Smart Waiter Software Requirements

Meraj Patel #1137491
Pavneet Jauhal #1149311
Shan Perera #1150394

February 2, 2016

Contents

List of Figures

List of Tables

Revision History

Date	Comments
October 9, 2015	first draft.
January 19, 2016	Modified function and non function requirements.

Table 1: Revision History Table

Template

This document uses Volere Template for its organization.

1 Project Drivers

1.1 The Purpose of the Project

This project aims to provide a solution that will allow users to order and pay through a mobile application at restaurants. This opportunity arose from the lack of a universal application in the market that allows; users to walk into a restaurant, scan a code to view the menu, and proceed to order and pay through the use of a singular app. Once the product is complete, Android users will be able to walk into any restaurant which offers our solution, and have the ability to use these services.

1.1.1 Project Goal

The delivered application will allow an Android user to use the application at any restaurant, which offers the Smart Waiter solution. More importantly, the client should be able to view restaurants complete menu, order selected items and pay for their meal through the application.

1.2 The Client, the Customer, and Other Stakeholders

1.2.1 The client

The prospective client of this application will be a restaurant owner or manager, who would like to implement an application based ordering system at their restaurant.

1.2.2 The Customers

The targeted customers will include clients of the restaurants, which offer Smart Waiter solution.

1.2.3 Other Stakeholders

Other Stakeholders of this project include:

- Supervisor Dr. Rong Zheng [Hyphens aren't showing up in the pdf for me, try using '–' —DS]

- Android Users Clients of restaurant

- Developers and Testers Pavneet Jaual, Meraj Patel and Shan Perera

- Beta Testers

- Prospective manager/restaurant owners . . . [Why do you have dots here? If there are more stakeholders you can include, include them. Otherwise, just end the list —DS]

1.3 Users of the Product

1.3.1 The Hands-On Users of the Product

Group - Restaurant clients
Clients who frequent restaurants, which offer Smart Waiter services.

User Role
These users will use the application to view restaurant menu, order their selection and pay for their meal.

Subject Matter Experience
These users can be categorized as novice, as they do not need extensive knowledge of the business. The user can simply be a client walking into a restaurant for the first time.

Technological Experience
The technological experience of this user can also be considered novice. The application will not require any training to use.

Other User Characteristics
There are some basic characteristics that the user should have, to qualify to use this application:

- Should have an android device and an internet connection

- Basic understanding of how to use an android application.

- Should have a credit card, to make payments through the application.

- Users who generally frequent restaurants at least once a month [Why do they need to frequent restaurants at least once a month to qualify for this app? Couldn't anyone use it regardless of how often they go out? —DS]

Group - Restaurant Owners
Restaurant managers who have Smart Waiter solutions implemented

User Role
These users will use the application to view restaurant menu, order their selection and pay for their meal.[Wouldn't they be doing something different than just what the customers can do? —DS]

Subject Matter Experience
To be considered masters, as they have in depth knowledge of the business.

Technological Experience
The technological experience of this user can be considered novice. The application will not require any training to use.

Other User Characteristics
There are some basic characteristics that the user should have, to qualify to use this application;

- Should have an android device and an internet connection

- Basic understanding of how to use an android application.

- Credit card required if they want to test payments method

- These users will pay attention to details. They will generally critique the UI and other aspects of the application to see how it fits their specific business needs.

Group Testers/Developers
Testers and developers of Smart Waiter

User Role

These users will perform end-to-to system testing. The tester and developers will verify that each use-case and each feature in the application works correctly.

Subject Matter Experience
They will have complete knowledge of the business needs and requirements. Thus they can be considered a master.

Technological Experience
They will be a master in technological experience. They will know the code base of the application thoroughly. Similarly, they would know the application infrastructure and other technology used in the application.

Other User Characteristics

- Advanced knowledge of android framework

- Advanced knowledge of the code base

- Evaluate the application from the engineering perspective

- These users will pay close attention to detail

- Knowledge of all possible use-cases and expected outcomes of the application

[Who will be setting the menu and how? —DS]

1.3.2   Priority assigned to the user

Key Users
The key users of this application will be the target market of this product. Namely, the restaurant owners/managers and the restaurant clients.

Secondary Users
The secondary users will be considered the Testers/Developers.

1.3.3   User Participation

Restaurant Clients
These users will be participating and helping us provide the requirements
and throughout the lifetime of the application. Specifically, they will be hav-
ing a report bug feature built into the application to report bugs and suggest
improvements. They will provide usability requirements.

Restaurant Owners
These users will also participate in providing requirements throughout the
lifetime of the application. However, their main concerns are expected within
1 day of application usage. These users will provide business knowledge and
business requirements for the application.

Testers/Developers
Throughout the development cycle of the application these users will contin-
uously contribute to the requirements of the application. Specifically, they
will provide interface prototyping, business knowledge and usability require-
ments for the application.

1.3.4   Maintenance Users

The testers/ Developers of the application will be considered the maintenance
users. When going through the steps of unit testing, end-to-end testing and
verification process, these users will continuously make changes to improve
the product.

2   Project Constraints

2.1   Mandated Constraints

2.1.1   Solution Constraints

Constraint 1
Description
The application shall pull the restaurants menu from the server via a barcode

scan.

Rationale
Provide a simple solution, which supports multiple restaurants.

Fit criterion
Must be approved by tester and developer. They must confirm that complete information about the restaurants menu has been pulled from the server onto the users device.

Constraint 2

Description
The barcode scan by user shall send table and other user information to the server.

Rationale
Provide the restaurant with clients information and table number to serve food too.

Fit criterion
Must be approved by tester and developer. They must confirm that complete information about the user has been pushed to the server onto the users device.

Constraint 3

Description
The application must run on Android operating system.

Rationale
We are catering to the entire android platform users.

Fit criterion
The application must be approved complaint [What does "approved complaint" mean? Did you mean "compliant"? —DS] from the developers and testers. This will involve thorough testing on the Android platform. [This paragraph is awkward —DS]

10

[You should avoid using pagebreaks —DS]

Constraint 4
Description
The application shall support credit card payments.

Rationale
This will transfer money to the restaurant account.

Fit criterion
The functionality must be approved complaint ["Approved complaint" again —DS] by the developer and testers. This will require approving the code base and the end-to-end functionality.

Constraint 5
Description
The payment information for the client must be stored safely

Rationale
This is very sensitive data and should not be abused and exploited.

Fit criterion
The functionality must be approved complaint ["Approved complaint" again —DS] by the developer and testers. This will require approving the code base and the end-to-end functionality. [The uses of "approved complaint" make the sentences awkward, and I do not know what they mean —DS]

2.1.2 Implementation Environment of the Current System

[These are design decisions that will be used, not constraints. If these are actual constraints, explain why you *must* use them. —DS] The environment will require the following;

- Source code will be written in Java

- Android Studio will be used as the IDE

- Couchbase API Library for Android will be used for cloud storage

- Stripe API will be used to process credit card payments

2.1.3   Partner or collaborative applications

Not Applicable

2.1.4   Off-the-shelf software

Sketch will be used to make UI blueprints. In addition, Photoshop will be used to make custom logos and icons. [Again, not constraints unless you explain why they *must* be used instead of existing alternatives. —DS]

2.1.5   Anticipated workplace environment

The users will be using this application in a restaurant. Typically, this work environment would be loud and distracting. Thus, we plan on making the UI of the application visually attractive, where the user does not need to read much. [This could use some clarification. —DS]

2.1.6   How long will the developers have on the project

The final deadline for the project is mid April 2016. The detailed deliverables and their respective deadlines are listed below;

- Requirements Document Revision 0: October, 9, 2015

- Proof of Concept Plan: October, 23, 2015

- Test Plan Revision 0: October, 30, 2015

- Proof of concept demonstration: November, 16, 2015   November, 27, 2015

- Design document revision 0: January, 11, 2015

- Revision 0 Demonstration: February, 1, 2015 February, 12, 2015

- Users Guide Revision 0: February, 29, 2015

- Test Plan Revision 0: March, 21, 2015

- Final Demonstration: Mid-April, 2016

- Final Documentation for the product: April, 1, 2016

[Fix the hyphens —DS]

2.2 Naming Conventions and Terminology

N/A [Seems like you could have something here. You use "PCI" in the next fact —DS]

2.3 Relevant Facts and Assumptions

2.3.1 Relevant Facts

Fact 1
Stripe API used for payment processing is payment card industry (PCI) complaint. ["Compliant" —DS]

Fact 2
Restaurant clients should not be able to dine and dash. [This sounds more like an assumption or a constraint. —DS]

2.3.2 Assumptions

Assumption 1
Parse Database offers top-notch security for backend infrastructure from physical to application level.

Assumption 2
SSL connection used for socket programming is secure and cannot be sniffed.
   [These are implementation details. You should be a bit more abstract —DS]

3 Functional Requirements

3.1 The Scope of the Work

3.1.1 The Current Situation

An android application is required to allow customers to walk into a restaurant and use their smart device to view a menu, place an order and pay for there ["Their" —DS] meal. This will be a singular solution extended to multiple restaurants so users only have to use one mobile application to serve these functions.

3.1.2 The Context of the Work
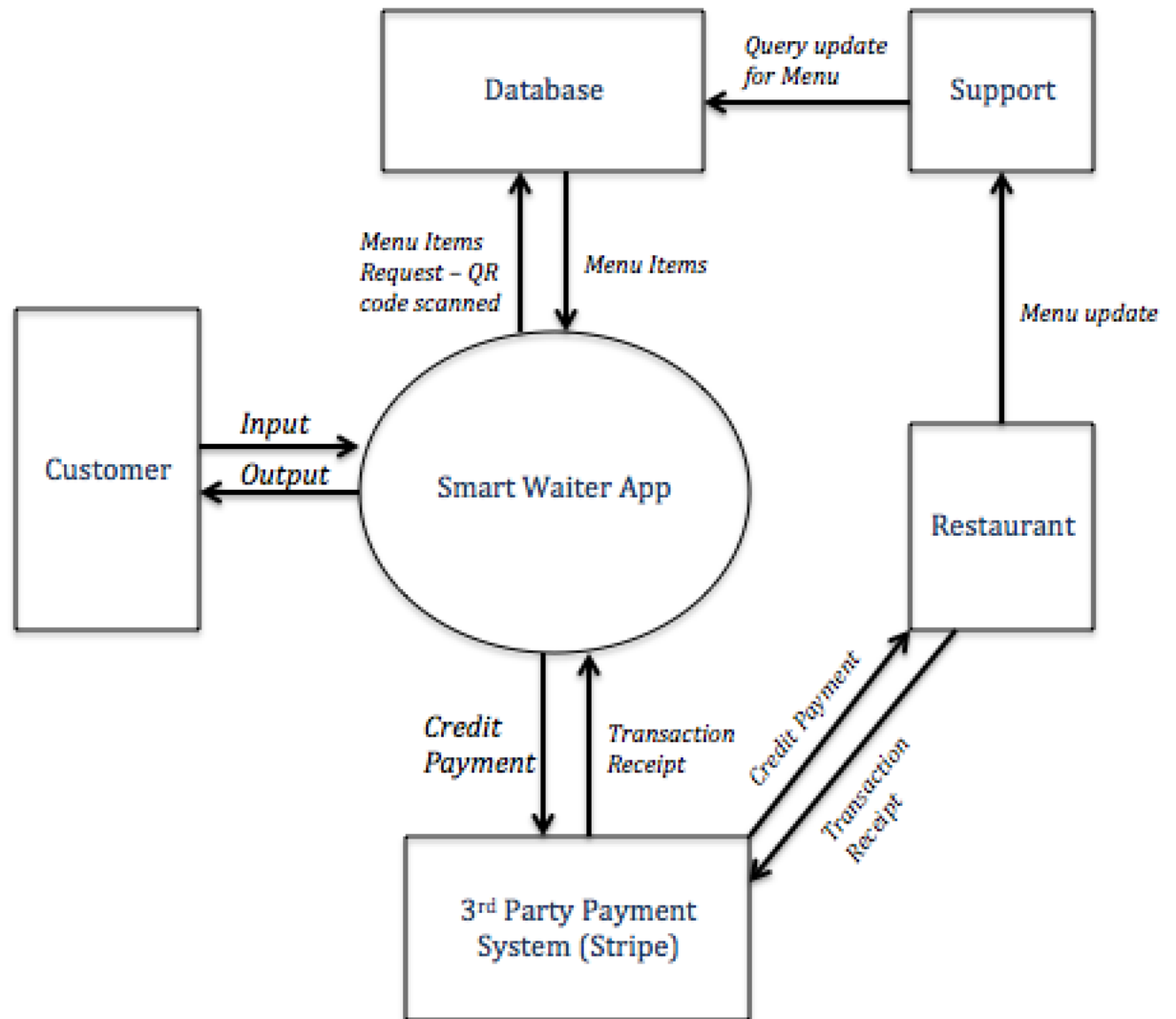
Figure 1: Context of the work figure.

3.2 Business Data Model and Data Dictionary

3.2.1 Business Data Model

3.2.2 Data Dictionary

Name	Content	Type
Restaurant	Restaurant Identifier	Class
User	User Identifier	Class
Waiter	Waiter Identifier	Class
Table	Table Identifier	Class
Order	Order Identifier	Class
Menu	Menu	Class
QR Code	QR Code	Class
User Identifier	Username + Password + Credit Card Info	Attribute/Element
Waiter Identifier	Employee Name + Employee Number	Attribute/Element
Table Identifier	Table Number + Waiter Assigned	Attribute/Element
Order Identifier	Table Number + User Identifier	Attribute/Element

Table 2: Data Dictionary Table

3.3 The Scope of the Product

3.3.1 Product Boundary

Figure 2: Product boundary figure.

3.3.2 Product Use Case Table

PUC #	PUC Name	Actor/s	Input/Output
1	Request Menu	Restaurant Customer	QR Code (In)
2	View Menu	Restaurant Customer	Menu Contents (out)
3	Order Customization	Restaurant Customer	Order Specification (in)
4	Submit Order	Restaurant Customer	Submit Order (in)
5	Process Transaction (3rd party)	Strip API	Process Transaction for order (in)
6	Process Transaction (restaurant)	Restaurant	Process and Record Transaction (in)
7	Display Receipt	Customer	Display Receipt to customer (out)
8	Rate Items	Customer	Submit Rating out of five stars (in)
9	View ratings	Customer	View ratings of menu items (out)
10	Provide Employee feedback	Customer	Provide feedback about employee (in)

Table 3: Product Use Case Table.

[Where does the table QR code come into play? Where is bug reporting? —DS]

3.4 Functional Requirements

3.4.1 Functional Requirements

Requirement 1
Description:

The product shall scan a QR code to access the restaurant's menu.

Fit Criterion:
The product should access the camera application on the respective device to take a picture of the QR code and it will bring up the restaurant menu or display an error.

Requirement 2
Description:
The product shall allow the user to register an account for complete access to the products payment services.

Fit Criterion:
The product will ask the user if they would like to register an account, if the user agrees: the product will bring them to the registration page asking the user to provide a username, password, and credit card information. If the user declines, the product will bring them back to the main menu of the application.

Requirement 3
Description:
The product shall allow the user to report bugs.

Fit Criterion:
The product offers a menu to allow the user to report a bug they encountered. The user presses a button to report the bug and a pop-up form with a text entry box allows the user to describe the bug in 400 characters or less and sends the information to the developers through an automated email.

Requirement 4
Description:
The product shall provide error codes for common errors.

Fit Criterion:
The user enters incorrect details when trying to login to their account. The product reports the error code 1, with the following message: Incorrect login information. i.e., Error Code 1: Incorrect login information.

Requirement 5

Description:

The product will display the menu in a similar fashion to the hard copy menu offered at the restaurant.

Fit Criterion:

A user accesses the menu using the product, it looks similar, if not identical to the paper menu offered at the restaurant.

Requirement 6

Description:

The product will provide detailed information on a menu item when selected.

Fit Criterion:

A user is browsing the menu and selects a hamburger, a new screen is brought up on the users device which details the ingredients of the hamburger and any extra ingredients they may add to the hamburger.

Requirement 7

Description:

The product will allow a user to request special instructions for food items they have ordered.

Fit Criterion:

A user is browsing the menu and selects a steak, the user decides to add the hamburger [is it a steak or a hamburger? —DS] to his order, a new screen is brought up on the users device with a text entry box that allows the user to give special cooking instructions to the chef. The user types medium rare into the text box and confirms the order.

Requirement 8

Description:

The product will allow the user to view their current order.

Fit Criterion:

A user is browsing the menu and adds a chicken entree [Should be "entrée" —DS] and a beer to her order. Afterwards she adds a side dish to her order. She wants to see the subtotal of her order, she views her order through the

interface.

Requirement 9
Description:
The product will update the total amount owed for the order in real time.

Fit Criterion:
A user is browsing the menu and adds an entree to his order, the total is calculated immediately afterwards. The user then adds a drink to his order, the total of the order updated immediately afterwards. [You shouldn't mention the specifics of what the user is adding. If they add an item to their current order, the total should update immediately (regardless of what the item is) —DS]

Requirement 10
Description:
The product will allow users to pay for their order using a credit card.

Fit Criterion:
A user is finishing his meal and all the waiters are busy with other tables. The user is in a rush, he pays for his order using the credit card info registered to his account associated with the product [You don't need to tell a story as part of your fit criterion, simply how to confirm that the requirement has been met —DS]

Requirement 11
Description:
The product must be Payment Card Industry (PCI) compliant. [Where's your fit criterion? —DS]

Requirement 12
Description:
The product will allow users to place their order.

Fit Criterion:
A user adds a hamburger, fries and a drink to her order. She is really hungry and doesn't want to wait for a waiter to take her order. She places the order through the application. [Not a fit criterion —DS]

22

Requirement 13

Description:

The product will show a summary of the user's order before they place it.

Fit Criterion:

A users adds a hamburger, fries and a drink to her order. She is ready to place her order, a screen is brought up on her device which shows her what she is about to order.

[No need for the story —DS]

Requirement 14

Description:

The product will allow users to remove items from their order before it has been placed.

Fit Criterion:

A user adds a hamburger, fries and a milkshake to their order. The user then decides they no longer want a milkshake and remove it from their order before he place the order.

Requirement 15

Description:

The product will allow the menu to be viewed on a smart phone.

Fit Criterion:

A user sits down at a table in a restaurant and scans the QR code using the product. The menu is brought up on the smart phone.

[You do not follow the template's requirement shell —DS]

4 Nonfunctional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

The user interface for this app will be simple minimal interface that allows the user to view restaurant menu and place an order with no complications.

4.1.2   Style Requirements

The style of the app will be consistent between most pages by using a template. As well, the style will be consistent between multiple restaurants. Only menu items will change. Doing this would make it easier for the user to get used to the interface and keep it memorable.

4.2   Usability and Humanity Requirements

4.2.1   Ease of Use Requirements

The app will be easy to use for any casual customer equipped with a smart device in a dining establishment. This can range from an elementary student to a senior citizen.

4.2.2   Personalization and Internationalization Requirements

Considering this app can be used in various ethic restaurants, customers may prefer to communicate in their own language. Thus, for a future functional requirement, the app will provide option to restaurant mangers to allow them to add an additional language setting. The app will always have English setting.

4.2.3   Learning Requirements

The learning curve will be kept as minimal as possible. The requirement is a user should be able to flawlessly use the app to view the restaurant menu and place and order without the use of a help function. The user should be comfortable and have complete understanding with this process as soon as he/she does it once. To meet this requirement, usability tests will be conducted to until high percentage of users feel comfortable with the process. [Will there be a help function just in case? How can you guarantee that anyone will be able to use it? —DS]

4.2.4   Understandability and Politeness Requirements

To promote understandability, appropriate images, text descriptions and symbols will be in place. That is, the user will understand this app is meant to serve as a menu and waiter by incorporating pictures, description and

price of food available. A checkout button will clearly be indicated when the user is ready to place the order.

4.2.5 Accessibility Requirements

This app will be accessible to those with relatively good eyesight and have use of their index finger.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

Query restaurant information should take no longer than five seconds. Processing transactions should take no longer than 10 seconds.

4.3.2 Safety-Critical Requirements

To maintain confidentiality, credit card and personal information will not be stored in the backend database. As well, an secure connection will always be instantiated when placing an order to ensure data protection.

4.3.3 Precision or Accuracy Requirements

The restaurant will set all monetary amounts for food and beverages seen on the app. All appropriate taxes will be included in the bill when making an order. [There should be functional requirements detailing how this is done —DS]

4.3.4 Reliability and Availability Requirements

Mobile app will be working whenever the user wants. However, services such as viewing menu and placing orders will only be available during restaurant hours. [There should be functional requirements to say what happens if a user tries to place an order outside of restaurant hours —DS]

4.3.5 Robustness or Fault-Tolerance Requirements

If the app fails to scan the QR code repeatedly (after 3 times), it will prompt the user to reach out to their waiter/waitress for service. It will also provide

an option to report this error. [This should be a functional requirement —DS]

4.3.6 Capacity Requirements

App capacity is dependent on how many people can be seated at a restaurant. This will vary, as it is dependent on restaurant capacity [Are you talking about the app on the phone or the restaurant backend that can handle a certain capacity? —DS]

4.3.7 Scalability or Extensibility Requirements

Increasing number of participating restaurants may induce database upgrade in terms of efficiency in querying. [Is there one central database for all restaurants? Where are the database specifications? —DS]

4.3.8 Longevity Requirements

App is reliant on the restaurant. [What does that mean? —DS] For example, if restaurant decides to shutdown, the app will no longer server that restaurant.[Grammar issues with that sentence —DS]

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

Ordering component of the app will be functional only at restaurants. This includes, viewing a menu and placing an order. [Grammar —DS].

4.4.2 Requirements for Interfacing with Adjacent Systems

Our product must be compatible with various versions of android. The smart device must have a camera to operate.

4.4.3 Productization Requirements

App will be available on Google Play for download by customers.QR codes will be implemented at participating restaurants. These codes will be placed on tables so customers can scan appropriately if they have the app.

4.4.4 Release Requirements

A new app release will not affect restaurant data or compromise integrity. For example, customers with an older app version would still be able to view the same information as those with updated version.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

Restaurants menu changes will be processed after restaurant hours and take effect the next day. This ensures all customers see the same information.

4.5.2 Supportability Requirements

If a restaurant manager requires support on maintenance (updating menu items), they will log a request with us to make this change. Customers using the app will rely on self-support (download and update when new release is available).

4.5.3 Adaptability Requirements

For a future requirement, the app will be extended to other mobile platforms (IOS, blackberry, windows)

4.6 Security Requirements

4.6.1 Access Requirements

User is only allowed access to restaurant data by scanning QR code. Other participating restaurants data will not be available unless they have QR code.

4.6.2 Integrity Requirements

Credit information is never recorded on backend database, thus no chance of leakage.

4.6.3 Privacy Requirements

Will inform user that credit card information transactions will be secured through the use of encrypted connection. Also, will ensure restaurant mangers that data is read from QR code cannot be tampered with.

4.6.4 Audit Requirements

N/A

4.6.5 Immunity Requirements

Allow QR code to be read only, prevent tampering of data. Always ensure secure connections with processing credit card transactions.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

N/A

4.8 Legal Requirements

N/A

4.8.1 Compliance Requirements

N/A

4.8.2 Standards Requirements

N/A

5 Project Issues

5.1 Open Issues

- Amount of data to be queried is unknown as its ["it's" is the correct spelling, but don't use contractions. "it is" —DS] specific to each restaurant.

- Amount of concurrent clients requesting data is variable. Will effect [affect. I will not be pointing out any more spelling errors, please find them on your own. —DS] efficiency

5.2 Off the Shelf Solutions

5.2.1 Ready Made Products

N/A

5.2.2 Reusable Components

Stripe API will be incorporated as a form of payment system. This will reduce the hassle of reinventing a payment processing system. As well, it insures data integrity.

5.2.3 Products That Can Be Copied

Open table [Emphasize and/or capitalize appropriately for titles/names — DS] offers similar solution. However this product lacks functionality in terms of ordering. Similar UI can be constructed to appeal to mass audience.

5.3 New Problems

N/A

5.4 Tasks

- Develop product prototype and perform testing

- Introduce prototype to supervisor for feedback

- Work on revisions

[Very vague. What about design decisions? What kind of testing? —DS]

5.5 Migration to New Product

N/A

5.6   Risks

- Dependency on third party API Stripe. If company happen to shut down, unable to process credit card payments until alternative solution is found

- Network threats. If new threats are discovered (e.g., heartbleed), may comprise data integrity.

5.7   Costs

N/A

5.8   User Documentation and Training

Work in progress

5.9   Waiting Room

Work in progress

5.10   Ideas for Solutions

Work in progress