461	NamespaceScope	TouchForFood.Util.Category	70	8	12	1
462	TypeScope	CategoryUtil	70	8	12	1
463	MemberScope	CategoryUtil()	100	1	0	1
464	MemberScope	CategoryUtil()	94	1	1	1
465	MemberScope	filterListByMenu(menu) : IList<category>	58	6	12	13

Figure 10-5 Code Quality Report

10.6 Retrospective

In Iteration 3, we underachieved. Although we met our goals, our goals were probably not set high enough. Although this was with due reason, it was probably still not enough and we are not satisfied with our overall effort. However, on the other hand, the two stories planned were finished which makes this sprint a success.

10.6.1 Velocity

Sprint 3 velocity (story points): 11 story points
Cumulative velocity (story points): 13 story points
Average velocity (story points): 4.3 story points

Sprint 3 velocity (p-h): 20 person-hours
Cumulative velocity (p-h): 98 person-hours
Average velocity per sprint: 32.7 person-hours

Our sprint 3 velocity in story points greatly exceeded our previous velocities. However, this was due to the fact that the stories completed did not have much work left to be done. In terms of person-hours, we spent less time working on stories than in the previous two sprints.

10.6.2 Budget

Total person-hours budgeted to date: 2,484 person-hours
Total person-hours worked to date: 306.725 person-hours

We are grossly under-budget. This can be attributed to both a large over-estimation in the budget and also not enough time commitment from team members.

10.6.3 What was learned

The following positive and negative points were gathered from this sprint.

Positives:

- Completed 11 story points which is greater than any previous sprint
- Team members slowly becoming more familiar with the technologies in place

Negatives:

- Spent a lot of time on overhead and not so much on coding new stories and moving the project forward
- Did not commit enough time/effort to this sprint
- Still figuring out how to use Jira to its full potential

11 Iteration 4 Plan

11.1 Planned Activities

The stories planned for this sprint are CAP-25 – Order Food and CAP-29 – Manage Order. They are necessary to complete before moving forward with other aspects of the application as well as comprise the most important aspects of the Touch for Food application.

Table 11-1 Planned Stories

User Story ID	Total Story Points
CAP-25, CAP-29	18.00

The following defects were also planned. There are no story points associated to these defects.

Table 11-2 Planned Defects

Defect ID	Hours Estimated
CAP-98, CAP-100, CAP-111, CAP-125, CAP-156, CAP-160, CAP-168	15.00 hours

11.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations. Estimations were made for the worst case, most likely case, and best case. The worst case and best case are +30% and -30% of the most likely case. The expected case was then calculated with the formula:

$$\text{Expected case} = 1/6([\text{worst case}] + 4[\text{most likely case}] + [\text{best case}])$$

Table 11-3 Person-Hour Estimation

User Story ID	Worst Case	Most Likely Case	Best Case	Expected Case
CAP-25	89.7	69	48.3	69
CAP-29	75.4	58	40.6	58
Total (ph)	165.1	127	88.9	127
Velocity (ph/day)	11.8	9.1	6.4	9.1
Velocity (ph/team member/day)	1.3	1.0	0.7	1.0

12 Iteration 4 Report

12.1 Person-Hour Work Log

The following table shows the person-hour work log for each team member and activity during this iteration. This table was generated from the Jira management system. As planned, much more time was committed this sprint and it is reflected in this work log. These hours reflect time spent on all tasks.

Start Date: 18/Dec/12 End Date: 2/Jan/13 [Change] (UNREGISTERED)					Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total
Issue				Total	26.5h	10.667h	7.25h	23h	38.433h	18.5h	36.85h	32.533h	16.25h	209.983h
CAP-8	Documentation - Proposal			0.017h	0h	0h	0h	0h	0.017h	0h	0h	0h	0h	0.017h
CAP-10	Documentation - SRS			0.5h	0h	0h	0h	0h	0.5h	0h	0h	0h	0h	0.5h
CAP-11	Documentation - SAD			0.917h	0h	0h	0h	0h	0.917h	0h	0h	0h	0h	0.917h
CAP-14	Correspondence			14.9h	0h	1.633h	0.25h	0.25h	6.217h	0.5h	2.3h	2.25h	1.5h	14.9h
CAP-18	Meetings			24.333h	2h	1.75h	0h	3.75h	5.667h	2h	3.667h	5.5h	0h	24.333h
CAP-23	Setup			10.75h	0h	0h	0h	2.25h	3.25h	1h	1h	3.25h	0h	10.75h
CAP-25	Order Food			81.633h	24.25h	0h	0h	2h	9.517h	4.5h	18.617h	8h	14.75h	81.633h
CAP-29	Manage Order			42.55h	0h	7.283h	7h	11.75h	2h	10.5h	2h	2.017h	0h	42.55h
CAP-57	Documentation - Management			5.017h	0h	0h	0h	3h	0.017h	0h	0h	2h	0h	5.017h
CAP-98	Page 76-80			0.017h	0h	0h	0h	0h	0h	0h	0h	0.017h	0h	0.017h
CAP-100	Eliminate Redundancy			10.333h	0h	0h	0h	0h	10.333h	0h	0h	0h	0h	10.333h
CAP-111	Items getting added to menu instead of menu_items			8.5h	0h	0h	0h	0h	0h	1h	7.5h	0h	8.5h	
CAP-125	Ability to create a category and associate it to a menu from the menu page			6h	0h	0h	0h	0h	6h	0h	0h	0h	0h	6h
CAP-156	Cannot create new menu - Cannot implicitly convert type 'bool?' to 'bool'.			0.267h	0h	0h	0h	0h	0h	0h	0.267h	0h	0h	0.267h
CAP-160	Unable to add/remove items from a menu			4h	0h	0h	0h	0h	0h	2h	2h	0h	0h	4h
CAP-168	OrderStats Enum and Database Mismatch			0.25h	0.25h	0h	0h	0h	0h	0h	0h	0h	0h	0.25h

Figure 12-1 Person-Hour Work Log

12.2 Hour Burndown Chart

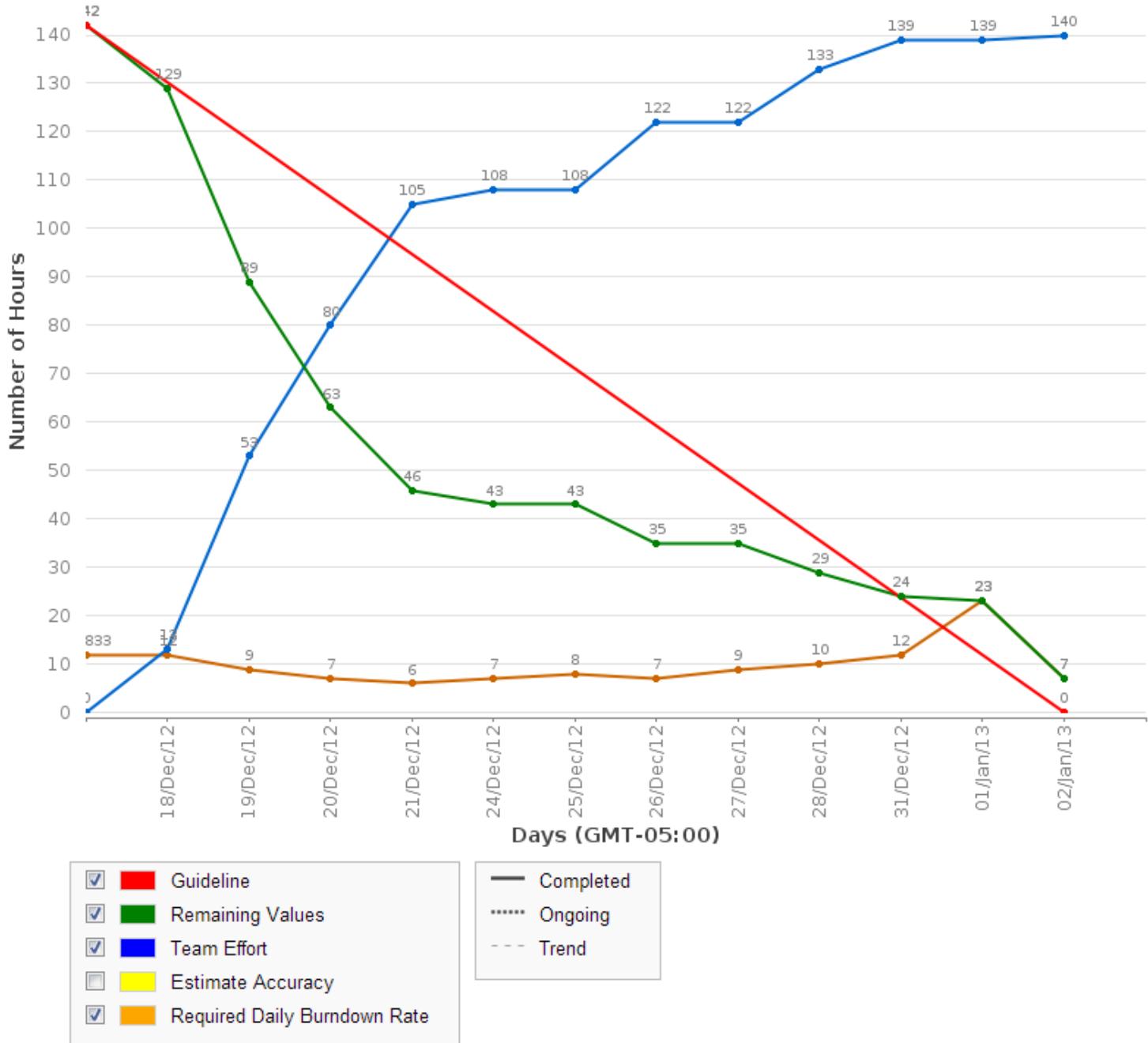


Figure 12-2 Hour Burndown Chart

For the first time, the estimated number of hours worked and the actual number of hours worked were very close. By the end of the sprint, we worked 7 hours less than initially planned. However, there was no delay in working and tasks were started at the beginning of the sprint. This is a very positive thing for the team.

12.3 Issue Burndown Chart

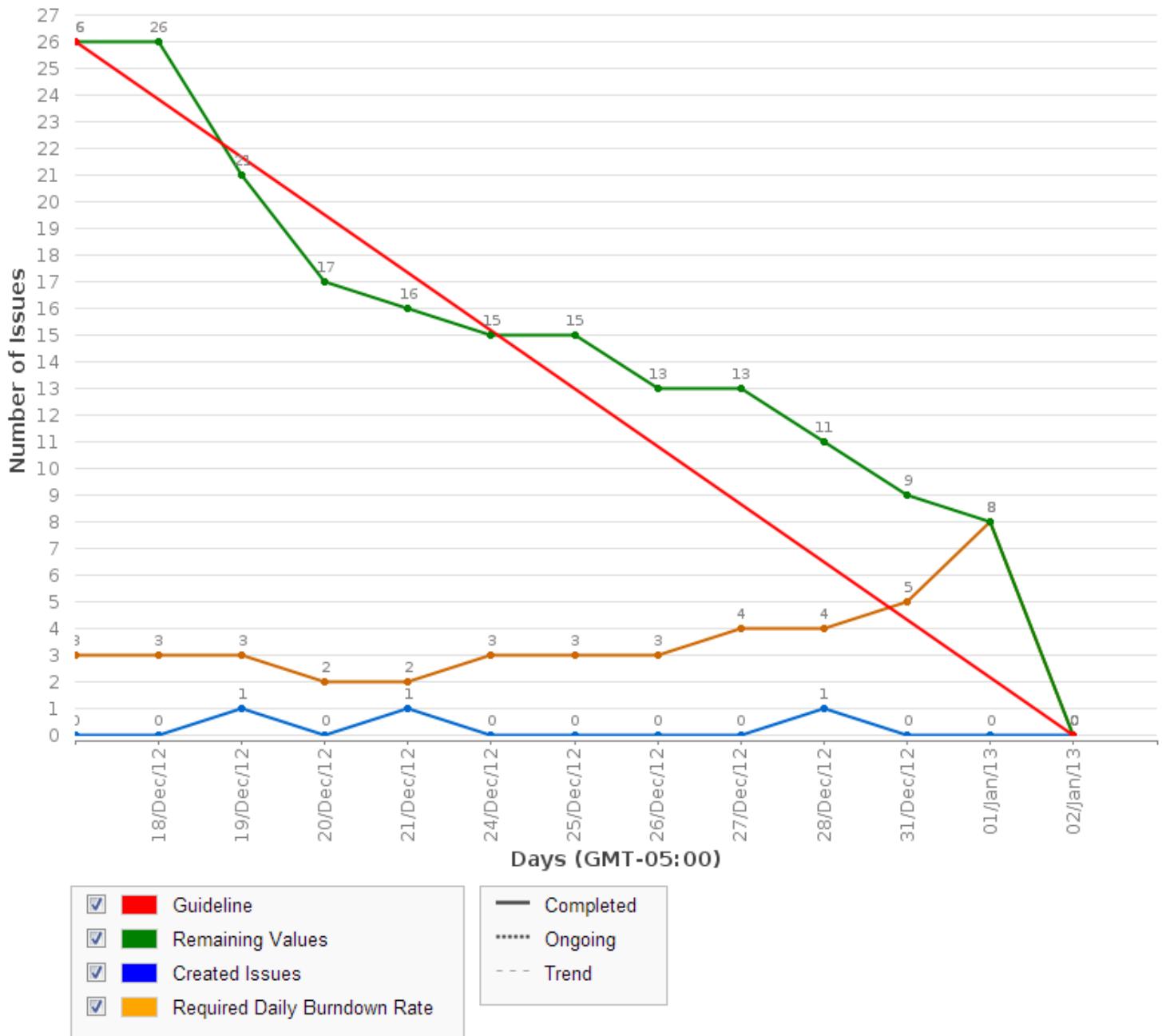


Figure 12-3 Issue Burndown Chart

As seen in the issue burndown chart, all of the issues planned were completed. Tasks were completed in a timely manner and burned down at a relatively steady rate.

12.4 Cumulative Flow Diagram

The following diagram shows the cumulative flow of the project so far based on the number of issues completed, in progress, and to do. Although we are completing issues, we are also finding bugs and adding them to the backlog which explains the increase of issues to do at the end of the sprint.

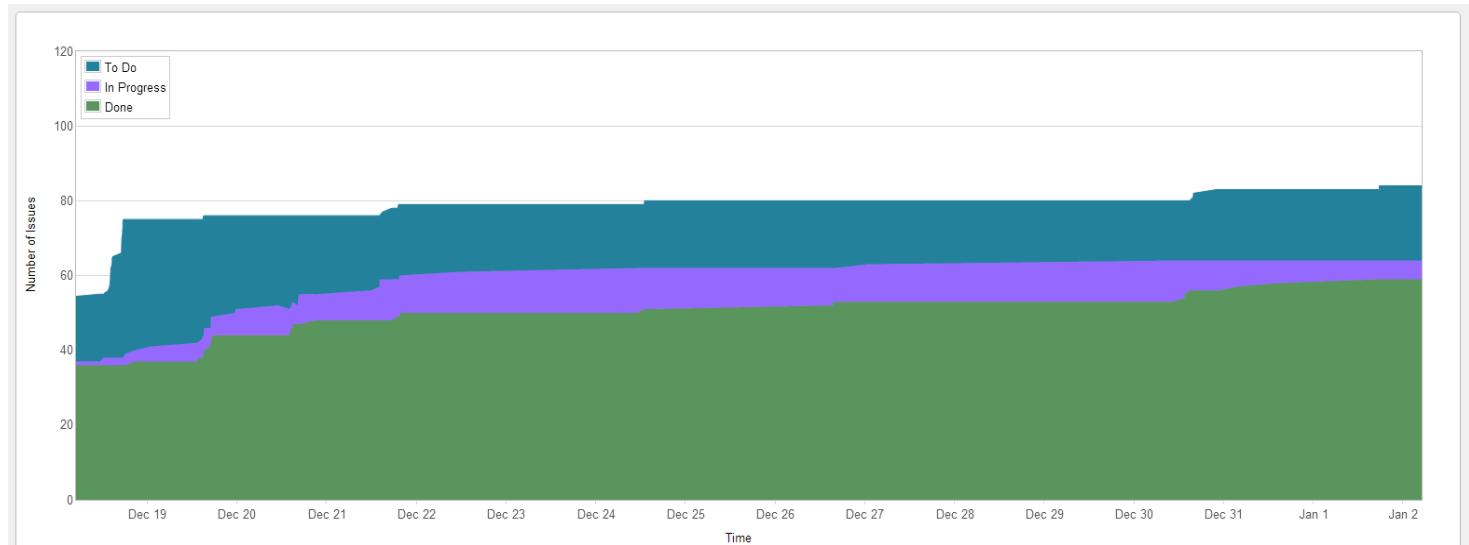


Figure 12-4 Cumulative Flow Diagram

12.5 Measurement Report

12.5.1 Code Quality Analysis

The following report was generated after analyzing the code. Three methods did not meet code quality goals by exceeding the expected number of lines of code. One of these was found in an earlier iteration and has not been yet corrected while the other two instances were introduced in this sprint. All of them have been entered as bugs and will be fixed in a future iteration.

Analysis tool used: Code Metrics Viewer
 Found at: <http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>
 Version: 1.5.3
 Last updated: 2/5/2012

Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
1						
2	ModuleScope TouchForFood.dll	83	673	108	3	
3	NamespaceScope TouchForFood	81	4	9	2	9
4	TypeScope MvcApplication	81	4	9	2	9
5	MemberScope Application_Start() : void	80	1	3		
6	MemberScope MvcApplication()	100	1	1		1
7	MemberScope RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
8	MemberScope RegisterRoutes(RouteCollection) : void	71	1	3		4
9	NamespaceScope TouchForFood.Attributes	84	2	5	3	4
10	TypeScope AjaxAttribute	84	2	5	3	4
11	MemberScope AjaxAttribute(bool)	87	1	1		2
12	MemberScope IsValidForRequest(ControllerContext, MethodInfo) : bool	83	1	4		2
13	NamespaceScope TouchForFood.Controllers	69	272	80	3	708
14	TypeScope CategoryController	72	17	19	3	54
28	TypeScope FriendshipController	69	20	29	3	42
39	TypeScope HomeController	80	3	9	3	4
42	TypeScope ItemController	70	22	36	3	63
57	TypeScope Menu_CategoryController	67	26	33	3	65
71	TypeScope Menu_ItemController	69	22	30	3	50
83	TypeScope MenuController	70	20	35	3	53
96	TypeScope Order_ItemController	66	30	43	3	66
109	TypeScope OrderController	61	61	50	3	157
129	TypeScope RestaurantController	76	14	15	3	35
140	TypeScope ReviewController	62	13	30	3	32
145	TypeScope UserController	66	24	38	3	87
146	MemberScope Create() : ActionResult	74	2	9		3
147	MemberScope Create(user) : ActionResult	52	5	16		19
148	MemberScope Delete(int) : ActionResult	77	1	5		3
149	MemberScope DeleteConfirmed(int) : ActionResult	71	1	8		5
150	MemberScope Details(int) : ViewResult	77	1	5		3
151	MemberScope Dispose(bool) : void	87	1	3		2
152	MemberScope Edit(int) : ActionResult	77	1	5		3

153 MemberScope	Edit(user, HttpPostedFileBase) : ActionResult		59	4	11		11
154 MemberScope	Index() : ViewResult		73	2	12		3
155 MemberScope	LogOff() : ActionResult		82	1	3		3
156 MemberScope	LogOn() : ViewResult		87	1	2		2
157 MemberScope	LogOn(string, string) : ActionResult		47	3	21		29
158 MemberScope	UserController()		92	1	2		1
159 NamespaceScope	TouchForFood.Models		92	338	36	2	397
160 TypeScope	category		92	9	4	1	11
170 TypeScope	CategoryFilterVM		83	9	6	1	16
179 TypeScope	friendship		93	11	2	1	11
191 TypeScope	item		92	17	5	1	18
209 TypeScope	ItemFilterVM		83	9	6	1	16
218 TypeScope	menu		93	13	5	1	14
232 TypeScope	menu_category		93	13	5	1	14
246 TypeScope	menu_item		93	15	7	1	16
262 TypeScope	MenuItemStatusHelper		100	0	0	1	0
263 TypeScope	MenuItemStatusHelper.MenuItemStatusEnum		100	0	0	1	0
264 TypeScope	order		89	29	11	1	36
291 TypeScope	order_item		92	20	5	1	22
312 TypeScope	OrderItemMetadata		100	1	0	1	1
314 TypeScope	OrderMetadata		100	1	0	1	1
316 TypeScope	OrderStatusHelper		66	24	7	1	44
322 TypeScope	OrderStatusHelper.OrderItemStatusEnum		100	0	0	1	0
323 TypeScope	OrderStatusHelper.OrderStatusEnum		100	0	0	1	0
324 TypeScope	restaurant		92	21	9	1	25
346 TypeScope	restaurant_user		93	11	3	1	11
358 TypeScope	RestaurantMetadata		94	9	1	1	9
368 TypeScope	review		93	21	6	1	21
390 TypeScope	ReviewMetadata		93	11	5	1	11
402 TypeScope	sysdiagram		93	11	1	1	11
414 TypeScope	touch_for_foodEntities		92	28	17	2	28
443 TypeScope	user		91	27	7	1	32
471 TypeScope	UserMetadata		93	15	5	1	15
487 TypeScope	waiter		93	13	5	1	14
488 MemberScope	first_name.get() : string		98	1	0		1
489 MemberScope	first_name.set(string) : void		95	1	0		1
490 MemberScope	id.get() : int		98	1	0		1
491 MemberScope	id.set(int) : void		95	1	0		1
492 MemberScope	last_name.get() : string		98	1	0		1
493 MemberScope	last_name.set(string) : void		95	1	0		1
494 MemberScope	orders.get() : ICollection<order>		98	1	2		1
495 MemberScope	orders.set(ICollection<order>) : void		95	1	2		1
496 MemberScope	restaurant.get() : restaurant		98	1	1		1
497 MemberScope	restaurant.set(restaurant) : void		95	1	1		1
498 MemberScope	resto_id.get() : int?		98	1	1		1
499 MemberScope	resto_id.set(int?) : void		95	1	1		1
500 MemberScope	waiter()		87	1	2		2
501 NamespaceScope	TouchForFood.Util.Category		70	8	12	1	15
502 TypeScope	CategoryUtil		70	8	12	1	15
503 MemberScope	CategoryUtil()		100	1	0		1
504 MemberScope	CategoryUtil()		94	1	1		1
505 MemberScope	filterListByMenu(menu) : IList<category>		58	6	12		13
506 NamespaceScope	TouchForFood.Util.Item		69	10	14	1	16
507 TypeScope	ItemUtil		69	10	14	1	16
508 MemberScope	filterListByItem(menu_category) : IList<item>		56	8	14		14
509 MemberScope	ItemUtil()		100	1	0		1
510 MemberScope	ItemUtil()		94	1	1		1
511 NamespaceScope	TouchForFood.Util.Order		80	16	16	1	34
512 TypeScope	OrderStatusUtil		95	2	1	1	2
515 TypeScope	OrderUtil		66	14	16	1	32
516 MemberScope	filterItem(menu_item) : item		77	1	4		3
517 MemberScope	filterMenuItem(order_item) : menu_item		74	1	5		4
518 MemberScope	mergeExistingOrderToDb(order) : void		48	10	14		23
519 MemberScope	OrderUtil()		100	1	0		1

520 MemberScope	OrderUtil()	94	1	1	1
521 NamespaceScope	TouchForFood.Util.Session	57	5	15	1
522 TypeScope	SessionUtil	57	5	15	1
523 MemberScope	getOpenOrder(user) : order	48	4	15	26
524 MemberScope	SessionUtil()	100	1	0	1
525 NamespaceScope	TouchForFood.Util.User	77	4	8	1
526 TypeScope	UserUtil	77	4	8	1
527 MemberScope	getAuthenticatedUser(HttpServletRequestBase) : user	66	2	8	7
528 MemberScope	UserUtil()	100	1	0	1
529 MemberScope	UserUtil()	94	1	1	1
530 NamespaceScope	TouchForFood.ViewModels	79	14	11	1
531 TypeScope	OrderItemVM	76	1	2	1
533 TypeScope	OrderVM	82	13	9	1
534 MemberScope	addItem(OrderItemVM) : void	94	1	2	1
535 MemberScope	getItemById(int) : OrderItemVM	71	4	5	5
536 MemberScope	orderItemVMs.get() : IList<OrderItemVM>	98	1	2	1
537 MemberScope	orderItemVMs.set(IList<OrderItemVM>) : void	95	1	2	1
538 MemberScope	OrderVM()	80	1	3	3
539 MemberScope	OrderVM(order)	80	1	3	3
540 MemberScope	OrderVM(order, IList<OrderItemVM>)	81	1	3	3
541 MemberScope	removeById(int) : void	78	2	2	3
542 MemberScope	removeItem(int) : void	94	1	2	1

Figure 12-5 Code Quality Report

12.6 Retrospective

Iteration 4 was a success because we completed all the stories planned for this sprint. The project started gaining momentum as both the stories completed were critical. There was a greater effort put forth by the entire team and this produced immediate results. All other issues planned were also completed.

12.6.1 Velocity

Sprint 4 velocity (story points): 18 story points

Cumulative velocity (story points): 31 story points

Average velocity (story points): 7.75 story points

Sprint 4 velocity (p-h): 124.2 person-hours

Cumulative velocity (p-h): 222.2 person-hours

Average velocity per sprint: 55.6 person-hours

In sprint 4, the team's velocity in story points increased once again compared to previous sprints. We attribute this to simply more time being spent on coding than previously. The team's velocity in person-hours also increased. It is clear that as we dedicate more person-hours to the stories planned, the results will follow.

12.6.2 Budget

Total person-hours budgeted to date: 2,898 person-hours

Total person-hours worked: 516.705 person-hours

Person-hours budgeted per sprint: 414 person-hours

Person-hours worked in Sprint 4: 209.98 person-hours

Although we are still under budget, the effort produced in this sprint exceeded previous sprints. There was an over-estimation in the budget so it is unlikely to ever match the budget, but an effort closer to 300 person-hours will probably be necessary in future sprints.

12.6.3 What was learned

The following positive and negative points were gathered from this sprint.

Positives:

- Completed 18 story points which is greater than any previous sprint
- Greater team effort and contribution overall
- Critical stories were completed
- More organized configuration management
- Team members more familiar with the technologies in place

Negatives:

- Still need to catch up to get back on schedule

13 Iteration 5 Plan

In iteration 5, the plan is to make up for a lack of quality or thoroughness in past sprints. The story point count is low so that the team can focus on catching up on testing, documenting and amending defects. In addition, the user interface will be implemented throughout the application during this sprint.

13.1 Planned Activities

Tables featured in this section of the document were created with JIRA [2].

13.1.1 Tasks

Table 13-1 Iteration 5 Planned Tasks

Key	Issue Type	Priority	Assignee	Original Estimate (hours)
CAP-164	Task	Major	Katrina Anderson	0.00
CAP-165	Sub-task	Major	Katrina Anderson	10.00
CAP-166	Sub-task	Major	Katrina Anderson	10.00
CAP-172	Sub-task	Major	Katrina Anderson	1.00
CAP-57	Task	Major	<i>Unassigned</i>	0.00
CAP-173	Sub-task	Minor	Mikhail Levkovsky	1.00
CAP-174	Sub-task	Minor	Mikhail Levkovsky	1.00
CAP-175	Sub-task	Minor	Mikhail Levkovsky	1.00
CAP-176	Sub-task	Minor	Mikhail Levkovsky	1.00
CAP-177	Sub-task	Major	Patrick Modafferri	1.00
CAP-178	Sub-task	Minor	Patrick Modafferri	1.00
CAP-180	Sub-task	Minor	Josh Hum	1.00
CAP-181	Sub-task	Minor	Josh Hum	1.00
CAP-182	Sub-task	Minor	Katrina Anderson	1.00
CAP-183	Sub-task	Minor	Josh Hum	1.50
CAP-184	Sub-task	Minor	Josh Hum	1.50
CAP-185	Sub-task	Minor	Josh Hum	1.50
CAP-186	Sub-task	Minor	Josh Hum	2.00
CAP-187	Sub-task	Minor	Josh Hum	2.00
CAP-188	Sub-task	Minor	Josh Hum	2.00
CAP-189	Sub-task	Minor	Josh Hum	5.00
CAP-11	Task	Major	<i>Unassigned</i>	0.00
CAP-215	Sub-task	Minor	Katrina Anderson	2.00
CAP-162	Task	Major	Josh Hum, Patrick Modafferri & Matt Tam	0.00
CAP-197	Sub-task (CAP-162)	Major	Josh Hum	2.00
CAP-202	Sub-task (CAP-162)	Major	Patrick Modafferri	6.00
CAP-208	Sub-task (CAP-162)	Major	Patrick Modafferri	5.00
CAP-209	Sub-task (CAP-162)	Major	Josh Hum, Patrick Modafferri & Matt Tam	16.00
CAP-210	Sub-task (CAP-162)	Major	Josh Hum, Patrick Modafferri & Matt Tam	16.00

CAP-211	Sub-task (CAP-162)	Major	Patrick Modafferi	4.00
CAP-212	Sub-task (CAP-162)	Major	Patrick Modafferi	18.00
CAP-213	Sub-task (CAP-162)	Major	Matthew Tam	16.00
CAP-214	Sub-task (CAP-162)	Major	Josh Hum, Patrick Modafferi & Matt Tam	8.00
CAP-216	Sub-task (CAP-162)	Major	Josh Hum, Patrick Modafferi & Matt Tam	10.00
CAP-217	Sub-task (CAP-162)	Major	Josh Hum	4.00
CAP-218	Sub-task (CAP-162)	Major	Josh Hum	6.00
CAP-230	Sub-task (CAP-162)	Major	Josh Hum	4.00
CAP-231	Sub-task (CAP-162)	Major	<i>Unassigned</i>	5.00
CAP-198	Task	Major	Mikhail Levkovsky, Katrina Anderson & Cristian Asenjo	0.00
CAP-199	Sub-task (CAP-198)	Major	Katrina Anderson	3.00
CAP-200	Sub-task (CAP-198)	Major	Katrina Anderson	3.00
CAP-201	Sub-task (CAP-198)	Major	Mikhail Levkovsky	3.00
CAP-203	Sub-task (CAP-198)	Major	Cristian Asenjo	16.00
CAP-204	Sub-task (CAP-198)	Major	Mikhail Levkovsky	3.00
CAP-205	Sub-task (CAP-198)	Major	Mikhail Levkovsky	5.00
CAP-206	Sub-task (CAP-198)	Major	Katrina Anderson	6.00
CAP-207	Sub-task (CAP-198)	Major	Mikhail Levkovsky	1.00
				TOTAL 207.50

13.1.2 Stories

In this iteration, two stories are planned to be resolved, for a total of six story points.

Table 13-2 Iteration 5 Planned Stories

Key	Issue Type	Priority	Assignee	Story Points	Original Estimate (hours)
CAP-40	Story	Critical	Cristian Asenjo & Cynthia Donato	3.00	0.00
CAP-163	Sub-task (CAP-40)	Major	Cristian Asenjo & Cynthia Donato		16.00
CAP-194	Sub-task (CAP-40)	Major	Cristian Asenjo & Cynthia Donato		4.00
CAP-196	Sub-task (CAP-40)	Major	Cristian Asenjo & Cynthia Donato		12.00
CAP-192	Story	Blocker	Christian Daher & Ryan Nasr	3.00	0.00
CAP-219	Sub-task (CAP-192)	Major	Ryan Nasr		8.00
CAP-220	Sub-task (CAP-192)	Major	Ryan Nasr		8.00
CAP-221	Sub-task (CAP-192)	Major	Christian Daher & Ryan Nasr		20.00
CAP-222	Sub-task (CAP-192)	Major	Ryan Nasr		6.00
CAP-224	Sub-task (CAP-192)	Major	Christian Daher & Ryan Nasr		4.00
CAP-225	Sub-task (CAP-192)	Major	Christian Daher & Ryan Nasr		16.00
CAP-244	Sub-task (CAP-192)	Major	Christian Daher & Ryan Nasr		2.00
CAP-245	Sub-task (CAP-192)	Major	Christian Daher & Ryan Nasr		2.00
				TOTAL 6.00	98.00

13.1.3 Defects

In this iteration, twelve defects are planned to be resolved. Blocker or Critical category defects can be added to the iteration at any time.

Table 13-3 Iteration 5 Planned Defects

Key	Issue Type	Priority	Assignee	Reporter	Original Estimate (hours)
CAP-85	Bug	Critical	Ryan Nasr	Cristian Asenjo	0.50
CAP-108	Bug	Minor	Christian Daher	Christian Daher	4.00
CAP-109	Bug	Major	Mikhail Levkovsky	Christian Daher	4.00
CAP-128	Bug	Trivial	Mikhail Levkovsky	Patrick Modafferri	2.00
CAP-158	Bug	Minor	Mikhail Levkovsky	Katrina Anderson	3.00
CAP-159	Bug	Major	Ryan Nasr	Matthew Tam	0.50
CAP-161	Bug	Critical	Patrick Modafferri	Cynthia Donato	0.17
CAP-169	Bug	Major	Cynthia Donato	Katrina Anderson	5.00
CAP-170	Bug	Trivial	Katrina Anderson	Katrina Anderson	0.25
CAP-190	Bug	Major	Ryan Nasr	Patrick Modafferri	4.00
CAP-191	Bug	Minor	Cristian Asenjo	Cristian Asenjo	4.00
CAP-193	Bug	Minor	Patrick Modafferri	Patrick Modafferri	6.00
CAP-226	Sub-task (CAP-193)	Major	Patrick Modafferri	Patrick Modafferri	0.00
CAP-227	Sub-task (CAP-193)	Major	Patrick Modafferri	Patrick Modafferri	0.00
CAP-228	Sub-task (CAP-193)	Major	Patrick Modafferri	Patrick Modafferri	0.00
					TOTAL 33.42

13.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations.

Table 13-4 Iteration 5 Person-Hour Estimation

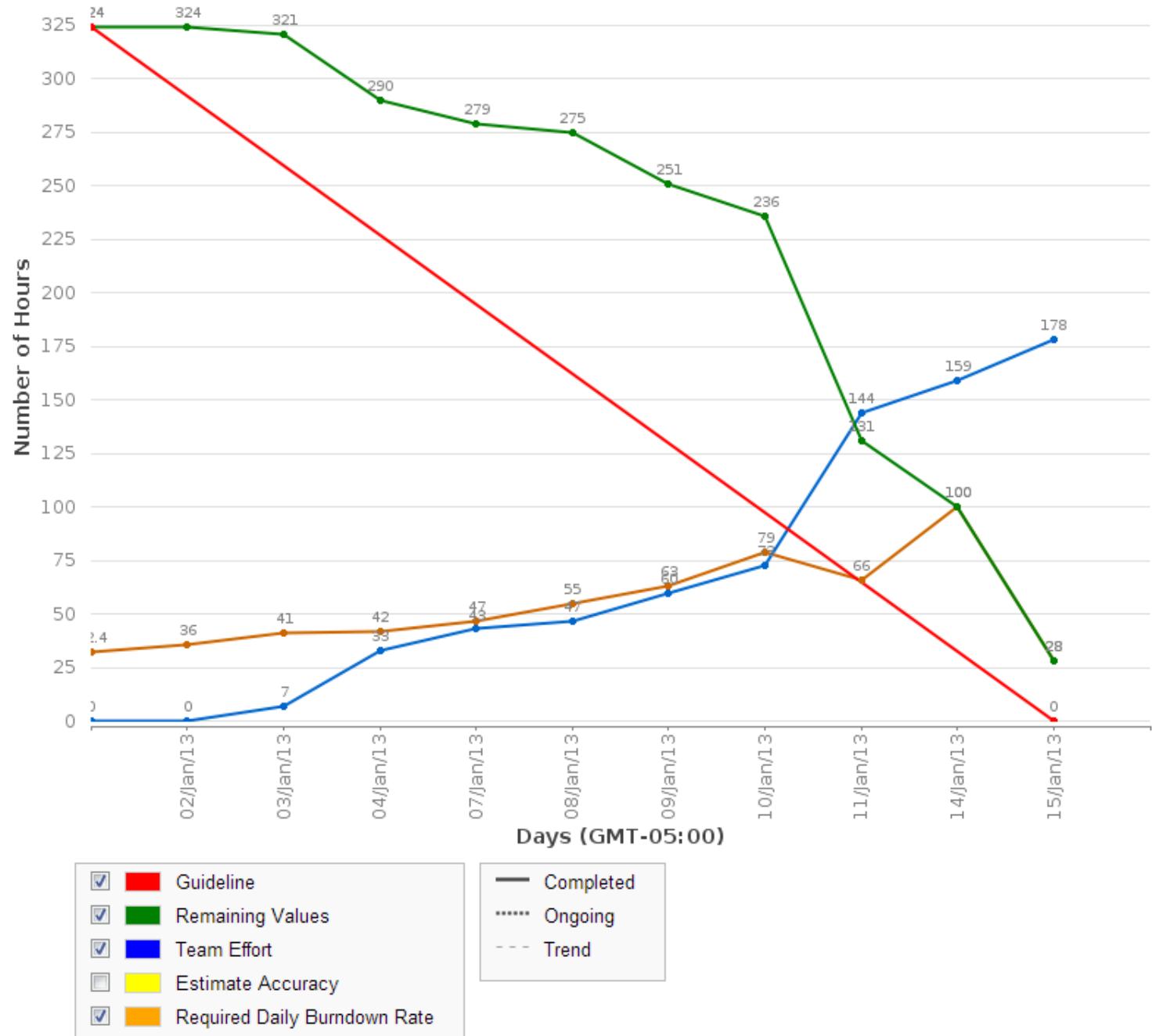
Description	Worst Case	Most Likely Case	Best Case	Expected Case
Tasks	269.75	207.50	145.25	207.50
Stories	127.40	98.00	68.60	98.00
Defects	43.45	33.42	23.39	33.42
Total(ph)	440.60	338.92	237.24	338.92
Velocity(ph/day)	31.47	24.21	16.95	24.21
Velocity (ph/team member/day)	3.50	2.69	1.88	2.69

14 Iteration 5 Report

14.1 Person-Hour Work Log

Start Date: 2/Jan/13 End Date: 15/Jan/13 [Change] (UNREGISTERED)		Total	Christian Dahir	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total
Issue		Total	21.833h	27.517h	24.767h	52.008h	38.292h	30.767h	34.45h	41.9h	17.633h	289.167h
CAP-8	Documentation - Proposal	↑ 0.017h	0h	0h	0h	0h	0.017h	0h	0h	0h	0h	0.017h
CAP-11	Documentation - SAD	↑ 2.5h	0h	0h	0h	1h	0h	0h	1.5h	0h	2.5h	
CAP-14	Correspondence	↑ 34.467h	1.083h	2.467h	4.583h	5.692h	9.925h	2.15h	3.067h	4.5h	1h	34.467h
CAP-18	Meetings	↑ 49.417h	2.5h	3.75h	5h	8.167h	6.917h	5.333h	9.083h	8.667h	0h	49.417h
CAP-23	Setup	↑ 13.917h	0h	0h	3.5h	0.333h	2.75h	1h	0.667h	3.167h	2.5h	13.917h
CAP-35	Manage Personal Profile	↑ 0.033h	0h	0h	0h	0h	0h	0h	0h	0.033h	0h	0.033h
CAP-40	Sign in	↑ 5.417h	0h	5.417h	0h	0h	0h	0h	0h	0h	0h	5.417h
CAP-57	Documentation - Management	↑ 25.283h	0h	0h	0h	19.75h	2.017h	0h	3.017h	0.5h	0h	25.283h
CAP-67	Add unit tests project into main TFF project	↑ 4h	0h	4h	0h	0h	0h	0h	0h	0h	0h	4h
CAP-85	Creating a review should use the last visit (i.e. the order) you created	↑ 0.333h	0h	0h	0h	0h	0h	0h	0h	0h	0.333h	0.333h
CAP-98	Page 76-80	↓ 0.017h	0h	0h	0h	0h	0h	0h	0.017h	0h	0h	0.017h
CAP-108	Activate/disable menu from index page of menu	↓ 2.5h	2h	0h	0h	0h	0h	0h	0h	0.5h	0h	2.5h
CAP-109	Soft Deleting a Menu_Category when there is a Menu_Item attached to it	↑ 4.933h	0h	0h	0h	0.183h	0h	0h	3.5h	1.25h	0h	4.933h
CAP-158	SAD Section 3 Invalid	↓ 2h	0h	0h	0h	0h	0h	2h	0h	0h	0h	2h
CAP-159	Database Data Type for Price and Total	↑ 0.5h	0h	0h	0h	0h	0h	0h	0h	0.5h	0.5h	0.5h
CAP-161	Setting Order Status to editing changes status to 6 instead of 3	↑ 0.833h	0h	0h	0.667h	0h	0h	0h	0.167h	0h	0h	0.833h
CAP-162	Look and Feel	↑ 56.333h	0h	0h	17.883h	0h	22.283h	0h	16.167h	0h	0h	56.333h
CAP-164	Documentation - Test Report	↑ 15.25h	0h	0h	0h	15.25h	0h	0h	0h	0h	0h	15.25h
CAP-168	OrderStats Enum and Database Mismatch	↑ 0.25h	0.25h	0h	0h	0h	0h	0h	0h	0h	0h	0.25h
CAP-169	Optimistic offline locks logic missing from orders	↑ 7.583h	0h	0h	7.583h	0h	0h	0h	0h	0h	0h	7.583h
CAP-170	Test Plan Section 5 Inaccurate	↓ 0.417h	0h	0h	0h	0h	0.417h	0h	0h	0h	0h	0.417h
CAP-190	Order Status Enum should no longer be in Database	↑ 4.017h	2h	0h	0h	0h	0h	0h	0h	2.017h	0h	4.017h
CAP-191	Fix Decline Order/Menu Item Status to use enum-only status rather than an enum/DB status	↓ 1.033h	0h	1.033h	0h	0h	0h	0h	0h	0h	0h	1.033h
CAP-192	Manage Tables	↑ 22.05h	13h	0h	0h	0h	0h	0h	0.017h	9.033h	22.05h	
CAP-193	Menu Items don't get a price set to them	↓ 6.5h	0h	0h	0h	0h	0h	1.25h	5.25h	0h	6.5h	
CAP-198	Unit Testing Retroactive	↑ 22.517h	1h	10.667h	0h	0h	0h	0h	9.85h	0h	1h	22.517h
CAP-229	Soft Deleting Menu when categories or items are associated to it	↓ 2.517h	0h	0h	0.5h	0h	0h	2.017h	0h	0h	0h	2.517h
CAP-252	build break - order does not have a field table number	↑ 0.517h	0h	0h	0.267h	0h	0h	0h	0h	0.25h	0h	0.517h
CAP-253	Order status enum runtime error in drop down list display	↑ 1h	0h	0h	0h	0h	0h	0h	0h	1h	0h	1h
CAP-259	Edit Order has a compilation error	↑ 0.167h	0h	0h	0h	0h	0h	0h	0.167h	0h	0h	0.167h
CAP-262	User edit not working	↑ 2.6h	0h	0.1h	2.5h	0h	0h	0h	0h	0h	0h	2.6h
CAP-275	Should not be able to decline an order that is in edit	↑ 0.25h	0h	0.083h	0.167h	0h	0h	0h	0h	0h	0h	0.25h

Figure 14-1 Person-Hour Work Log

14.2 Hour Burndown Chart**Figure 14-2 Hour Burndown Chart**

Almost 300 hours were spent working in Iteration 5. The effort was steady as can be seen by this graph with quite a few hours being worked from the beginning of the sprint.

14.3 Issue Burndown Chart

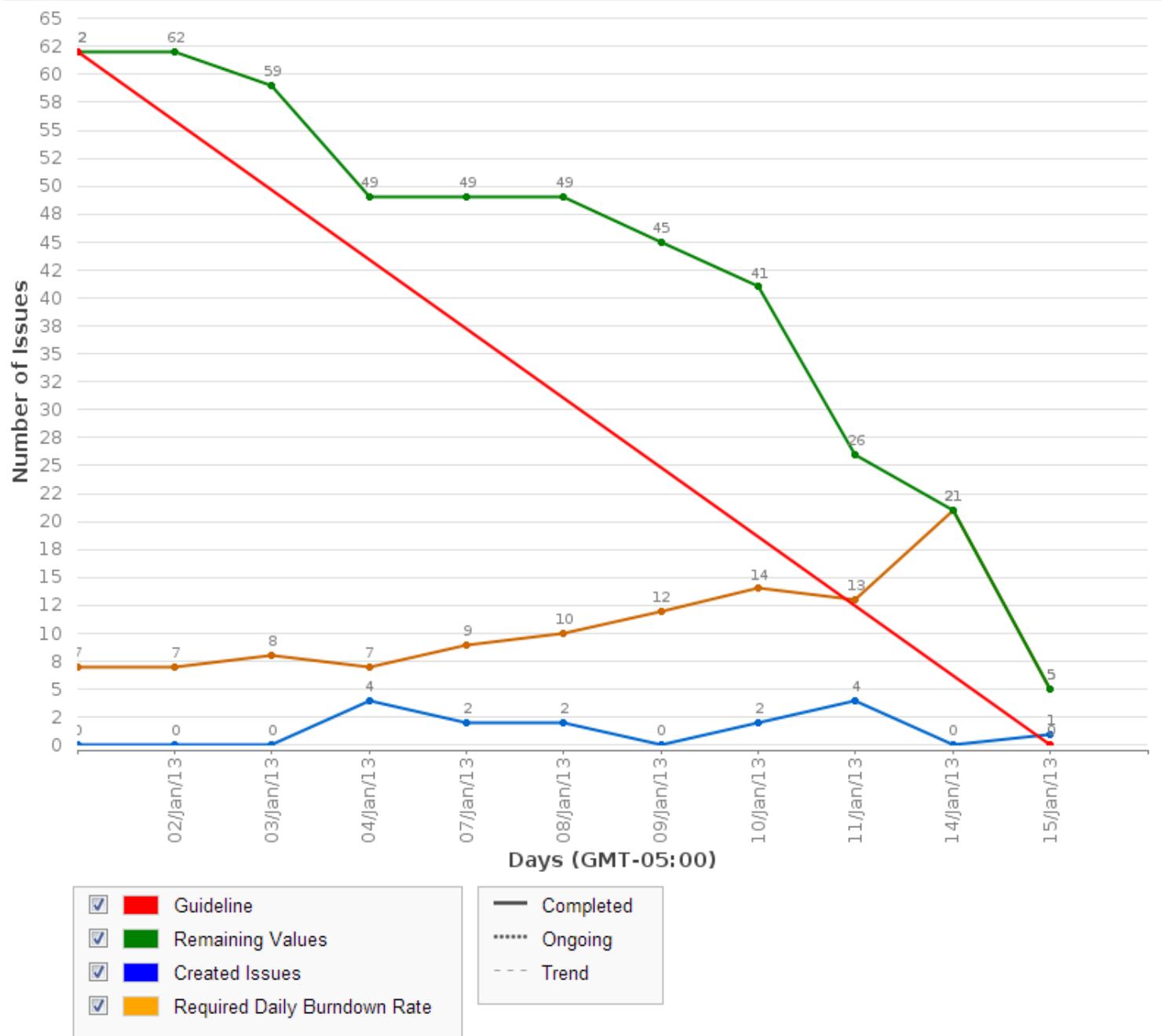


Figure 14-3 Issue Burndown Chart

As seen in the above chart, issues were burned down in correspondence with the amount of effort put in. Again, almost all the issues planned in this sprint were completed. CAP-40 – Sign In was pushed to the next sprint because it was not yet completed.

14.4 Cumulative Flow Diagram

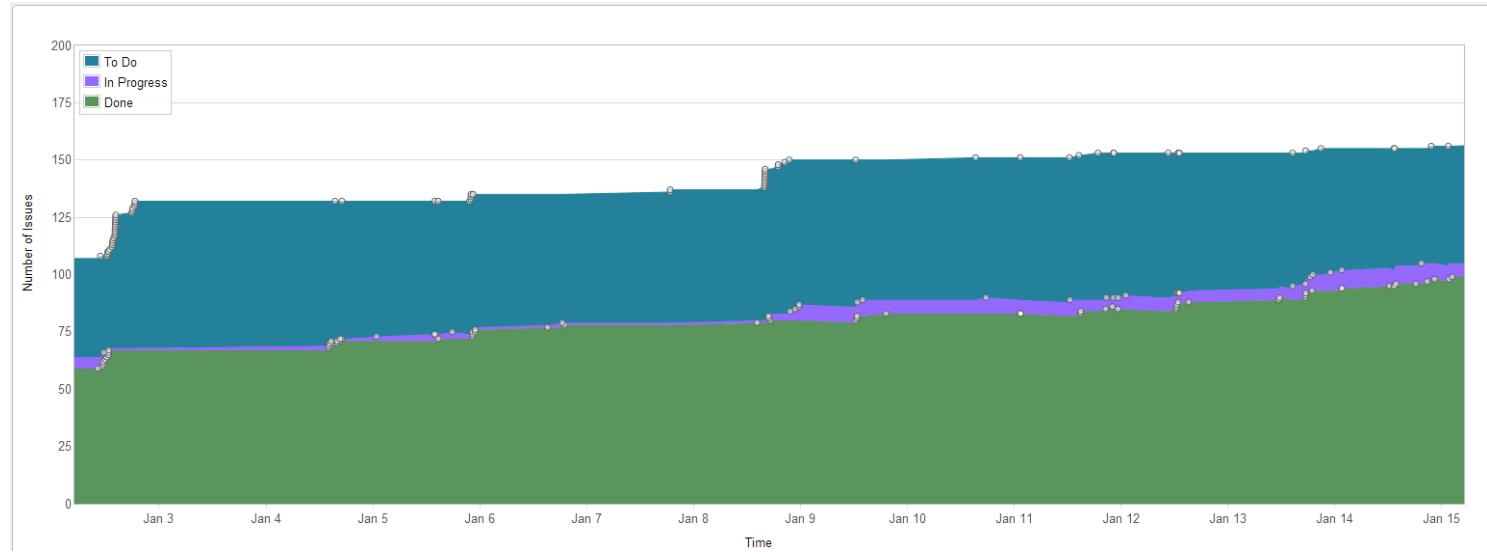


Figure 14-4 Cumulative Flow Diagram

Although new bugs are found and added as issues, our progress is steady in completing issues. Overall, almost 100 issues have been completed, leaving 50 more issues to complete until completing the project. Of course, we expect the number of remaining issues to increase as more bugs are added or new tasks defined.

14.5 Measurement Report

14.5.1 Code Quality Analysis

The following report was generated after analyzing the code. Four methods did not meet code quality goals by exceeding the expected number of lines of code. Three of these were found in an earlier iteration and has not been yet corrected while the new instance were introduced in this sprint. All of them have been entered as bugs and will be fixed in a future iteration.

Analysis tool used:	Code Metrics Viewer
Found at:	http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3
Version:	1.5.3
Last updated:	2/5/2012

Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
2	ModuleScope TouchForFood.Tests.dll	72	115	84	2	408
103	ModuleScope TouchForFood.dll	82	870	149	4	
104	NamespaceScope TouchForFood	81	4	9	2	9
105	TypeScope MvcApplication	81	4	9	2	9
106	MemberScope Application_Start() : void	80	1	3		3
107	MemberScope MvcApplication()	100	1	1		1
108	MemberScope RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
109	MemberScope RegisterRoutes(RouteCollection) : void	71	1	3		4
110	NamespaceScope TouchForFood.Attributes	84	2	5	3	4
111	TypeScope AjaxAttribute	84	2	5	3	4
112	MemberScope AjaxAttribute(bool)	87	1	1		2
113	MemberScope IsValidForRequest(ControllerContext, MethodInfo) : bool	83	1	4		2
114	NamespaceScope TouchForFood.Controllers	68	341	95	3	891
115	TypeScope CategoryController	71	19	28	3	60
129	TypeScope FriendshipController	69	20	29	3	42
140	TypeScope HomeController	80	3	9	3	4
143	TypeScope ItemController	70	25	39	3	74
159	TypeScope Menu_CategoryController	67	37	41	3	93
176	TypeScope Menu_ItemController	67	24	32	3	56
188	TypeScope MenuController	67	33	45	3	81
203	TypeScope Order_ItemController	64	32	45	3	74
216	TypeScope OrderController	60	71	49	3	184
237	TypeScope RestaurantController	72	16	24	3	41
248	TypeScope ReviewController	62	14	31	3	34
253	TypeScope TableController	67	23	42	3	55
265	TypeScope UserController	64	24	43	3	93
266	MemberScope Create() : ActionResult	74	2	9		3
267	MemberScope Create(user) : ActionResult	51	5	20		21
268	MemberScope Delete(int) : ActionResult	77	1	5		3
269	MemberScope DeleteConfirmed(int) : ActionResult	71	1	8		5
270	MemberScope Details(int) : ViewResult	77	1	5		3

271 MemberScope	Dispose(bool) : void
272 MemberScope	Edit(int) : ActionResult
273 MemberScope	Edit(user, HttpPostedFileBase) : ActionResult
274 MemberScope	Index() : ViewResult
275 MemberScope	LogOff() : ActionResult
276 MemberScope	LogOn() : ViewResult
277 MemberScope	LogOn(string, string) : ActionResult
278 MemberScope	UserController()
279 NamespaceScope	TouchForFood.Exceptions
280 TypeScope	ItemActiveException
281 MemberScope	ItemActiveException()
282 MemberScope	ItemActiveException(string)
283 NamespaceScope	TouchForFood.Mappers
284 TypeScope	MenuCategoryOM
287 TypeScope	MenuOM
288 MemberScope	delete(menu) : void
289 MemberScope	MenuOM()
290 NamespaceScope	TouchForFood.Models
291 TypeScope	category
303 TypeScope	CategoryFilterVM
312 TypeScope	friendship
324 TypeScope	item
344 TypeScope	ItemFilterVM
357 TypeScope	menu
375 TypeScope	menu_category
395 TypeScope	menu_item
417 TypeScope	MenuMetadata
425 TypeScope	order
455 TypeScope	order_item
476 TypeScope	OrderItemMetadata
478 TypeScope	OrderMetadata
480 TypeScope	OrderStatusHelper
486 TypeScope	OrderStatusHelper.OrderItemStatusEnum
487 TypeScope	OrderStatusHelper.OrderStatusEnum
488 TypeScope	restaurant
514 TypeScope	restaurant_user
526 TypeScope	RestaurantMetadata
536 TypeScope	review
558 TypeScope	ReviewMetadata
570 TypeScope	sysdiagram
582 TypeScope	table
597 TypeScope	TableMetadata
599 TypeScope	touch_for_foodEntities
632 TypeScope	user
668 TypeScope	UserMetadata
684 TypeScope	waiter
685 MemberScope	first_name.get() : string
686 MemberScope	first_name.set(string) : void
687 MemberScope	id.get() : int
688 MemberScope	id.set(int) : void
689 MemberScope	last_name.get() : string
690 MemberScope	last_name.set(string) : void
691 MemberScope	orders.get() : ICollection<order>
692 MemberScope	orders.set(IICollection<order>) : void
693 MemberScope	restaurant.get() : restaurant
694 MemberScope	restaurant.set(restaurant) : void
695 MemberScope	resto_id.get() : int?
696 MemberScope	resto_id.set(int?) : void
697 MemberScope	version.get() : int
698 MemberScope	version.set(int) : void
699 MemberScope	waiter()
700 NamespaceScope	TouchForFood.Util.Category
701 TypeScope	CategoryUtil
702 MemberScope	CategoryUtil()

87	1	3			2
77	1	6			3
59	4	12			11
73	2	13			3
76	1	5			4
87	1	2			2
46	3	24			32
92	1	2			1
97	2		1	2	2
97	2		1	2	2
100	1		1		1
98	1		1		1
73	14	7	1		17
77	5	6	1		7
69	9	7	1		10
61	8	7			9
100	1	0			1
92	409	45	2		497
92	11	4	1		13
83	9	6	1		16
93	11	2	1		11
92	19	5	1		20
80	18	16	1		34
92	17	5	1		18
92	19	5	1		20
92	21	6	1		22
94	7	4	1		7
88	31	12	1		52
92	20	5	1		22
100	1	0	1		1
100	1	0	1		1
66	24	7	1		44
100	0	0	1		0
100	0	0	1		0
91	25	10	1		30
93	11	3	1		11
94	9	1	1		9
93	21	6	1		21
93	11	5	1		11
93	11	1	1		11
91	14	8	1		19
100	1	0	1		1
92	32	19	2		32
91	35	9	1		40
93	15	5	1		15
93	15	5	1		16
98	1	0			1
95	1	0			1
98	1	0			1
95	1	0			1
98	1	0			1
95	1	2			1
95	1	2			1
98	1	1			1
95	1	1			1
98	1	1			1
95	1	1			1
98	1	0			1
95	1	0			1
87	1	2			2
70	8	12	1		15
70	8	12	1		15
100	1	0			1

703 MemberScope	CategoryUtil()		94	1	1	1
704 MemberScope	filterListByMenu(menu) : IList<category>		58	6	12	13
705 NamespaceScope	TouchForFood.Util.Html		68	10	17	1
706 TypeScope	HtmlDropDownExtensions		68	10	17	1
707 MemberScope	EnumDropDownListFor<TModel, TEnum>(this HtmlHelper<TModel>, Expression<TModel, TEnum> expression)		84	1	5	2
708 MemberScope	EnumDropDownListFor<TModel, TEnum>(this HtmlHelper<TModel>, Expression<TModel, TEnum> expression, string keyName)		59	3	13	10
709 MemberScope	GetEnumDescription<TEnum>(TEnum) : string		68	3	4	6
710 MemberScope	GetNonNullableModelType(ModelMetadata) : Type		70	2	3	6
711 MemberScope	HtmlDropDownExtensions()		88	1	1	1
712 NamespaceScope	TouchForFood.Util.Item		63	9	15	1
713 TypeScope	ItemUtil		63	9	15	1
714 MemberScope	filterListByItem(menu_category) : IList<item>		54	8	15	16
715 MemberScope	ItemUtil()		100	1	0	1
716 NamespaceScope	TouchForFood.Util.Order		82	18	19	1
717 TypeScope	OrderStatusUtil		95	2	1	1
720 TypeScope	OrderUtil		68	16	19	1
721 MemberScope	filterItem(menu_item) : item		77	1	4	3
722 MemberScope	filterMenuItem(order_item) : menu_item		74	1	5	4
723 MemberScope	filterTable(order) : table		77	1	5	3
724 MemberScope	filterUser(order) : user		77	1	5	3
725 MemberScope	filterWaiter(order) : waiter		77	1	5	3
726 MemberScope	mergeExistingOrderToDb(order) : void		48	9	13	23
727 MemberScope	OrderUtil()		100	1	0	1
728 MemberScope	OrderUtil()		94	1	1	1
729 NamespaceScope	TouchForFood.Util.Security		79	20	19	4
730 TypeScope	AES		67	14	11	1
740 TypeScope	CustomAuthorizationAttribute		70	6	7	4
743 TypeScope	SiteRoles		100	0	1	0
744 NamespaceScope	TouchForFood.Util.Session		56	5	15	1
745 TypeScope	SessionUtil		56	5	15	1
746 MemberScope	getOpenOrder(user) : order		47	4	15	26
747 MemberScope	SessionUtil()		100	1	0	1
748 NamespaceScope	TouchForFood.Util.User		77	4	8	1
749 TypeScope	UserUtil		77	4	8	1
750 MemberScope	getAuthenticatedUser(HttpServletRequestBase) : user		64	2	8	8
751 MemberScope	UserUtil()		100	1	0	1
752 MemberScope	UserUtil()		94	1	1	1
753 NamespaceScope	TouchForFood.ViewModels		76	24	13	1
754 TypeScope	OrderItemVM		76	1	2	1
756 TypeScope	OrderVM		76	23	11	1
757 MemberScope	addItem(OrderItemVM) : void		94	1	2	1
758 MemberScope	checkObjects() : void		64	8	3	7
759 MemberScope	getItemById(int) : OrderItemVM		71	4	5	5
760 MemberScope	orderItemVMs.get() : IList<OrderItemVM>		98	1	2	1
761 MemberScope	orderItemVMs.set(IList<OrderItemVM>) : void		95	1	2	1
762 MemberScope	OrderVM()		80	1	3	3
763 MemberScope	OrderVM(order)		73	2	3	5
764 MemberScope	OrderVM(order, IList<OrderItemVM>)		74	2	3	5
765 MemberScope	removeById(int) : void		78	2	2	3
766 MemberScope	removeItem(int) : void		94	1	2	1

Figure 14-5 Code Quality Report

14.6 Retrospective

In iteration 5, we completed 3 out of a planned 6 story points. Although this decreases our story point velocity, many other important issues were addressed and completed. As planned, we focused this sprint on testing, defects and missing documentation. The incomplete story was moved to the next sprint for completion.

14.6.1 Velocity

Sprint 5 velocity (story points): 3 story points

Cumulative velocity (story points): 34 story points

Average velocity (story points): 6.8 story points

Sprint 5 velocity (p-h): 289.17 person-hours

Cumulative velocity (p-h): 805.872 person-hours

Average velocity per sprint: 161.17 person-hours

It should be noted that velocity in person-hours was being calculated as just counting tasks with story points associated to them. From now on, it will be calculated to include all person-hours worked.

14.6.2 Budget

Total person-hours budgeted to date: 2,898 person-hours

Total person-hours worked to date: 805.872 person-hours

Person-hours budgeted per sprint: 414 person-hours

Person-hours worked in Sprint 5: 289.17 person-hours

The number of hours worked has increased each sprint and this was no exception. With our person-hour effort reaching almost 300 hours this iteration, many issues were taken care of and rapid progress is being made in terms of catching up to where we need to be in the project.

14.6.3 What was learned

The following positive and negative points were gathered from this sprint.

Positives:

- Completed 3 story points
- Greater team effort and contribution overall
- More organized and structured meetings
- Better processes established
- Many lingering issues (docs, defects, etc.) were dealt with

Negatives:

- Still need to catch up to get back on schedule

15 Iteration 6 Plan

In iteration 6, the goal is to continue working on increasing quality of our work while also moving the project forward. New user stories will be worked on while we will have one person dedicated to fixing bugs and one other person doing QA. To build off our last iteration, all new pages will include the UI theme that was implemented in sprint 5.

15.1 Planned Activities

Tables featured in this section of the document were created with JIRA [2].

Table 15-1 Planned Activities

Key	Issue Type	Priority	Assignee	Original Estimate (hours)
CAP-26	Story	Major	Josh Hum	
CAP-42	Story	Major	Ryan Nasr	
CAP-72	Sub-task	Major	Josh Hum	24
CAP-110	Task	Major	Unassigned	0
CAP-128	Bug	Trivial	Christian Daher	2
CAP-130	Sub-task	Major	Cynthia Donato	2
CAP-131	Sub-task	Major	Cristian Asenjo	4
CAP-134	Sub-task	Major	Mikhail Levkovsky	2
CAP-135	Sub-task	Major	Christian Daher	1
CAP-136	Sub-task	Major	Matthew Tam	4
CAP-137	Sub-task	Major	Katrina Anderson	3
CAP-139	Sub-task	Major	Ryan Nasr	3
CAP-141	Sub-task	Major	Josh Hum	3
CAP-142	Sub-task	Major	Patrick Modaffer	0.25
CAP-143	Sub-task	Major	Patrick Modaffer	0.25
CAP-144	Sub-task	Major	Cristian Asenjo	0.5
CAP-145	Sub-task	Major	Cynthia Donato	0.5
CAP-147	Sub-task	Major	Patrick Modaffer	0.5
CAP-148	Sub-task	Major	Matthew Tam	0.5
CAP-149	Sub-task	Major	Katrina Anderson	0.5
CAP-150	Sub-task	Major	Josh Hum	0.5
CAP-151	Sub-task	Major	Mikhail Levkovsky	0.5
CAP-152	Sub-task	Major	Christian Daher	0.5
CAP-153	Sub-task	Major	Ryan Nasr	0.083333333
CAP-154	Sub-task	Major	Ryan Nasr	0.083333333
CAP-155	Sub-task	Major	Katrina Anderson	0.5
CAP-169	Bug	Major	Cynthia Donato	5
CAP-194	Sub-task	Major	Unassigned	4
CAP-196	Sub-task	Major	Cristian Asenjo	12

CAP-198	Task	Major	Matthew Tam	0
CAP-199	Sub-task	Major	Matthew Tam	3
CAP-205	Sub-task	Major	Matthew Tam	5
CAP-206	Sub-task	Major	Matthew Tam	6
CAP-234	Sub-task	Minor	Unassigned	0
CAP-235	Bug	Critical	Christian Daher	2
CAP-236	Bug	Critical	Christian Daher	2
CAP-237	Bug	Critical	Christian Daher	2
CAP-238	Bug	Critical	Christian Daher	2
CAP-239	Bug	Critical	Christian Daher	2
CAP-240	Bug	Critical	Christian Daher	2
CAP-241	Bug	Critical	Christian Daher	2
CAP-242	Bug	Critical	Christian Daher	2
CAP-243	Bug	Critical	Christian Daher	2
CAP-246	Bug	Critical	Christian Daher	1
CAP-247	Task	Minor	Unassigned	0
CAP-250	Bug	Minor	Christian Daher	8
CAP-251	Bug	Minor	Unassigned	0
CAP-255	Improvement	Minor	Mikhail Levkovsky	4
CAP-261	Bug	Major	Ryan Nasr	8
CAP-263	Sub-task	Minor	Josh Hum	2
CAP-265	Sub-task	Major	Matthew Tam	10
CAP-267	Sub-task	Major	Patrick Modafferi	3
CAP-268	Sub-task	Major	Mikhail Levkovsky	3
CAP-269	Sub-task	Major	Christian Daher	3
CAP-270	Sub-task	Major	Cynthia Donato	3
CAP-271	Sub-task	Major	Ryan Nasr	3
CAP-272	Sub-task	Major	Cristian Asenjo	3
CAP-273	Sub-task	Major	Katrina Anderson	3
CAP-274	Bug	Major	Ryan Nasr	8
CAP-276	Bug	Minor	Cristian Asenjo	3
CAP-277	Bug	Critical	Christian Daher	3
CAP-278	Sub-task	Major	Josh Hum	16
CAP-281	Sub-task	Major	Mikhail Levkovsky	2
CAP-282	Sub-task	Major	Cynthia Donato	8
CAP-283	Sub-task	Major	Mikhail Levkovsky	1
CAP-284	Sub-task	Major	Katrina Anderson	8
CAP-285	Sub-task	Major	Katrina Anderson	2
CAP-286	Sub-task	Major	Cynthia Donato	2
CAP-287	Sub-task	Major	Cynthia Donato	2
CAP-288	Sub-task	Major	Unassigned	2

CAP-289	Sub-task	Major	Mikhail Levkovsky	2
CAP-290	Sub-task	Major	Mikhail Levkovsky	4
CAP-291	Sub-task	Major	Katrina Anderson	5
CAP-292	Sub-task	Major	Katrina Anderson	2
CAP-293	Sub-task	Major	Unassigned	3
CAP-294	Sub-task	Minor	Josh Hum	4
CAP-295	Sub-task	Minor	Mikhail Levkovsky	0.5
CAP-296	Sub-task	Minor	Josh Hum	0.75
CAP-297	Sub-task	Minor	Josh Hum	2
CAP-298	Sub-task	Minor	Josh Hum	2
CAP-299	Improvement	Major	Patrick Modafferri	0
CAP-300	Sub-task	Major	Ryan Nasr	5
CAP-301	Sub-task	Major	Ryan Nasr	8
CAP-302	Sub-task	Major	Christian Daher	4
CAP-303	Sub-task	Major	Ryan Nasr	12
CAP-304	Sub-task	Major	Ryan Nasr	6
CAP-305	Sub-task	Major	Unassigned	8
CAP-306	Sub-task	Major	Unassigned	8
CAP-307	Sub-task	Major	Unassigned	4
CAP-308	Bug	Major	Ryan Nasr	
				TOTAL 297.62

15.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations.

Description	Worst Case	Most Likely Case	Best Case	Expected Case
Activities	386.91	297.62	208.33	297.62
Total(ph)	386.91	297.62	208.33	297.62
Velocity(ph/day)	27.64	21.26	14.88	21.26
Velocity (ph/team member/day)	3.07	2.36	1.65	2.36

16 Iteration 6 Report

16.1 Person-Hour Work Log

Start Date: 16/Jan/13 End Date: 29/Jan/13 [Change] (UNREGISTERED)		Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total		
Issue		Total	17.917h	20h	22.567h	54.883h	37.483h	25.15h	21.5h	21.617h	30.417h	251.533h	
CAP-11	Documentation - SAD		1h	0h	0h	1h	0h	0h	0h	0h	0h	1h	
CAP-14	Correspondence		18.317h	0h	3.383h	1.3h	2.217h	6.833h	0.483h	1.717h	2.383h	0h	18.317h
CAP-18	Meetings		31.917h	1h	4h	1h	4.25h	3h	4.5h	4h	5.167h	5h	31.917h
CAP-23	Setup		3.667h	0h	0h	0.5h	0h	0h	0h	0h	3.167h	0h	3.667h
CAP-26	View Menu		32.083h	0h	0h	0h	32.083h	0h	0h	0h	0h	0h	32.083h
CAP-34	Call Waiter		34.75h	0h	0h	9h	0.333h	24.167h	0h	0.5h	0h	0.75h	34.75h
CAP-40	Sign in		5.65h	0h	5.65h	0h	0h	0h	0h	0h	0h	0h	5.65h
CAP-42	Restaurant Bill Management		26.117h	13h	0h	0h	0h	0h	0h	0h	0h	13.117h	26.117h
CAP-57	Documentation - Management		9.75h	0h	0h	0h	9.5h	0h	0h	0.25h	0h	0h	9.75h
CAP-110	Documentation - UIR		18.3h	0.417h	4.483h	3.033h	1.667h	0.983h	3.167h	1.7h	0.317h	2.533h	18.3h
CAP-162	Look and Feel		1h	0h	0h	0h	0h	1h	0h	0h	0h	0h	1h
CAP-164	Documentation - Test Report		11.567h	2h	1.517h	0.767h	2h	2.5h	0h	0.767h	0h	2.017h	11.567h
CAP-198	Unit Testing Retroactive		6h	0h	0h	0h	0h	6h	0h	0h	0h	0h	6h
CAP-235	FK Issue for Deleting Restaurants		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-236	FK Issue for Deleting Menu		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-237	FK constraint - User delete		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-238	FK Issue for Deleting Category		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-239	FK constraint - order delete		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-240	FK constraint - waiter delete		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-241	FK constraint - Review delete		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-242	FK Issue for Deleting Menu Item		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-243	FK constraint - Table delete		0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-246	Removing order item causes price to go below 0		1h	1h	0h	0h	0h	0h	0h	0h	0h	0h	1h
CAP-255	is_active and is_deleted field add extra complexity, factor that out		16.867h	0h	0h	5.967h	0h	0h	10.9h	0h	0h	0h	16.867h
CAP-261	Database context inconsistent		1h	0h	0h	0h	0h	0h	0h	0h	0h	1h	1h
CAP-264	User Interface Testing		8h	0h	0h	0h	0h	8h	0h	0h	0h	0h	8h
CAP-276	Custom routing has to be removed or fixed to work when logging in (CustomAuthorize is affecting it)		0.167h	0h	0.167h	0h	0h	0h	0h	0h	0h	0h	0.167h
CAP-277	Order half disappears		0.5h	0.5h	0h	0h	0h	0h	0h	0h	0h	0h	0.5h
CAP-280	Documentation - DB backup instructions		2h	0h	0h	0h	0h	2h	0h	0h	0h	0h	2h
CAP-299	Flow and Improvements		11.833h	0h	0h	0h	0h	0h	1.5h	10.333h	0h	0h	11.833h
CAP-308	The review controller should receive the order_id instead of the restaurant_id		0.083h	0h	0h	0h	0h	0h	0h	0.083h	0h	0h	0.083h
CAP-314	Code Review		8.833h	0h	0h	0h	2.833h	0h	0h	0h	6h	0h	8.833h

	CAP-315	null user role			0.8h	0h	0.8h	0h	0h	0h	0h	0h	0h	0h	0.8h
	CAP-319	Decline order not working			0.167h	0h	0h	0h	0h	0h	0h	0h	0.167h	0h	0.167h
	CAP-339	String Localization			0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0.017h	

Figure 16-1 Person-Hour Work Log

The above table shows the person-hours spent on various tasks in Sprint 6. The effort continued to be high as the team worked 251 hours in total. These hours include overhead tasks such as communication and setup.

16.2 Hour Burndown Chart

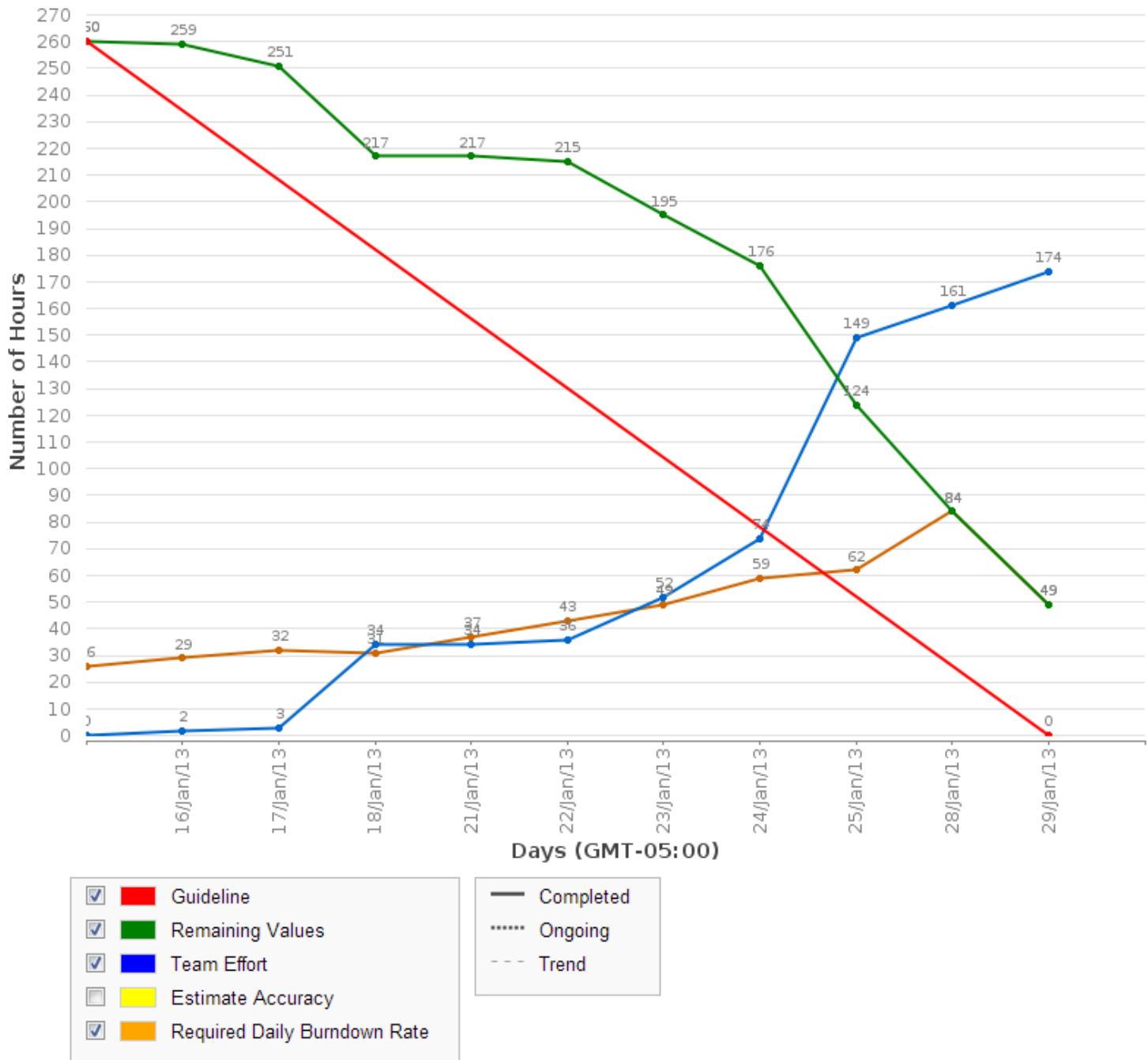


Figure 16-2 Hour Burndown Chart

Approximately 210 of the 260 estimated hours were completed. Some hours remained as a few tasks had to be carried over into Iteration 7. As seen in the chart, the team effort was pretty consistent with a little more effort being put in at the end of the sprint.

16.3 Issue Burndown Chart

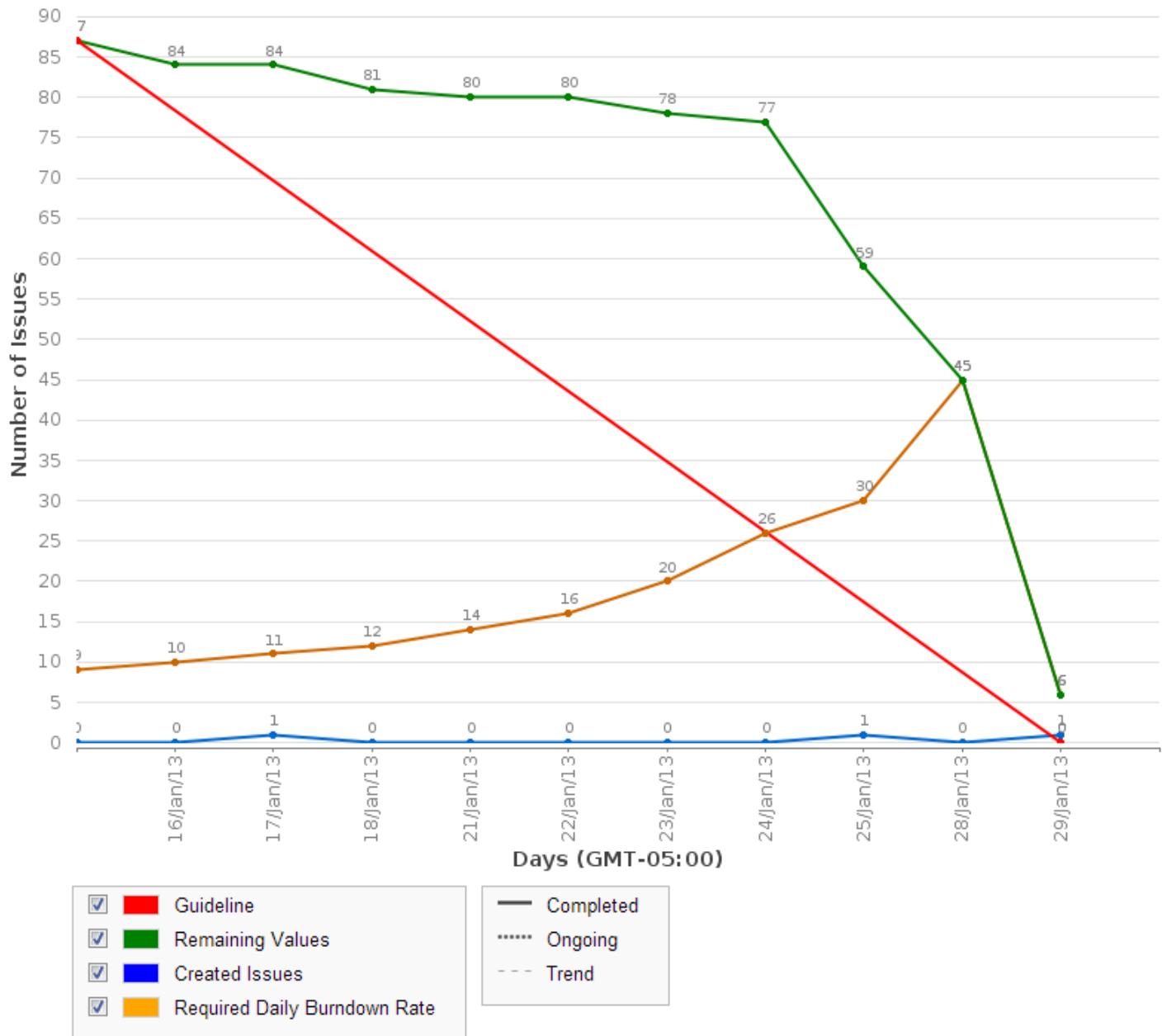


Figure 16-3 Issue Burndown Chart

Considering that 81 out of the 87 planned issues were burned down, we are happy with our progress. The two major issues not completed were CAP-350 – Search Menu and CAP-34 – Call Waiter. However, the majority of the work on them was done and they will be completed in Iteration 7. After discussing in our weekly meeting, it was noted that a few of the issues completed earlier in the sprint were not marked as resolved which is why the line is fairly flat and then shoots down at the end.

16.4 Cumulative Flow Diagram

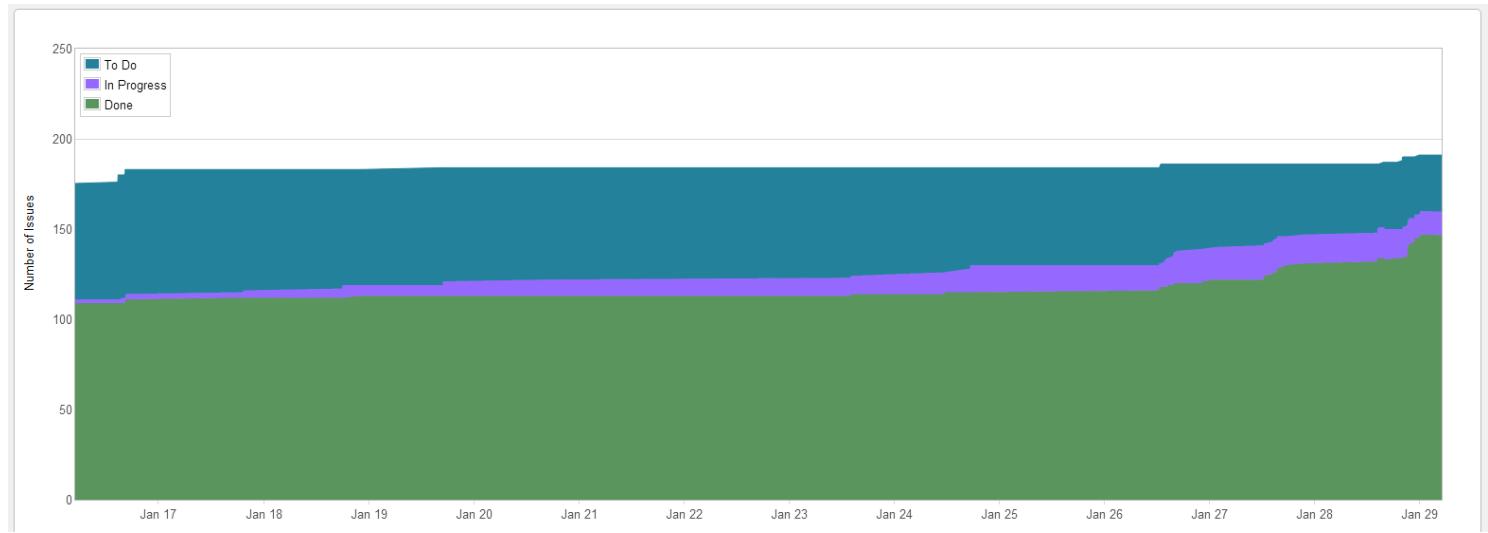


Figure 16-4 Cumulative Flow Diagram

Since each new task that we take on gets new subtasks created, there is always an increase in “to do” tasks at the end of the sprint as we start planning the next one. However, progress is continuous and the tasks that are done are starting to catch up to the tasks to do.

16.5 Measurement Report

16.5.1 Code Quality Analysis

The following report was generated after analyzing the code. Five methods did not meet code quality goals by exceeding the expected number of lines of code. Four of these were found in an earlier iteration and have not been yet corrected while the new instance was introduced in this sprint. All of them have been entered as bugs and will be fixed in a future iteration. Since it is a minor bug, the priority is not high for it.

Since our only issues have been with lines of code, we took note for the future that the following scale defines lines of code quality:

- 0: red
- 1 – 10: green
- 11 – 10: yellow
- 21+: red

Analysis tool used: Code Metrics Viewer

Found at: <http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>

Version: 1.5.3

Last updated: 2/5/2012

Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
ModuleScope	TouchForFood.Tests.dll	73	185	100	2	691
ModuleScope	TouchForFood.dll	80		178	4	
NamespaceScope	TouchForFood	80	4	9	2	11
TypeScope	MvcApplication	80	4	9	2	11
MemberScope	Application_Start() : void	80	1	3		3
MemberScope	MvcApplication()	100	1	1		1
MemberScope	RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
MemberScope	RegisterRoutes(RouteCollection) : void	66	1	3		6
NamespaceScope	TouchForFood.Attributes	84	2	5	3	4
TypeScope	AjaxAttribute	84	2	5	3	4
MemberScope	AjaxAttribute(bool)	87	1	1		2
MemberScope	IsValidForRequest(ControllerContext, MethodInfo)	83	1	4		2
NamespaceScope	TouchForFood.Controllers	66	502	110	3	
TypeScope	BillController	68	29	40	3	80
TypeScope	CategoryController	71	19	29	3	60
TypeScope	FriendshipController	69	20	29	3	42
TypeScope	HomeController	74	5	15	3	6
TypeScope	ItemController	70	25	40	3	74
TypeScope	Menu_CategoryController	67	41	31	3	85
TypeScope	Menu_ItemController	69	25	34	3	59
TypeScope	MenuController	67	39	44	3	81
TypeScope	Order_ItemController	66	28	40	3	66
TypeScope	OrderController	60	80	57	3	208
TypeScope	RestaurantController	72	16	25	3	41
TypeScope	ReviewController	62	14	31	3	34
TypeScope	SearchController	56	33	19	3	75
TypeScope	ServiceRequestController	62	67	45	3	132
TypeScope	TableController	66	35	46	3	69
TypeScope	UserController	64	26	45	3	101
MemberScope	Create() : ActionResult	74	2	9		3
MemberScope	Create(user) : ActionResult	51	5	20		21

MemberScope	Delete(int) : ActionResult	77	1	6		3
MemberScope	DeleteConfirmed(int) : ActionResult	71	1	9		5
MemberScope	Details(int) : ViewResult	77	1	6		3
MemberScope	Dispose(bool) : void	87	1	3		2
MemberScope	Edit(int) : ActionResult	77	1	6		3
MemberScope	Edit(user, HttpPostedFileBase) : ActionResult	50	6	22		19
MemberScope	Index() : ViewResult	73	2	13		3
MemberScope	LogOff() : ActionResult	76	1	5		4
MemberScope	LogOn() : ViewResult	87	1	2		2
MemberScope	LogOn(string, string) : ActionResult	46	3	24		32
MemberScope	UserController()	92	1	2		1
NamespaceScope	TouchForFood.Exceptions	97	4	1	2	4
TypeScope	AssociationExistsException	97	2	1	2	2
TypeScope	ItemActiveException	97	2	1	2	2
MemberScope	ItemActiveException()	100	1	1		1
MemberScope	ItemActiveException(string)	98	1	1		1
NamespaceScope	TouchForFood.Mappers	77	159	55	2	372
TypeScope	CategoryIM	84	6	6	2	10
TypeScope	CategoryOM	70	7	13	2	22
TypeScope	ItemIM	84	6	6	2	10
TypeScope	ItemOM	77	4	8	2	10
TypeScope	MenuCategoryIM	80	6	13	2	10
TypeScope	MenuCategoryOM	65	8	15	2	30
TypeScope	MenuIM	80	6	13	2	10
TypeScope	MenuItemIM	80	6	13	2	10
TypeScope	MenuItemOM	70	8	13	2	22
TypeScope	MenuOM	65	8	15	2	30
TypeScope	OrderIM	84	6	6	2	10
TypeScope	OrderOM	65	11	14	2	31
TypeScope	RestaurantIM	81	6	13	2	10
TypeScope	RestaurantOM	71	18	24	2	40
TypeScope	ReviewIM	84	6	6	2	10
TypeScope	ReviewOM	71	7	14	2	19
TypeScope	TableIM	84	6	6	2	10
TypeScope	TableOM	75	9	12	2	23
TypeScope	UserIM	84	6	6	2	10
TypeScope	UserOM	72	8	15	2	23
TypeScope	WaiterIM	84	6	6	2	10
TypeScope	WaiterOM	76	5	10	2	12
MemberScope	clearOrder(IICollection<order>) : void	75	2	8		3
MemberScope	delete(int) : int	67	1	5		7
MemberScope	WaiterOM()	100	1	1		1
MemberScope	WaiterOM(touch_for_foodEntities)	98	1	2		1
NamespaceScope	TouchForFood.Mappers.Abstract	90	5	1	1	8
TypeScope	GenericIM	86	2	1	1	4
TypeScope	GenericOM	94	3	1	1	4
MemberScope	delete(int) : int	100	1	0		0
MemberScope	GenericOM()	86	1	1		2
MemberScope	GenericOM(touch_for_foodEntities)	87	1	1		2
NamespaceScope	TouchForFood.Models	92	453	46	2	544
TypeScope	bill	92	21	7	1	22
TypeScope	category	92	11	4	1	13
TypeScope	CategoryFilterVM	83	9	6	1	16
TypeScope	friendship	93	11	2	1	11
TypeScope	item	92	19	5	1	20
TypeScope	ItemFilterVM	80	18	16	1	34
TypeScope	menu	92	17	5	1	18
TypeScope	menu_category	92	19	5	1	20
TypeScope	menu_item	92	21	6	1	22
TypeScope	MenuMetadata	94	7	4	1	7
TypeScope	order	89	35	14	1	58
TypeScope	order_item	92	24	6	1	26
TypeScope	OrderItemMetadata	100	1	0	1	1
TypeScope	OrderMetadata	100	1	0	1	1

TypeScope	OrderStatusHelper	65	25	7	1	44
TypeScope	OrderStatusHelper.OrderItemStatusEnum	100	0	0	1	0
TypeScope	OrderStatusHelper.OrderStatusEnum	100	0	0	1	0
TypeScope	restaurant	91	27	10	1	32
TypeScope	restaurant_user	93	11	3	1	11
TypeScope	RestaurantMetadata	94	9	1	1	9
TypeScope	review	92	25	7	1	25
TypeScope	ReviewMetadata	93	11	5	1	11
TypeScope	service_request	93	15	3	1	15
TypeScope	table	91	16	9	1	22
TypeScope	TableMetadata	100	1	0	1	1
TypeScope	touch_for_foodEntities	92	34	20	2	34
TypeScope	user	91	35	9	1	40
TypeScope	UserMetadata	93	15	5	1	15
TypeScope	waiter	93	15	5	1	16
MemberScope	first_name.get() : string	98	1	0		1
MemberScope	first_name.set(string) : void	95	1	0		1
MemberScope	id.get() : int	98	1	0		1
MemberScope	id.set(int) : void	95	1	0		1
MemberScope	last_name.get() : string	98	1	0		1
MemberScope	last_name.set(string) : void	95	1	0		1
MemberScope	orders.get() : ICollection<order>	98	1	2		1
MemberScope	orders.set(ICollection<order>) : void	95	1	2		1
MemberScope	restaurant.get() : restaurant	98	1	1		1
MemberScope	restaurant.set(restaurant) : void	95	1	1		1
MemberScope	resto_id.get() : int?	98	1	1		1
MemberScope	resto_id.set(int?) : void	95	1	1		1
MemberScope	version.get() : int	98	1	0		1
MemberScope	version.set(int) : void	95	1	0		1
MemberScope	waiter()	87	1	2		2
NamespaceScope	TouchForFood.Util.Bill	74	13	9	1	15
TypeScope	BillUtil	74	13	9	1	15
MemberScope	GetTotalAfterTax(bill) : decimal	78	3	3		2
MemberScope	GetTotalBeforeTax(bill) : decimal	85	1	3		2
MemberScope	GetTPS() : decimal	84	1	1		2
MemberScope	GetTVQ() : decimal	84	1	1		2
MemberScope	Update(ref bill) : void	62	7	9		7
NamespaceScope	TouchForFood.Util.Category	70	8	12	1	15
TypeScope	CategoryUtil	70	8	12	1	15
MemberScope	CategoryUtil()	100	1	0		1
MemberScope	CategoryUtil()	94	1	1		1
MemberScope	filterListByMenu(menu) : IList<category>	58	6	12		13
NamespaceScope	TouchForFood.Util.Html	69	16	26	1	39
TypeScope	HtmlDropDownExtensions	68	10	17	1	25
TypeScope	ImageActionLinkHelper	64	1	7	1	8
TypeScope	UrlUtils	75	5	4	1	6
MemberScope	ConvertRelativeUrlToAbsoluteUrl(string) : strin	67	4	4		5
MemberScope	UrlUtils()	100	1	0		1
NamespaceScope	TouchForFood.Util.Item	63	9	15	1	17
TypeScope	ItemUtil	63	9	15	1	17
MemberScope	filterListByItem(menu_category) : IList<item>	54	8	15		16
MemberScope	ItemUtil()	100	1	0		1
NamespaceScope	TouchForFood.Util.Order	82	22	20	1	49
TypeScope	OrderStatusUtil	95	2	1	1	2
TypeScope	OrderUtil	68	20	20	1	47
MemberScope	filterItem(menu_item) : item	77	1	4		3
MemberScope	filterMenuItem(order_item) : menu_item	74	1	5		4
MemberScope	filterTable(order) : table	77	1	5		3
MemberScope	filterUser(order) : user	77	1	5		3
MemberScope	filterWaiter(order) : waiter	77	1	5		3
MemberScope	mergeExistingOrderToDb(order) : void	48	9	13		23
MemberScope	OrderUtil()	100	1	0		1
MemberScope	OrderUtil()	94	1	1		1
MemberScope	UpdatePrice(ref order) : void	65	4	11		6

NamespaceScope	TouchForFood.Util.Search	66	8	19	1	43
TypeScope	SearchService	53	3	18	1	37
TypeScope	SearchViewModelHelper	78	5	9	1	6
MemberScope	PopulateSearchViewModel(menu_item) : Search	77	1	6		2
MemberScope	PopulateSearchViewModelList(IList<menu_item>) : Search	76	3	5		3
MemberScope	SearchViewModelHelper()	100	1	0		1
NamespaceScope	TouchForFood.Util.Security	79	20	19	4	73
TypeScope	AES	67	14	11	1	62
TypeScope	CustomAuthorizationAttribute	70	6	7	4	11
TypeScope	SiteRoles	100	0	1	1	0
NamespaceScope	TouchForFood.Util.ServiceRequest	100	1	0	1	1
TypeScope	ServiceRequestUtil	100	1	0	1	1
TypeScope	ServiceRequestUtil.ServiceRequestStatus	100	0	0	1	0
NamespaceScope	TouchForFood.Util.Session	55	5	15	1	30
TypeScope	SessionUtil	55	5	15	1	30
MemberScope	getOpenOrder(user) : order	46	4	15		29
MemberScope	SessionUtil()	100	1	0		1
NamespaceScope	TouchForFood.Util.User	77	5	11	1	12
TypeScope	UserUtil	77	5	11	1	12
MemberScope	getAuthenticatedUser(HttpContextBase) : user	64	2	8		8
MemberScope	isUserRole(SiteRoles, HttpContext) : bool	84	1	3		2
MemberScope	UserUtil()	100	1	0		1
MemberScope	UserUtil()	94	1	1		1
NamespaceScope	TouchForFood.ViewModels	81	38	13	1	56
TypeScope	OrderItemVM	76	1	2	1	4
TypeScope	OrderVM	76	23	11	1	32
TypeScope	SearchViewModel	92	14	0	1	20
MemberScope	catName.get() : string	98	1	0		1
MemberScope	catName.set(string) : void	95	1	0		1
MemberScope	description.get() : string	98	1	0		1
MemberScope	description.set(string) : void	95	1	0		1
MemberScope	menuItemId.get() : int	98	1	0		1
MemberScope	menuItemId.set(int) : void	95	1	0		1
MemberScope	menuName.get() : string	98	1	0		1
MemberScope	menuName.set(string) : void	95	1	0		1
MemberScope	metadata.get() : string	98	1	0		1
MemberScope	metadata.set(string) : void	95	1	0		1
MemberScope	name.get() : string	98	1	0		1
MemberScope	name.set(string) : void	95	1	0		1
MemberScope	SearchViewModel()	100	1	0		1
MemberScope	SearchViewModel(int, string, string, string, string)	70	1	0		7

Figure 16-5 Code Quality Report

16.6 Retrospective

In iteration 6, we completed 11 out of a planned 15 story points. The two stories CAP-350 – Search Menu and CAP-34 – Call Waiter were added to the next sprint as they were almost but not yet complete. We also introduced the official roles of bug basher and QA. From Sprint 6 and onward, everyone in the team will take a turn being one of these two positions. There was also one person appointed to working on the overall flow of the application. Code reviews were also established. Everyone works in their own branch and code has to be reviewed before being committed back to the trunk. There are three designated code reviewers per sprint although anyone can review code and leave feedback.

See Appendix B Glossary for definitions of Bug Basher, QA, and flow fixer.

16.6.1 Velocity

Sprint 6 velocity (story points): 11 story points

Cumulative velocity (story points): 45 story points

Average velocity (story points): 7.5 story points

Sprint 6 velocity (p-h): 251.53 person-hours

Cumulative velocity (p-h): 1,057.40 person-hours

Average velocity per sprint: 176.23 person-hours

Our average velocity in story points increased this iteration as did our average velocity in person-hours. The amount of effort is staying consistent above 200 hours. If we continue at this rate, we should be able to complete the project on time.

16.6.2 Budget

Total person-hours budgeted to date: 3,312 person-hours

Total person-hours worked to date: 1,057.40 person-hours

Person-hours budgeted per sprint: 414 person-hours

Person-hours worked in Sprint 6: 251.53 person-hours

Since the budget was over-estimated, our new goal is to work at least 200 hours per sprint. If this goal is reached, the project should be finished on time. In Iteration 6, we met this goal.

16.6.3 What was learned

The following positive and negative points were gathered from this sprint.

Positives:

- Completed 11 story points
- New roles such as QA, Bug masher, Flow fixer, and code reviewers improved the overall quality of the project
- UIR was created
- Back on schedule to finish the project in time

Negatives:

- None

17 Iteration 7 Plan

The goal of Iteration 7 is to continue our current progress on the project while also start preparing for presentations. We will be having meetings with our stakeholder and project supervisor during the sprint. We will also hand in deliverable 2 at the end of the sprint. Thus, the focus of the sprint is to make our documentation consistent and prepare for our meetings. We will continue to finish open stories and work on new stories. We will also have people working on QA, bug bashing, fixing flow, and reviewing code.

17.1 Planned Activities

Tables featured in this section of the document were created with JIRA [2].

Table 17-1 Planned Activities

Key	Issue Type	Priority	Assignee	Original Estimate (hours)
CAP-128	Bug	Trivial	Mikhail Levkovsky	2
CAP-234	Sub-task	Minor	Cynthia Donato	1
CAP-247	Task	Minor	Unassigned	0
CAP-251	Bug	Minor	Unassigned	0
CAP-255	Improvement	Minor	Cynthia Donato	4
CAP-263	Sub-task	Minor	Josh Hum	2
CAP-276	Bug	Minor	Cristian Asenjo	3
CAP-294	Sub-task	Minor	Josh Hum	4
CAP-295	Sub-task	Minor	Mikhail Levkovsky	0.5
CAP-296	Sub-task	Minor	Josh Hum	0.75
CAP-297	Sub-task	Minor	Josh Hum	2
CAP-298	Sub-task	Minor	Josh Hum	2
CAP-375	Sub-task	Minor	Josh Hum	32
CAP-26	Story	Major	Josh Hum	0
CAP-42	Story	Major	Ryan Nasr	0
CAP-110	Task	Major	Unassigned	0
CAP-130	Sub-task	Major	Cynthia Donato	2
CAP-131	Sub-task	Major	Cristian Asenjo	4
CAP-134	Sub-task	Major	Mikhail Levkovsky	2
CAP-135	Sub-task	Major	Christian Daher	1
CAP-136	Sub-task	Major	Matthew Tam	4
CAP-137	Sub-task	Major	Katrina Anderson	3
CAP-139	Sub-task	Major	Ryan Nasr	3
CAP-141	Sub-task	Major	Josh Hum	3
CAP-142	Sub-task	Major	Patrick Modafferri	0.25
CAP-143	Sub-task	Major	Patrick Modafferri	0.25
CAP-144	Sub-task	Major	Cristian Asenjo	0.5

CAP-145	Sub-task	Major	Cynthia Donato	0.5
CAP-147	Sub-task	Major	Patrick Modafferri	0.5
CAP-148	Sub-task	Major	Matthew Tam	0.5
CAP-149	Sub-task	Major	Katrina Anderson	0.5
CAP-150	Sub-task	Major	Josh Hum	0.5
CAP-151	Sub-task	Major	Mikhail Levkovsky	0.5
CAP-152	Sub-task	Major	Christian Daher	0.5
CAP-153	Sub-task	Major	Ryan Nasr	0.083333333
CAP-154	Sub-task	Major	Ryan Nasr	0.083333333
CAP-155	Sub-task	Major	Katrina Anderson	0.5
CAP-169	Bug	Major	Cynthia Donato	5
CAP-194	Sub-task	Major	Cristian Asenjo	4
CAP-196	Sub-task	Major	Cristian Asenjo	12
CAP-198	Task	Major	Katrina Anderson	0
CAP-199	Sub-task	Major	Katrina Anderson	3
CAP-205	Sub-task	Major	Katrina Anderson	5
CAP-206	Sub-task	Major	Katrina Anderson	6
CAP-261	Bug	Major	Ryan Nasr	8
CAP-265	Sub-task	Major	Matthew Tam	10
CAP-267	Sub-task	Major	Patrick Modafferri	3
CAP-268	Sub-task	Major	Mikhail Levkovsky	3
CAP-269	Sub-task	Major	Christian Daher	3
CAP-270	Sub-task	Major	Cynthia Donato	3
CAP-271	Sub-task	Major	Ryan Nasr	3
CAP-272	Sub-task	Major	Cristian Asenjo	3
CAP-273	Sub-task	Major	Katrina Anderson	3
CAP-274	Bug	Major	Mikhail Levkovsky	8
CAP-278	Sub-task	Major	Josh Hum	16
CAP-281	Sub-task	Major	Mikhail Levkovsky	2
CAP-282	Sub-task	Major	Katrina Anderson	8
CAP-283	Sub-task	Major	Mikhail Levkovsky	1
CAP-284	Sub-task	Major	Katrina Anderson	8
CAP-285	Sub-task	Major	Katrina Anderson	2
CAP-286	Sub-task	Major	Cynthia Donato	2
CAP-287	Sub-task	Major	Cynthia Donato	2
CAP-288	Sub-task	Major	Ryan Nasr	2
CAP-289	Sub-task	Major	Mikhail Levkovsky	2
CAP-290	Sub-task	Major	Mikhail Levkovsky	4
CAP-291	Sub-task	Major	Katrina Anderson	5
CAP-292	Sub-task	Major	Katrina Anderson	2
CAP-293	Sub-task	Major	Cynthia Donato	3

CAP-299	Improvement	Major	Josh Hum	0
CAP-300	Sub-task	Major	Ryan Nasr	5
CAP-301	Sub-task	Major	Ryan Nasr	8
CAP-302	Sub-task	Major	Christian Daher	4
CAP-303	Sub-task	Major	Ryan Nasr	12
CAP-304	Sub-task	Major	Ryan Nasr	6
CAP-305	Sub-task	Major	Patrick Modafferi	8
CAP-306	Sub-task	Major	Patrick Modafferi	8
CAP-307	Sub-task	Major	Patrick Modafferi	4
CAP-308	Bug	Major	Ryan Nasr	0
CAP-350	Sub-task	Major	Josh Hum	24
CAP-235	Bug	Critical	Christian Daher	2
CAP-236	Bug	Critical	Christian Daher	2
CAP-237	Bug	Critical	Christian Daher	2
CAP-238	Bug	Critical	Christian Daher	2
CAP-239	Bug	Critical	Christian Daher	2
CAP-240	Bug	Critical	Christian Daher	2
CAP-241	Bug	Critical	Christian Daher	2
CAP-242	Bug	Critical	Christian Daher	2
CAP-243	Bug	Critical	Christian Daher	2
CAP-246	Bug	Critical	Christian Daher	1
CAP-277	Bug	Critical	Christian Daher	3
CAP-319	Bug	Blocker	Unassigned	2
				TOTAL 324.42

17.2 Person-Hour Estimation

The Expert Judgement Method was used to calculate person-hour estimations.

Table 17-2 Person-Hour Estimation

Description	Worst Case	Most Likely Case	Best Case	Expected Case
Activities	421.75	324.42	227.09	324.42
Total(ph)	421.75	324.42	227.09	324.42
Velocity(ph/day)	30.12	23.17	16.22	23.17
Velocity (ph/team member/day)	3.35	2.57	1.80	2.57

17.3 Activity-on-Node Planning

The following graph reflects the progress of the planned user stories. The numbers represent the Early Start, Duration, Early Finish, Late Start, Slack, and Late Finish in days. View Menu was reopened to do search functionality. Toppings and Sides should have been completed earlier but it was missed. Both Customer and Restaurant Bill Management will have to be slightly edited after Toppings and Sides are complete.

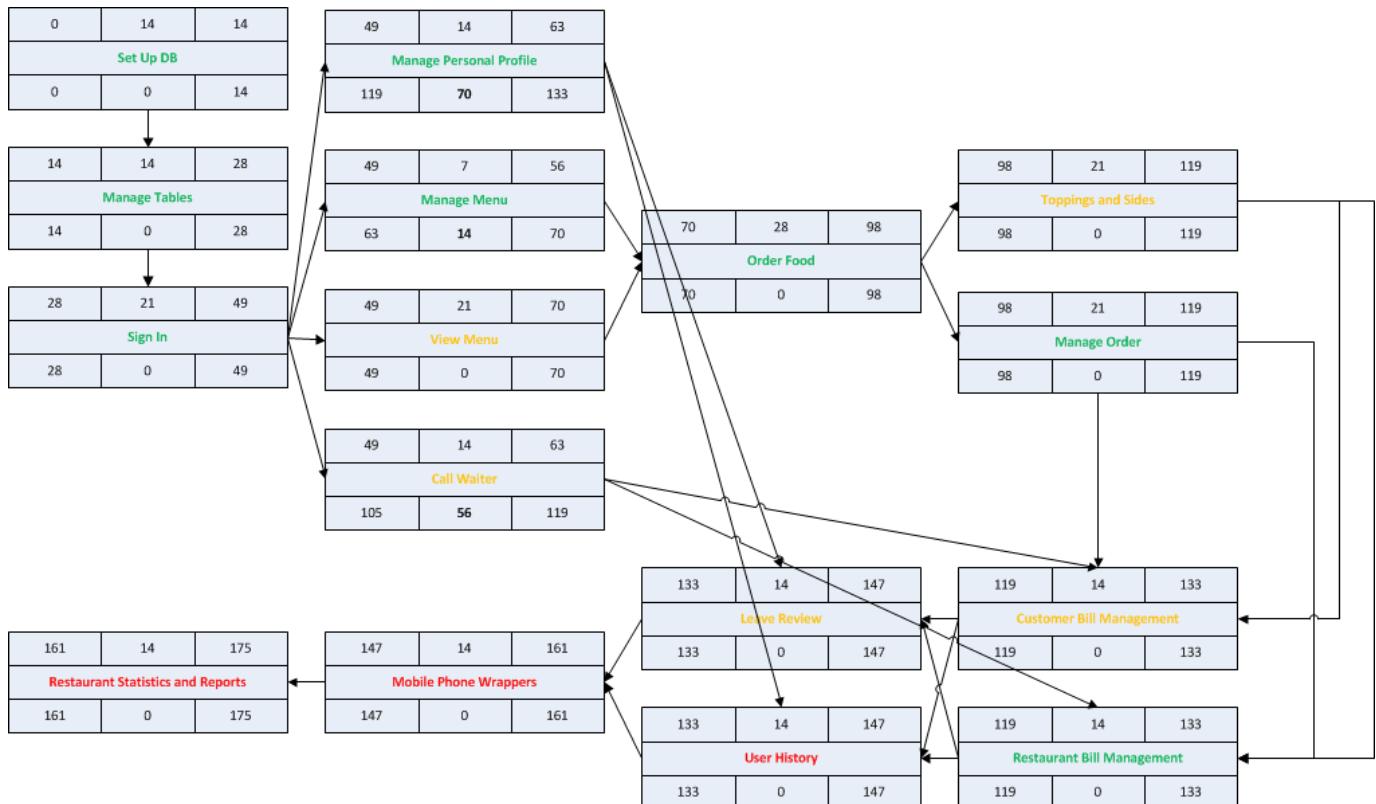


Figure 17-1 Activity-on-Node Diagram

18 Iteration 7 Report

18.1 Person-Hour Work Log

Start Date: 30/Jan/13 End Date: 12/Feb/13 [Change] (UNREGISTERED)		Total	Christian Daher	Cristian Asenjo	Cynthia Donato	Josh Hum	Katrina Anderson	Matthew Tam	Mikhail Levkovsky	Patrick Modafferi	Ryan Nasr	Total
Issue		Total	11.767h	7.633h	26.883h	39.2h	28.367h	6.7h	20.067h	17.167h	13.033h	170.817h
CAP-11	Documentation - SAD	⬆ 13.867h	3h	0.75h	4.333h	0.5h	0.017h	2h	1.267h	1h	1h	13.867h
CAP-14	Correspondence	⬆ 13.1h	0h	0.25h	0.667h	3.6h	4.833h	0.75h	0.083h	1.917h	1h	13.1h
CAP-18	Meetings	⬆ 26.05h	1.5h	1.7h	4.033h	4h	3.2h	2.95h	4.25h	2.917h	1.5h	26.05h
CAP-23	Setup	⬆ 9.667h	0h	0h	0.25h	2.833h	0.167h	0h	0h	4.417h	2h	9.667h
CAP-26	View Menu	⬆ 13.25h	0h	0h	0h	13.25h	0h	0h	0h	0h	0h	13.25h
CAP-34	Call Waiter	⬇ 8.25h	0h	0h	0h	0h	8.25h	0h	0h	0h	0h	8.25h
CAP-39	Leave Review	⬆ 11.883h	0h	0h	0h	0h	0h	0h	5.55h	6.333h	0h	11.883h
CAP-41	Customer Bill Management	⬇ 14.05h	7.017h	0h	0h	0h	0h	0h	0h	0h	7.033h	14.05h
CAP-44	Documentation - Test Plan	⬇ 0.167h	0h	0h	0h	0h	0h	0h	0.167h	0h	0h	0.167h
CAP-57	Documentation - Management	⬆ 12.083h	0h	0h	0h	11.917h	0h	0h	0.167h	0h	0h	12.083h
CAP-112	Toppings and Sides	⬆ 1h	0h	0h	0h	0h	1h	0h	0h	0h	0h	1h
CAP-128	The View Model Model (VMM) is not reflected in the SAD	⬇ 0.583h	0h	0h	0h	0h	0h	0h	0.583h	0h	0h	0.583h
CAP-162	Look and Feel	⬆ 0.017h	0h	0h	0h	0.017h	0h	0h	0h	0h	0h	0.017h
CAP-164	Documentation - Test Report	⬆ 5.75h	0h	0h	0h	0h	5.75h	0h	0h	0h	0h	5.75h
CAP-169	Optimistic offline locks logic missing from orders	⬆ 0.667h	0h	0h	0.667h	0h	0h	0h	0h	0h	0h	0.667h
CAP-164	Documentation - Test Report	⬆ 5.75h	0h	0h	0h	0h	5.75h	0h	0h	0h	0h	5.75h
CAP-169	Optimistic offline locks logic missing from orders	⬆ 0.667h	0h	0h	0.667h	0h	0h	0h	0h	0h	0h	0.667h
CAP-255	is_active and is_deleted field add extra complexity, factor that out	⬇ 9.417h	0h	0h	8.667h	0h	0.75h	0h	0h	0h	0h	9.417h
CAP-264	User Interface Testing	⬆ 3.5h	0h	0h	0h	0h	3.5h	0h	0h	0h	0h	3.5h
CAP-299	Flow and Improvements	⬆ 0.833h	0h	0h	0h	0.833h	0h	0h	0h	0h	0h	0.833h
CAP-313	The order should not have a 'VIEW_STATUS' status when a user wants to view the order.	⬆ 1.017h	0h	0h	0h	0h	0h	1.017h	0h	0h	0h	1.017h
CAP-314	Code Review	⬆ 3h	0.25h	0h	0h	1.25h	0h	0h	1h	0h	0.5h	3h
CAP-316	Canceling an order crashes the application	⬆ 0.517h	0h	0h	0h	0h	0h	0.517h	0h	0h	0h	0.517h
CAP-317	The roles for 'Order/Finalize' are invalid.	⬆ 0.017h	0h	0h	0h	0h	0h	0.017h	0h	0h	0h	0.017h
CAP-321	When logged in as a user (resto or customer), you can access the edit page for any other user	⬆ 0.383h	0h	0.383h	0h	0h	0h	0h	0h	0h	0h	0.383h
CAP-322	Viewing menu details as a customer will not work when site-wide authentication is in place	⬆ 0.017h	0h	0.017h	0h	0h	0h	0h	0h	0h	0h	0.017h
CAP-323	Menu category details shows editing links when viewing as a normal customer	⬇ 0.017h	0h	0.017h	0h	0h	0h	0h	0h	0h	0h	0.017h
CAP-324	Removing an order item from an order removes it right away and doesn't wait for the finalize button to be tapped	⬇ 0.017h	0h	0.017h	0h	0h	0h	0h	0h	0h	0h	0.017h
CAP-342	iPhone App	⬇ 4.5h	0h	4.5h	0h	0h	0h	0h	0h	0h	0h	4.5h
CAP-347	create string table for localization	⬇ 3.517h	0h	0h	0h	0h	0h	3.517h	0h	0h	0h	3.517h
CAP-348	have finders be overloaded for is deleted and is active to be passed as variables	⬆ 0.517h	0h	0h	0h	0h	0h	0.517h	0h	0h	0h	0.517h
CAP-377	Review Documents for Deliverable 2 Submission	⬆ 11.267h	0h	0h	8.267h	1h	0h	0h	2h	0h	0h	11.267h
CAP-388	User Interface Test Cases Written Incorrectly	⬆ 1.9h	0h	0h	0h	0h	1.9h	0h	0h	0h	0h	1.9h

Figure 18-1 Person-Hour Work Log

The above table shows the person-hours spent on various tasks in Sprint 7. The overall number of hours worked was lower than the past few iterations. See the iteration 7 retrospective for further explanation.

18.2 Hour Burndown Chart

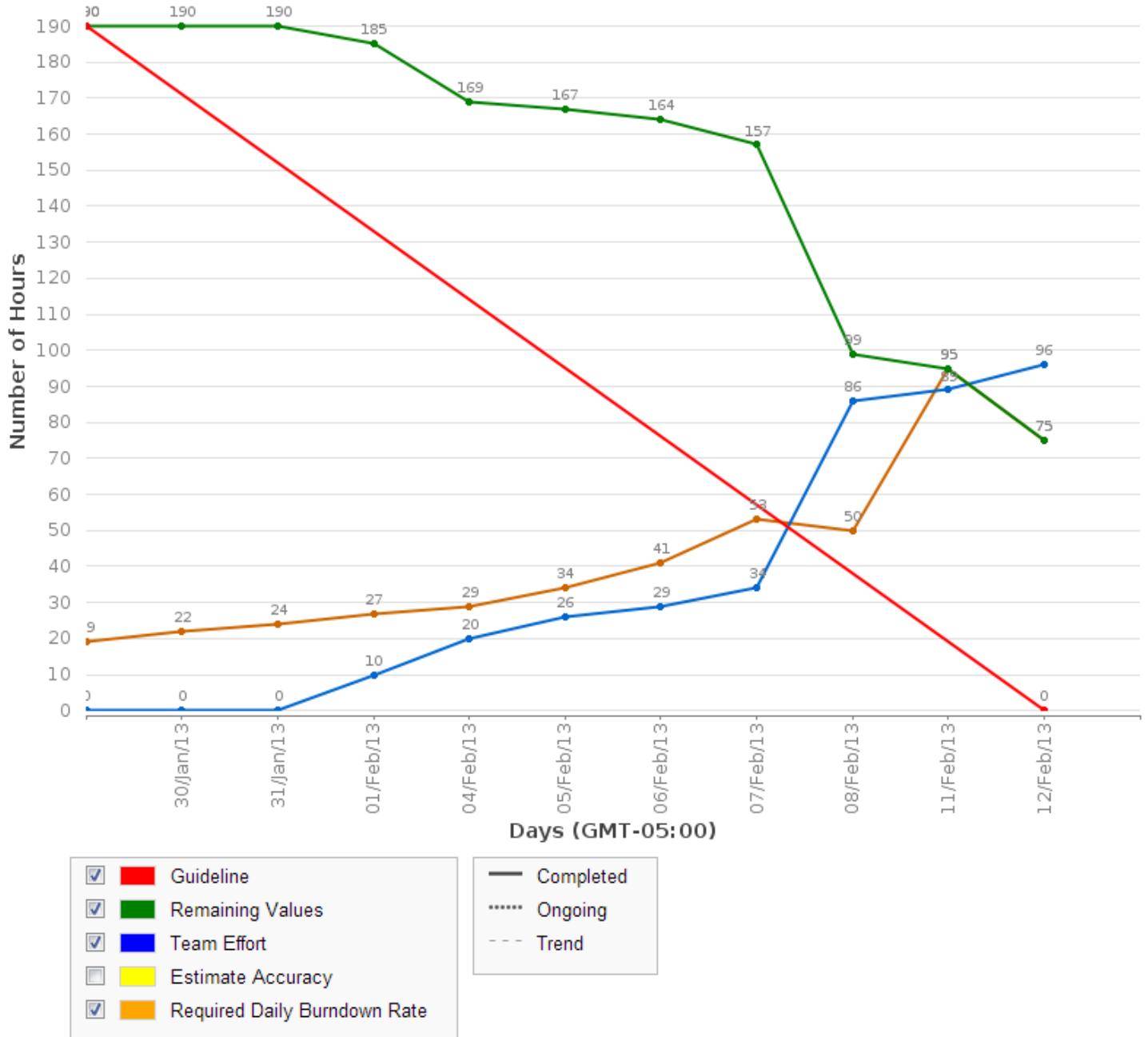


Figure 18-2 Hour Burndown Chart

Due to various reasons, there were 75 estimated hours left that were not burned down. This is not critical as essential parts of the system still got done. The chart above shows that the first week was slow in terms of effort but that more hours were worked at the end of the sprint. This was due to a SVN trunk merge that did not go well and other responsibilities preventing the team from working as much as usual.

18.3 Issue Burndown Chart

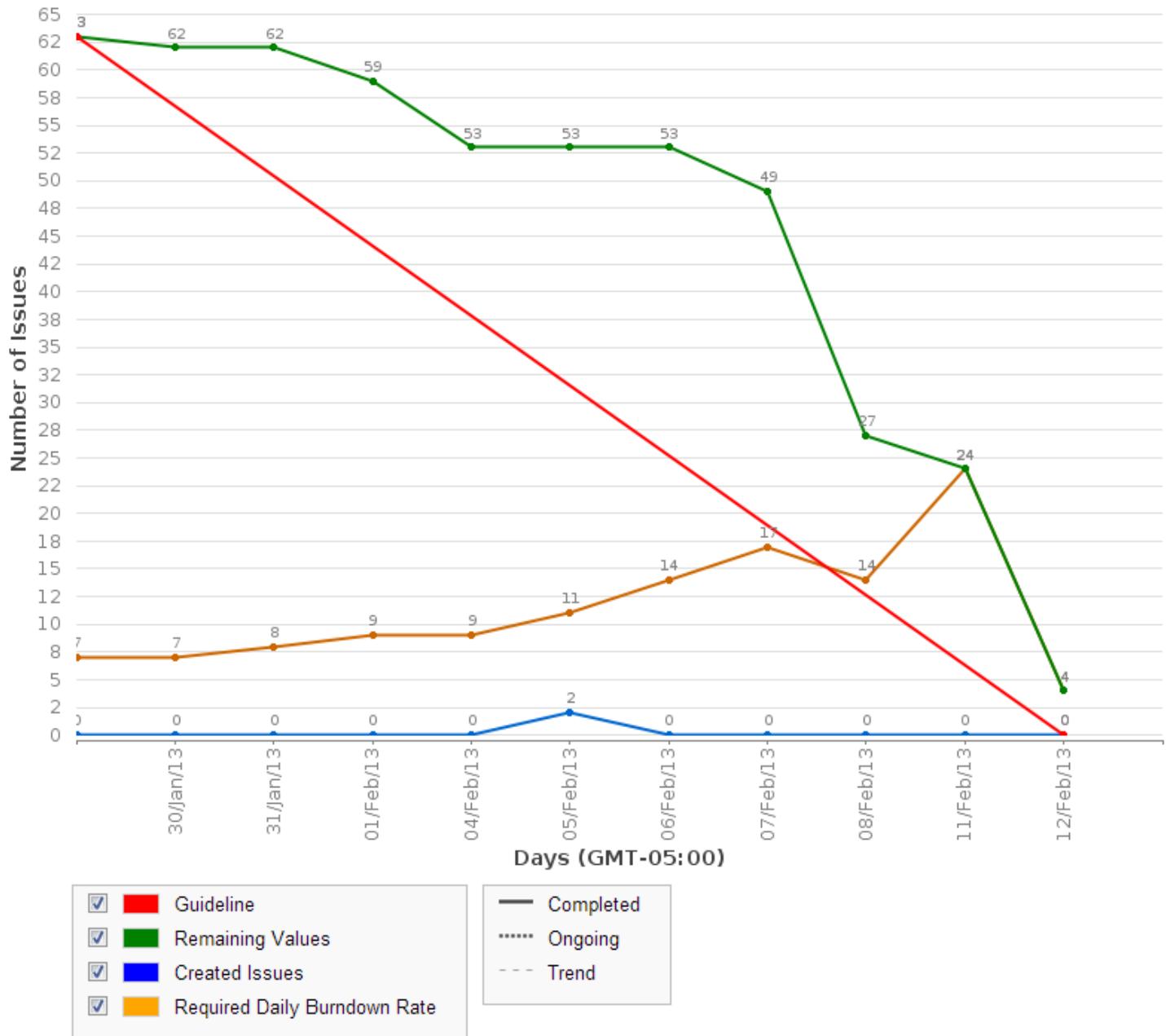


Figure 18-3 Issue Burndown Chart

Most issues were completed. However, two stories were not completed. These two stories are CAP-39 Leave Review and CAP-112 Toppings and Sides. They are scheduled to be pushed and completed in sprint 8.

18.4 Cumulative Flow Diagram

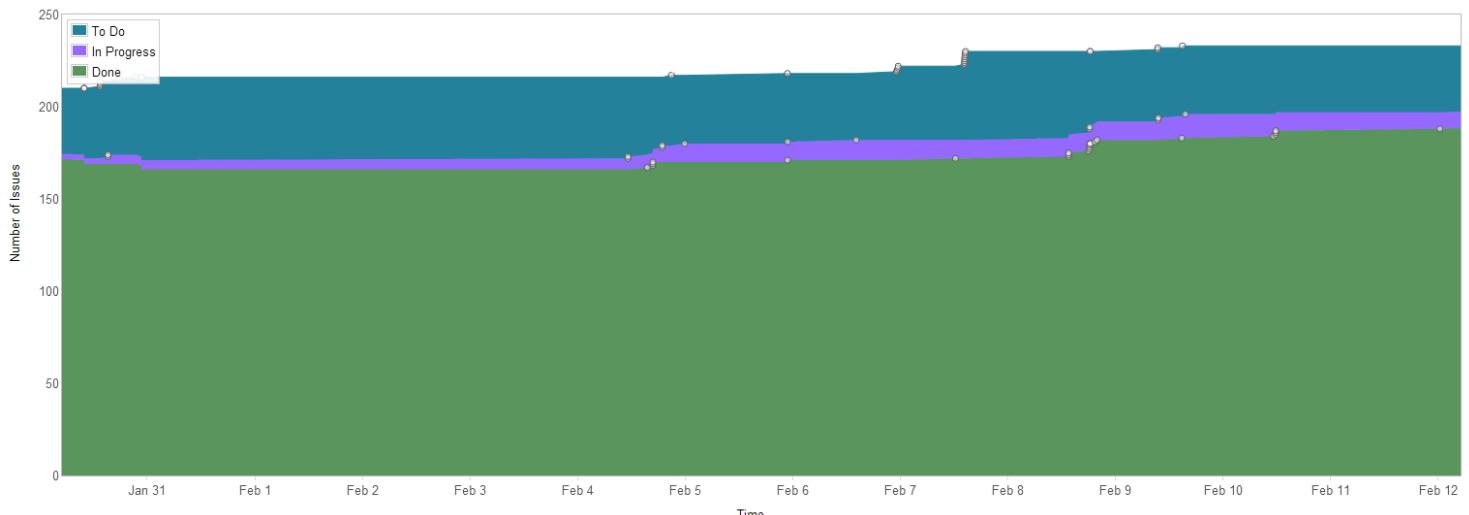


Figure 18-4 Cumulative Flow Diagram

The figure above shows that there has been a lot of progress since the beginning of the project. There is still work to do but many of the larger issues are getting cleared out quickly.

18.5 Measurement Report

18.5.1 Code Quality Analysis

The following report was generated after analyzing the code. Five methods did not meet code quality goals by exceeding the expected number of lines of code. All five were found in previous iterations. They have been entered as bugs and will be fixed in a future iteration. Since it is a minor bug, the priority is not high for it which is why they occur repeatedly in the code quality analysis report.

Since our only issues have been with lines of code, we took note for the future that the following scale defines lines of code quality:

- 0: red
- 1 – 10: green
- 11 – 10: yellow
- 21+: red

Analysis tool used: Code Metrics Viewer

Found at: <http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>

Version: 1.5.3

Last updated: 2/5/2012

Scope	Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
ModuleScope	TouchForFood.dll	80		192	4	
NamespaceScope	TouchForFood	82	4	10	2	9
TypeScope	MvcApplication	82	4	10	2	9
MemberScope	Application_Start() : void	73	1	4		5
MemberScope	MvcApplication()	100	1	1		1
MemberScope	RegisterGlobalFilters(GlobalFilterCollection) : void	94	1	2		1
MemberScope	RegisterRoutes(RouteCollection) : void	82	1	3		2
NamespaceScope	TouchForFood.Attributes	84	2	5	3	4
TypeScope	AjaxAttribute	84	2	5	3	4
MemberScope	AjaxAttribute(bool)	87	1	1		2
MemberScope	IsValidForRequest(ControllerContext, MethodInfo) : bool	83	1	4		2
NamespaceScope	TouchForFood.Controllers	66	516	127	3	
TypeScope	BillController	68	29	40	3	80
TypeScope	CategoryController	71	23	31	3	57
TypeScope	FriendshipController	69	20	29	3	42
TypeScope	HomeController	74	5	15	3	6
TypeScope	ItemController	70	27	36	3	72
TypeScope	Menu_CategoryController	67	41	31	3	85
TypeScope	Menu_ItemController	69	31	25	3	58
TypeScope	MenuController	67	39	39	3	82
TypeScope	Order_ItemController	66	28	40	3	66
TypeScope	OrderController	59	85	62	3	207
TypeScope	RestaurantController	71	21	25	3	40
TypeScope	ReviewController	59	15	35	3	40
TypeScope	SearchController	55	12	25	3	27
TypeScope	ServiceRequestController	64	73	45	3	111
TypeScope	TableController	66	37	50	3	71
TypeScope	UserController	64	30	43	3	96
MemberScope	Create() : ActionResult	74	2	9		3
MemberScope	Create(user) : ActionResult	51	5	20		21
MemberScope	Delete(int) : ActionResult	84	1	4		2

MemberScope	DeleteConfirmed(int) : ActionResult	68	3	14		5
MemberScope	Details(int) : ViewResult	84	1	4		2
MemberScope	Dispose(bool) : void	87	1	3		2
MemberScope	Edit(int) : ActionResult	84	1	4		2
MemberScope	Edit(user, HttpPostedFileBase) : ActionResult	54	8	18		14
MemberScope	Index() : ViewResult	74	2	11		3
MemberScope	LogOff() : ActionResult	76	1	5		4
MemberScope	LogOn() : ViewResult	87	1	2		2
MemberScope	LogOn(string, string) : ActionResult	46	3	24		32
MemberScope	UserController()	75	1	4		4
NamespaceScope	TouchForFood.Exceptions	97	4	1	2	4
TypeScope	AssociationExistsException	97	2	1	2	2
TypeScope	ItemActiveException	97	2	1	2	2
MemberScope	ItemActiveException()	100	1	1		1
MemberScope	ItemActiveException(string)	98	1	1		1
NamespaceScope	TouchForFood.Mappers	75	174	58	2	459
TypeScope	CategoryIM	84	6	6	2	10
TypeScope	CategoryOM	70	7	13	2	22
TypeScope	ItemIM	84	6	6	2	10
TypeScope	ItemOM	77	4	8	2	10
TypeScope	MenuCategoryIM	73	6	14	2	18
TypeScope	MenuCategoryOM	65	8	15	2	30
TypeScope	MenuItemIM	66	8	17	2	35
TypeScope	MenuItemOM	73	6	14	2	18
TypeScope	MenuIM	70	8	13	2	22
TypeScope	MenuOM	65	8	15	2	30
TypeScope	OrderIM	84	6	6	2	10
TypeScope	OrderOM	65	11	14	2	31
TypeScope	RestaurantIM	77	6	14	2	15
TypeScope	RestaurantOM	71	18	24	2	40
TypeScope	ReviewIM	84	6	6	2	10
TypeScope	ReviewOM	71	7	14	2	19
TypeScope	ServiceRequestIM	73	8	15	2	23
TypeScope	ServiceRequestOM	73	5	9	2	18
TypeScope	TableIM	84	6	6	2	10
TypeScope	TableOM	75	9	12	2	23
TypeScope	UserIM	84	6	6	2	10
TypeScope	UserOM	72	8	15	2	23
TypeScope	WaiterIM	84	6	6	2	10
TypeScope	WaiterOM	76	5	10	2	12
MemberScope	clearOrder(ICollection<order>) : void	75	2	8		3
MemberScope	delete(int) : int	67	1	5		7
MemberScope	WaiterOM()	100	1	1		1
MemberScope	WaiterOM(touch_for_foodEntities)	98	1	2		1
NamespaceScope	TouchForFood.Mappers.Abstract	90	5	1	1	8
TypeScope	GenericIM	86	2	1	1	4
TypeScope	GenericOM	94	3	1	1	4
MemberScope	delete(int) : int	100	1	0		0
MemberScope	GenericOM()	86	1	1		2
MemberScope	GenericOM(touch_for_foodEntities)	87	1	1		2
NamespaceScope	TouchForFood.Models	92	453	46	2	544
TypeScope	bill	92	21	7	1	22
TypeScope	category	92	11	4	1	13
TypeScope	CategoryFilterVM	83	9	6	1	16
TypeScope	friendship	93	11	2	1	11
TypeScope	item	92	19	5	1	20
TypeScope	ItemFilterVM	80	18	16	1	34
TypeScope	menu	92	17	5	1	18
TypeScope	menu_category	92	19	5	1	20
TypeScope	menu_item	92	21	6	1	22
TypeScope	MenuMetadata	94	7	4	1	7
TypeScope	order	89	33	13	1	55
TypeScope	order_item	92	26	9	1	29
TypeScope	OrderItemMetadata	100	1	0	1	1

TypeScope	OrderMetadata	100	1	0	1	1
TypeScope	OrderStatusHelper	65	25	7	1	44
TypeScope	OrderStatusHelper.OrderItemStatusEnum	100	0	0	1	0
TypeScope	OrderStatusHelper.OrderStatusEnum	100	0	0	1	0
TypeScope	restaurant	91	27	10	1	32
TypeScope	restaurant_user	93	11	3	1	11
TypeScope	RestaurantMetadata	94	9	1	1	9
TypeScope	review	92	25	7	1	25
TypeScope	ReviewMetadata	93	11	5	1	11
TypeScope	service_request	93	15	3	1	15
TypeScope	table	91	16	9	1	22
TypeScope	TableMetadata	100	1	0	1	1
TypeScope	touch_for_foodEntities	92	34	20	2	34
TypeScope	user	91	35	9	1	40
TypeScope	UserMetadata	93	15	5	1	15
TypeScope	waiter	93	15	5	1	16
MemberScope	first_name.get() : string	98	1	0		1
MemberScope	first_name.set(string) : void	95	1	0		1
MemberScope	id.get() : int	98	1	0		1
MemberScope	id.set(int) : void	95	1	0		1
MemberScope	last_name.get() : string	98	1	0		1
MemberScope	last_name.set(string) : void	95	1	0		1
MemberScope	orders.get() : ICollection<order>	98	1	2		1
MemberScope	orders.set(ICollection<order>) : void	95	1	2		1
MemberScope	restaurant.get() : restaurant	98	1	1		1
MemberScope	restaurant.set(restaurant) : void	95	1	1		1
MemberScope	resto_id.get() : int?	98	1	1		1
MemberScope	resto_id.set(int?) : void	95	1	1		1
MemberScope	version.get() : int	98	1	0		1
MemberScope	version.set(int) : void	95	1	0		1
MemberScope	waiter()	87	1	2		2
NamespaceScope	TouchForFood.Util.Bill	74	13	9	1	15
TypeScope	BillUtil	74	13	9	1	15
MemberScope	GetTotalAfterTax(bill) : decimal	78	3	3		2
MemberScope	GetTotalBeforeTax(bill) : decimal	85	1	3		2
MemberScope	GetTPS() : decimal	84	1	1		2
MemberScope	GetTVQ() : decimal	84	1	1		2
MemberScope	Update(ref bill) : void	62	7	9		7
NamespaceScope	TouchForFood.Util.Category	70	8	12	1	15
TypeScope	CategoryUtil	70	8	12	1	15
MemberScope	CategoryUtil()	100	1	0		1
MemberScope	CategoryUtil()	94	1	1		1
MemberScope	filterListByMenu(menu) : IList<category>	58	6	12		13
NamespaceScope	TouchForFood.Util.Html	69	16	26	1	39
TypeScope	HtmlDropDownExtensions	68	10	17	1	25
TypeScope	ImageActionLinkHelper	64	1	7	1	8
TypeScope	UrlUtils	75	5	4	1	6
MemberScope	ConvertRelativeUrlToAbsoluteUrl(string) : string	67	4	4		5
MemberScope	UrlUtils()	100	1	0		1
NamespaceScope	TouchForFood.Util.Item	63	9	15	1	17
TypeScope	ItemUtil	63	9	15	1	17
MemberScope	filterListByItem(menu_category) : IList<item>	54	8	15		16
MemberScope	ItemUtil()	100	1	0		1
NamespaceScope	TouchForFood.Util.Order	82	22	20	1	49
TypeScope	OrderStatusUtil	95	2	1	1	2
TypeScope	OrderUtil	68	20	20	1	47
MemberScope	filterItem(menu_item) : item	77	1	4		3
MemberScope	filterMenuItem(order_item) : menu_item	74	1	5		4
MemberScope	filterTable(order) : table	77	1	5		3
MemberScope	filterUser(order) : user	77	1	5		3
MemberScope	filterWaiter(order) : waiter	77	1	5		3
MemberScope	mergeExistingOrderToDb(order) : void	48	9	13		23
MemberScope	OrderUtil()	100	1	0		1
MemberScope	OrderUtil()	94	1	1		1

MemberScope	UpdatePrice(ref order) : void	65	4	11	6
NamespaceScope	TouchForFood.Util.Search	63	31	34	114
TypeScope	SearchService	51	3	18	45
TypeScope	SearchUtil	60	23	19	63
TypeScope	SearchViewModelHelper	78	5	9	6
MemberScope	PopulateSearchViewModel(menu_item) : SearchViewModel	76	1	6	2
MemberScope	PopulateSearchViewModelList(IList<menu_item>) : IList<Search	76	3	5	3
MemberScope	SearchViewModelHelper()	100	1	0	1
NamespaceScope	TouchForFood.Util.Security	77	21	22	78
TypeScope	AES	67	14	11	62
TypeScope	CustomAuthorizationAttribute	64	7	10	16
TypeScope	SiteRoles	100	0	1	0
NamespaceScope	TouchForFood.Util.ServiceRequest	100	1	0	1
TypeScope	ServiceRequestUtil	100	1	0	1
TypeScope	ServiceRequestUtil.ServiceRequestStatus	100	0	0	0
NamespaceScope	TouchForFood.Util.Session	55	5	15	30
TypeScope	SessionUtil	55	5	15	30
MemberScope	getOpenOrder(user) : order	46	4	15	29
MemberScope	SessionUtil()	100	1	0	1
NamespaceScope	TouchForFood.Util.User	75	7	23	19
TypeScope	UserUtil	75	7	23	19
MemberScope	getAuthenticatedUser(HttpContextBase) : user	64	2	14	7
MemberScope	getAuthenticatedUser(HttpContextBase) : user	64	2	8	8
MemberScope	isUserRole(SiteRoles, HttpContext) : bool	84	1	3	2
MemberScope	UserUtil()	100	1	0	1
MemberScope	UserUtil()	94	1	1	1
NamespaceScope	TouchForFood.ViewModels	81	40	14	59
TypeScope	OrderItemVM	76	1	2	4
TypeScope	OrderVM	76	23	11	32
TypeScope	SearchViewModel	91	16	1	23
MemberScope	catName.get() : string	98	1	0	1
MemberScope	catName.set(string) : void	95	1	0	1
MemberScope	description.get() : string	98	1	0	1
MemberScope	description.set(string) : void	95	1	0	1
MemberScope	menuitemId.get() : int	98	1	0	1
MemberScope	menuitemId.set(int) : void	95	1	0	1
MemberScope	menuName.get() : string	98	1	0	1
MemberScope	menuName.set(string) : void	95	1	0	1
MemberScope	metadata.get() : string	98	1	0	1
MemberScope	metadata.set(string) : void	95	1	0	1
MemberScope	name.get() : string	98	1	0	1
MemberScope	name.set(string) : void	95	1	0	1
MemberScope	price.get() : decimal	98	1	1	1
MemberScope	price.set(decimal) : void	95	1	1	1
MemberScope	SearchViewModel()	100	1	0	1
MemberScope	SearchViewModel(int, string, string, string, string, string, decimal)	68	1	1	8

Figure 18-5 Code Quality Report

18.6 Retrospective

In iteration 7, we completed 11 out of a planned 22 story points. The two stories CAP-39 – Leave Review and CAP-112 – Topping and Sides were added to the next sprint as they were not yet complete.

There was less time spent working this iteration than in the previous two. This can be attributed to a few things. There was a problem with the SVN when we merged a branch with the trunk. It took a few days to solve and code could not be worked on during that time. Also, other classes took time away from the TFF project. Quizzes and assignments happened to fall on the last day of this sprint which made it difficult to get a lot of work done. One team member was also out of the country for most of the iteration.

The team also had two meetings during iteration 7, one with Dr. Grogono and one with Dr. Constantinides. The meeting with Dr. Grogono was to introduce him to the project and get his feedback as he is our new stakeholder since Dr. Ormandjieva left. Everything went well and he seemed satisfied with the progress thus far. There were no changes to the requirements.

All things considered, the team achieved a satisfactory amount of work this iteration. We did not complete everything planned, but we are looking to complete all the rest of the user stories by the end of iteration 8. This is possible because the team has a one week break from school during iteration 8. Thus, we will make up the time lost in iteration 7 and more during iteration 8.

18.6.1 Velocity

Sprint 7 velocity (story points): 11 story points

Cumulative velocity (story points): 56 story points

Average velocity (story points): 8 story points

Sprint 7 velocity (p-h): 170.82 person-hours

Cumulative velocity (p-h): 1,228.22 person-hours

Average velocity per sprint: 175.46 person-hours

Our average velocity in story points increased this iteration. Our average velocity in person-hours dropped by one hour. Overall, we're expecting our velocity to increase next sprint as we have a one week break from school. This should give us much more time to commit to the project and complete it on time.

18.6.2 Budget

Total person-hours budgeted to date: 3,726 person-hours

Total person-hours worked to date: 1,228.22 person-hours

Person-hours budgeted per sprint: 414 person-hours

Person-hours worked in Sprint 7: 170.82 person-hours

We did not meet our goal of working at least 200 person-hours per sprint. However, we were missing one person because they were out of the country. Other school and work responsibilities also led to us not meeting our budget. In iteration 8, the team is planning to make up for that lost time.

18.6.3 What was learned

The following positive and negative points were gathered from this sprint.

Positives:

- Completed 11 story points
- Roles such as QA, Bug masher, Flow fixer, and code reviewers continued to improve the quality of the project
- Still on schedule to finish the project in time
- Communication and meetings are going very well
- Met with our stakeholder and project supervisor

Negatives:

- Did not meet our goal in person-hours worked
- Did not finish the planned stories

Appendix A References

- [1] Microsoft. (2012, December) Code Metrics Viewer. [Online]. <http://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>
- [2] Atlassian. (2013, January) JIRA; Issue & Project Tracking Software. [Online]. <http://www.atlassian.com/software/jira/overview>

Appendix B Glossary

Term	Definition
Bug Basher	A developer role. At the beginning of each sprint, all defects in the backlog are assigned to the bug basher. It is their responsibility to fix as many defects as possible according to priority. Defects can relate to any aspect of the project but are usually code defects.
QA	A developer role. The QA person writes new UI tests for the sprint. They execute the tests during the last two days of the sprint. They are also in charge of running unit tests and writing the test report. Once usability tests start, they will also be in charge of executing each usability test.
Flow Fixer	A developer role. There are often issues where the flow between two parts of the system is not correct or multiple parts work in singularity do not work together. The flow fixer's job is to ensure that all parts of the TFF system work together as designed. Flow fixing is always a coding task.

For all other terms and definitions, please refer to the SRS document - Appendix B Glossary and Appendix C Acronyms.