Concordia University
Department of Computer Science
and Software Engineering

Touch For Food

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

Table of Figures

List of Tables

Touch For Food

Requirements

Version 11.36

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
2012-09-20	0.02	Contributed Purchased Components, Supportability, and Glossary	Josh Hum
2012-09-22	0.03	Contributed Sections 1.2.1 and 1.2.2	Cynthia Donato
2012-09-22	0.04	Contributed Section 1.4	Christian Daher
2012-09-22	0.05	Contributed Sections 1.1 and 1.5	Katrina Anderson
2012-09-22	0.06	Added reliability, online user documentation and help system requirements sections	Cristian Asenjo
2012-09-22	0.07	Added interfaces, licensing requirements, and legal, copyright and other notices	Ryan Nasr
2012-09-22	0.08	Added Introduction, Functionality and Usability	Matthew Tam
2012-09-22	0.09	Added Design Constraints and Performance	Patrick Modafferi
2012-09-23	0.10	Revised Section 1.3 Stakeholder Descriptions	Mikhail Levkovsky
2012-09-23	0.11	Added Categorization of Bug Rate to Reliability, updated Glossary	Josh Hum
2012-09-25	0.12	Reviewed Ryan’s part	Patrick Modafferi
2012-09-29	0.13	Made grammar/spelling corrections. Fixed formatting issues.	Josh Hum
2012-09-29	0.14	Revised Section 1.1,1. 2 & 1.5. Created works cited.	Katrina Anderson
2012-09-29	0.15	Reviewed Pat’s part	Cristian Asenjo
2012-09-29	0.16	Made spelling, grammar and formatting corrections	Josh Hum
2012-09-30	0.17	Revised Section 1.3	Cynthia Donato
2012-09-30	0.18	Contributed Use Case Model	Cynthia Donato
2012-09-30	0.19	Reviewed Cristian’s part	Matthew Tam
2012-11-04	2.20	Reviewed and made minor corrections	Josh Hum
2012-12-16	3.21	Added information to design constraints section.	Cristian Asenjo
2012-12-16	3.22	Reviewed document and reduced redundancies	Katrina Anderson
2012-12-17	3.23	Reviewed document and corrected defect in Section 1.5	Matthew Tam
2012-12-17	3.24	Changed caption for table 4-1	Cristian Asenjo
2012-12-17	3.25	Updated the glossary.	Patrick Modafferi

2012-12-17	3.26	Added 2 extra types of bug classes.	Cristian Asenjo
2012-12-17	3.27	Fixed defect, fixed formatting and reviewed entire document	Matthew Tam
2012-12-20	4.28	Reviewed document and reduced redundancies	Katrina Anderson
2013-02-06	7.29	Review Document and made necessary corrections and indicated features which have been scoped out	Cynthia Donato
2013-02-10	7.30	Reviewed document and updated	Cynthia Donato
2013-02-12	7.31	Reviewed document	Josh Hum
2013-02-27	9.32	Renamed to Requirements Document, merged in use case model and SRS. Updated use case references. Merged Revision History.	Josh Hum
2013-03-01	9.33	Update section 1.4 table to reflect all needs and features planned and delivered during the various releases	Patrick Modafferi
2013-03-06	9.34	Reviewed document	Josh Hum
2013-03-30	11.35	Reviewed and updated document	Katrina Anderson
2013-04-03	11.36	Updated use case model	Katrina Anderson

1 Vision

1.1 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**].

1.2 Positioning

1.2.1 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 dinning at a restaurant to order food, request services from the staff, ~~integrate their dining experience with social networking sites~~ and manage their bills.

1.2.2 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 ait staff to physically come over to the table in order to communicate a need. TFF will also provide statistics derived from food reviews provided by dinners.

1.3 Stakeholder Descriptions

1.3.1 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.
Wait Staff	The individuals that serve 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.

1.3.2 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. 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.

1.4 Product Overview

1.4.1 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.

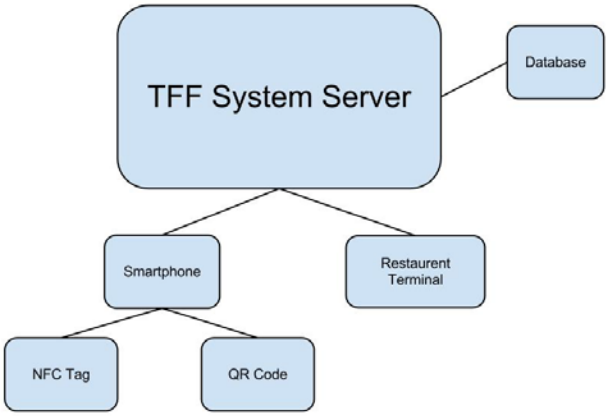

Figure 1-1 System Overview

1.4.2 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.

1.4.3 Needs and Features

Need	Priority	Features	Planned Release (Milestone#)
Order food (customer)	High	Order management, bill management	4
~~Make reservation (customer)~~	~~Medium~~	~~Food reservation, table reservation~~	~~5~~
Call waiter (customer)	Low	Assistance	7
Manage personal page (customer)	High	Review order history, ~~managing friends, social networking,~~ User Profile	2
User History	Low	User profile, Order Management	8
View menu (customer)	High	Menu management, Order management	2
Leave Review (customer)	High	Review order history, ~~Social networking,~~ rating system	7
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
Sign In	High	User Profile, Tap in*	5
Manage tables (restaurant)	Medium	~~Table reservation,~~ Tap in*	5
Manage bills (restaurant)	High	Bill management	6
Manage bills (customer)	Medium	Bill management	7
~~Restaurant page (restaurant)~~	~~High~~	~~Social networking, menu management, order management~~	~~4~~
Restaurant reporting and statistics (restaurant)	Low	Reporting system	8
Toppings and Sides	Low	Menu management, order management	7

*see Glossary

1.4.4 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

1.5 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 restaurant closes and before the restaurant 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.

2 Use Case View

2.1 Actor Goal List

Table 2-1 Actor Goal List

Actor	Goal
Restaurant Client	1. Place Order 2. View Menu 3. Service Request 4. Create User Profile 5. Login to Personal Profile 6. Update User Settings 7. Submit A Review 8. Manage Bill 9. Lookup Customer Restaurant and Order History
Restaurant	1. Manage Menu 2. Deliver Order Item 3. Service Request Response 4. Create User Profile 5. Login to Personal Profile 6. Update User Settings 7. Generate Report 8. Manage Bills 9. Manage Tables

2.2 Use Case Model

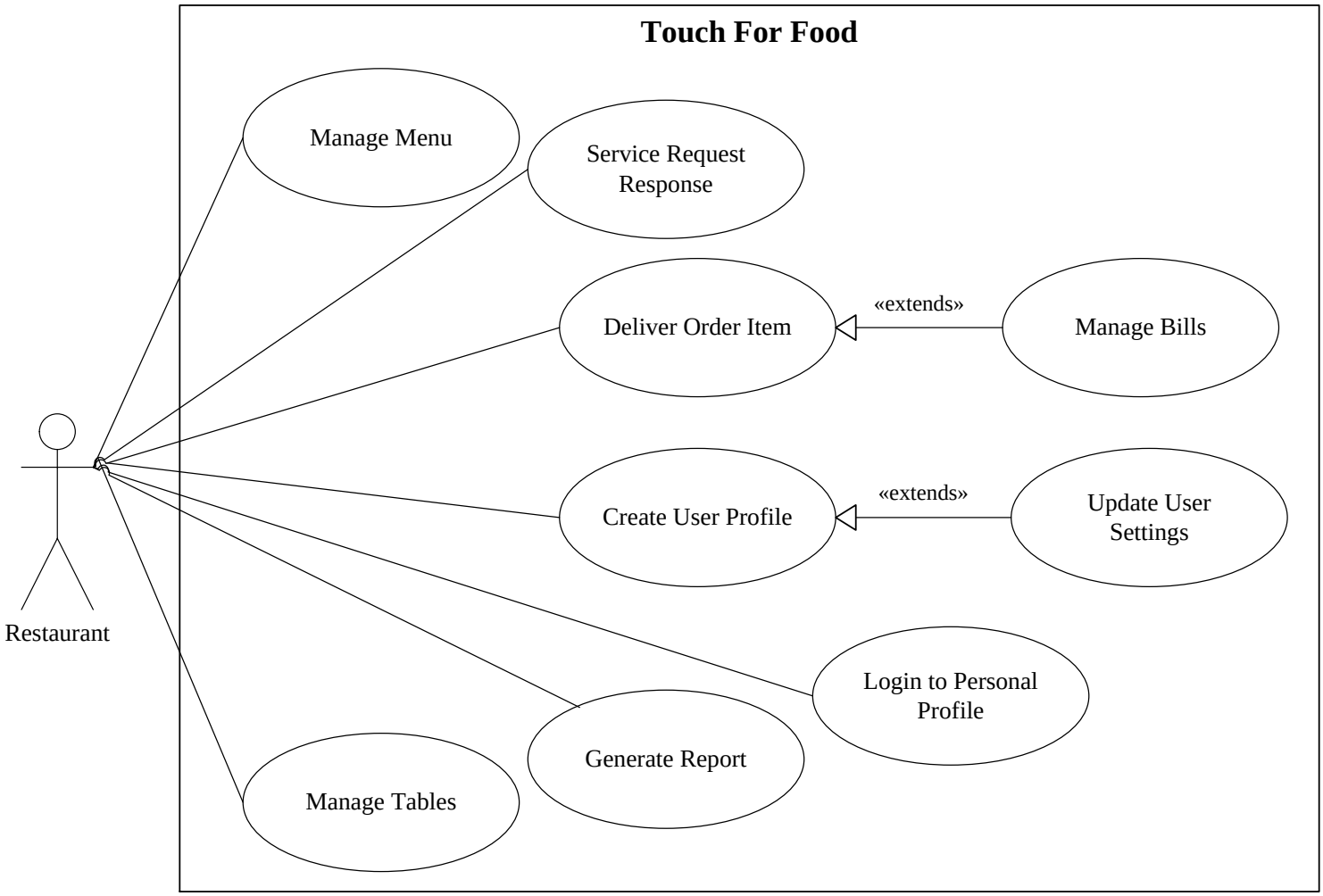

Figure 2-1 Restaurant Use Case Model

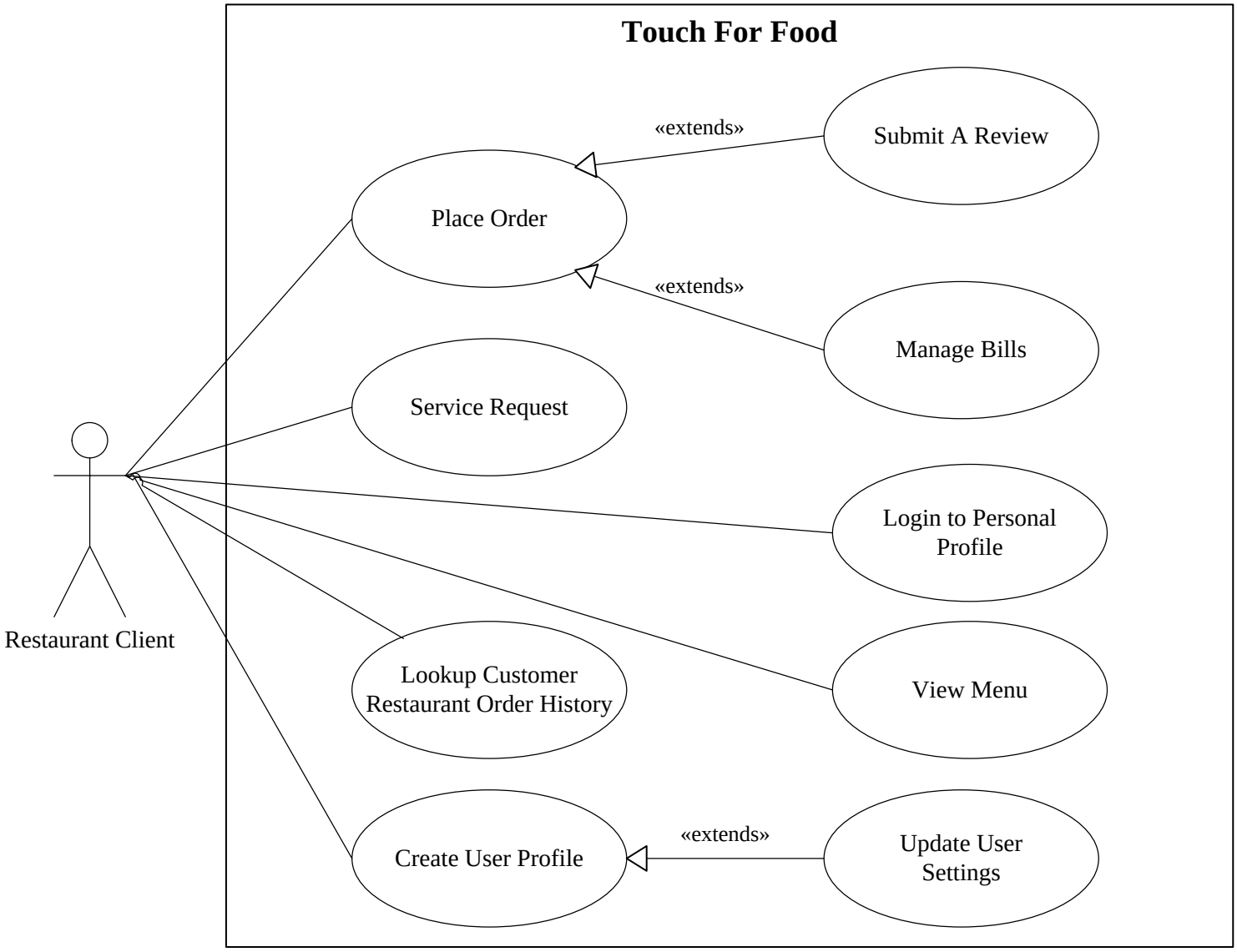

Figure 2-2 Client Use Case Model

3 Supplementary Requirements & Specifications

3.1 Introduction

3.1.1 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 Analysis and Architecture documents, for the major functional requirements.

3.1.2 Scope

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

3.1.3 Definitions, Acronyms and Abbreviations

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

3.1.4 References

Please see Appendix A, References, of this document.

3.1.5 Overview

The supplementary specifications 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 [**2**].

3.2 Functionality

3.2.1 Restaurant Client User Controls

3.2.1.1 Description

The TFF application shall allow a restaurant client to place food orders, manage how the bill will be separated, call for assistance, manage their personal settings and review meals.

3.2.1.2 Input

Restaurant client controls shall be accessed through a mobile device.

3.2.1.3 System Processing

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

3.2.1.4 Output

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

3.2.2 Restaurant Administrative Controls

3.2.2.1 Description

The TFF application shall allow a restaurant administrator to manage: the items on their menus, ~~table reservations~~,customer orders, service requests, bills and statistical reports.

3.2.2.2 Input

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

3.2.2.3 System Processing

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

3.2.2.4 Output

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

3.3 Usability

3.3.1 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.

3.3.2 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.

3.4 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.

3.4.1 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. [**3**]

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

3.5 Performance

3.5.1 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.

3.5.2 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.

3.5.3 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.

3.5.4 Customer Capacity

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

3.5.5 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.

3.5.6 Native App Hard Drive Space

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

3.5.7 ~~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.~~

3.5.8 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.

3.5.9 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.

3.6 Supportability

3.6.1 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 [**4**].

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 3.4.1 of this document for categorization of bug types.

3.7 Design Constraints

3.7.1 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**	**Yes**	**Yes**
iOS	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**No**	**Yes**	**Yes**

Table 7-2 Operating system features to be used

3.7.1.1 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.

3.7.1.2 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.

3.7.1.3 ~~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.~~

3.7.2 Web application Front End and Back End Development

3.7.2.1 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.

3.7.2.2 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.

3.7.3 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.

3.7.4 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 [**3**]. Team members will report bugs, assign tasks, log hours and track sprint progress with this tool.

3.7.5 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 [**5**]. 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.

3.7.6 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.

3.8 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 wait staff.
- 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

3.9 Purchased Components

NFC tags will need to be purchased for testing. An iOS developer licenses will also be needed to create the iOS mobile wrapper. A one-time purchase of JIRA and the JIRA Agile plugin will also be needed to manage the software development process.

3.10 Interfaces

3.10.1 User Interfaces

The core system application will be based on a web application that will run on Google Chrome. 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~~
- Service Requests
- 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
- Service request management
- Bill management
- ~~Restaurant page management~~
- Order viewing and management
- Reporting and statistics

3.10.2 Hardware Interfaces

3.10.2.1 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.

3.10.2.2 Input devices

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

3.10.2.3 Database and storage

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

3.10.2.4 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

3.10.2.5 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

3.10.2.6 Additional hardware

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

3.10.3 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.

3.10.4 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.

3.11 Licensing Requirements

3.11.1 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~~.

3.11.2 Domain Name

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

3.11.3 Copyright

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

3.11.4 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," Concordia University, Montreal, Canada, Vision Document, v6.16, 2012.

[2] 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.

[3] Atlassian. (2012, December) Issue & Project Tracking Software. [Online]. http://www.atlassian.com/software/jira/overview

[4] Microsoft. (2012, December) C# Coding Conventions (C# Programming Guide). [Online]. http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx

[5] 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.
Tap In	The action of signing in to TFF is more complex that one would expect. This involves the use of NFC and QR codes to initiate a session with a user and a given table.
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