



Michigan State University

Team Meijer

Mobile Customer Satisfaction Application

Project Plan

Spring 2014

Meijer Staff:

Jim Becher

Chris Laske

Michigan State University Capstone Members:

Noor Hanan Ahmad Kamel

Kaleb Friskey

Taylor Jones

Table of Contents

1. Executive Summary	3
2. Functional Specification	4
3. Design Specification	5
3.1 Overview	5
3.2 User Interface	5
3.2.1 Landing Page	6
3.2.2 Survey	7
3.2.3 Find My Store	8
3.2.4 Submit Issue	9
3.2.5 FAQ	10
3.2.6 Login	11
3.2.7 Store Ranking	12
3.2.8 Select a Store	13
3.2.9 List All Surveys	14
3.2.10 List All Issues	15
4. Technical Specification	16
5. Risks	18
6. Timeline	19

1. Executive Summary

Meijer is a regional supercenter providing quality food and general merchandise in five states throughout the Midwest. Headquartered in Grand Rapids, Michigan, Meijer has over 200 stores and approximately 60,000 employees.

The Mobile Customer Satisfaction Application allows customers to quickly and easily comment on their shopping experience, providing vital feedback to Meijer. This solution is for android devices and is a proof-of-concept which is planned to be integrated into the existing Meijer application.

The current Meijer application allows users to make a shopping list, find a specific store and obtain information about the store such as phone numbers, address, and inner layout. The current application also enables customers to make a shopping list and clip coupons. The addition of the Customer Satisfaction Application allows users to submit a brief 5 question survey regarding their shopping experience, product quality, cleanliness, friendliness, and availability of products.

In addition, the new application allows customers to submit an issue or concern, should one arise. There is a list of default issues populated in the application which can be modified by Meijer. If the customer wishes to add more detail, an explanation can be entered into a text field and submitted as well. If they wish to be notified about a specific issue, the customer may also include contact information.

The overall objective of this system is to provide a quick and easy way to collect information from the customer to improve future shopping experiences. Meijer already gathers customer experience information for each store. This application will allow them to acquire more data from a larger base of customers more frequently and send that data to a Meijer website. During or after their shopping trip, a customer can open the application and fill out a few brief questions about their general experience. They will also have the option to submit a specific issue and identify themselves should they like a response. If they submit a specific issue, it will be flagged and examined by Meijer staff from the website, which will facilitate an appropriate and timely response. Each such issue will also have an indication about its status such as whether action has been taken, if action is needed, or if the problem is resolved. There is also a text field where Meijer staff can indicate action that has been taken on an issue in order to communicate throughout the organization.

This application is much quicker than filling out a traditional survey. This will allow more customers to provide feedback in a timelier manner, giving a better data set on which to base customer service information. This will lead to better service, increased customer satisfaction, and increased profits.

2. Functional Specification

The primary focus of the project is to provide feedback from customers about their shopping experience quickly and easily. Currently Meijer does gather information about customer satisfaction from their website and on-site in their stores. They take the information gathered very seriously and are always working to improve the service they provide their customers. However, only having a few sources of feedback available limits the range of customers from whom they can get feedback. Also, many customers will be in a hurry or choose not to give feedback.

The application will allow people on the go an opportunity to relay their experiences, with a few brief questions, while allowing the opportunity to expand upon any issue. The user interface makes submitting customer satisfaction information expedient and efficient. The application will eventually be integrated into the existing Meijer application allowing more functionality at no additional input from users. Having this additional feedback will feed into existing database structures which are accessed by Meijer staff from a website. This will allow them to better facilitate responses to issues and give a more accurate picture of the environment of each store. This will lead to more accurate improvements and alterations to make the Meijer shopping experience better for everyone involved.

As a Meijer customer, now you can find coupons for items, find specific stores, compose a shopping list, and, if you are an mPerks member, it keeps track of your purchases and savings. Our proof-of-concept expands the functionality of Meijer's pre-existing mobile application to incorporate customer satisfaction feedback. Customers can submit a five question survey about their shopping experience, report an issue about a specific concern they may have encountered, and access a frequently asked questions page. We will also provide connections between feedback submissions and user-selected stores in order for Meijer to review feedback and improve customer experiences. We have set up a backend, Meijer employee accessible website to view, search, and filter through customer feedback to better enable Meijer to improve their service.

For example, a user in a Meijer store who finds rotten fruit in the produce section can report this issue through the app, real-time, on their mobile device. When submitting, they can provide their personal information if they would like to have a response from Meijer about the issue they encountered. They can also enter an issue in a text field if the issue they had is not a default (provided) option. Issues will be linked to the specific store that the customer is in so Meijer can fix the issue faster. A customer that had a very good or very bad experience in Meijer may want to provide feedback about the overall experience through our five question survey. This submission will also be linked to a specific store so managers and Meijer employees can review submissions and compare service between different Meijer locations.

Meijer employees can access and analyze the data collected from the application through the website. Aggregates of all of the survey responses will be viewable for every store so that they may be compared and so that problem areas, or areas of exception, can be noted and addressed. In addition, each issue will also be listed and can be sorted by severity or status. Employees can also modify the status, severity, or actions taken on an issue allowing communication on how these issues were addressed and what still needs to happen.

3. Design Specification

3.1 Overview

The application will be downloadable to any android device and can be accessed by the customer from their phone by selecting the relevant icon and starting the application. By selecting a specific store, the application will be able to connect responses to the appropriate database while also providing detailed and specific information about a store.

When the user enters the application, the interface will have options to access store information, enter a specific issue, quickly enter a few brief questions about their shopping experience, or view a frequently asked questions page.

3.2 User Interface

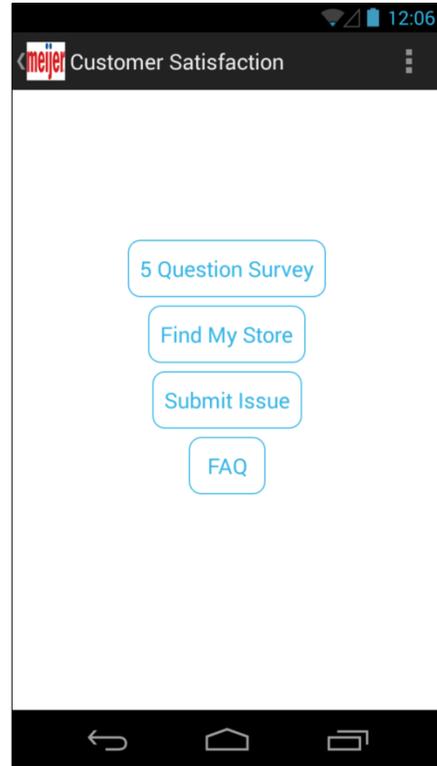
Upon starting the application, the user will arrive at the landing page. Each option should prove to be simple and intuitive. The options from the landing page are to take a survey, submit an issue, select a store, and view frequently asked questions. A menu option on every page will allow swift navigation between pages at any time from any page. The goal of the user interface is to mirror the layout and flow of the current Meijer application so Meijer can integrate our application into their own.

3.2.1 Landing Page

This is the home page of the application. From here, a user can go to the survey, find my store, submit an issue, and view FAQ. A dropdown menu with these four options will provide access to each of these pages from any of the other pages.



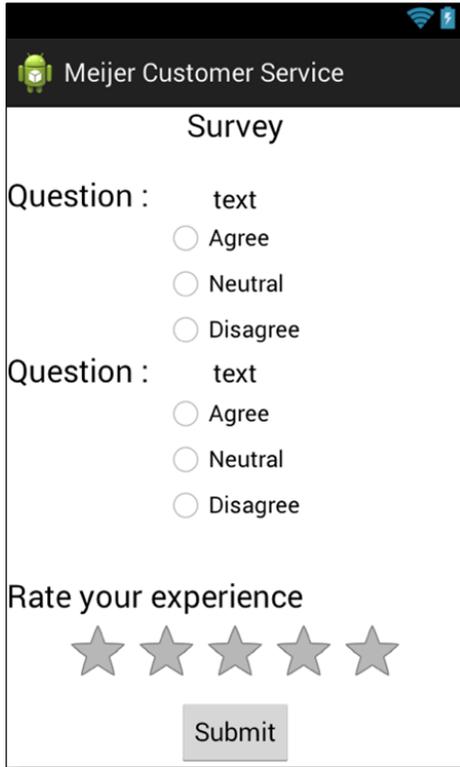
[Figure 1 – Initial design of the landing page]



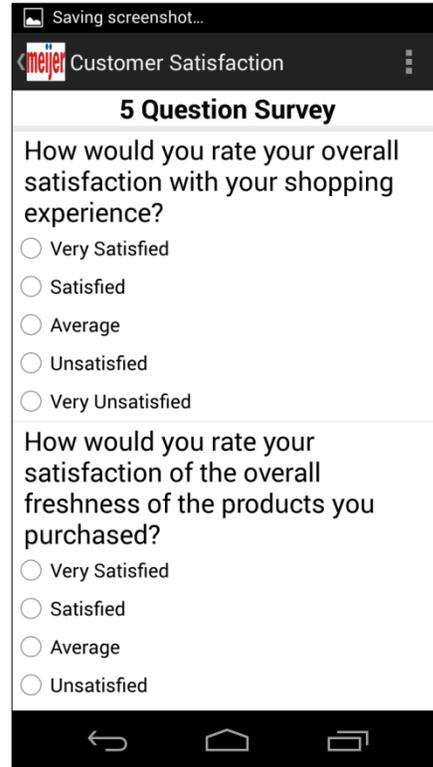
[Figure 2 – final design of the landing page]

3.2.2 Survey

This is the page which hosts the customer satisfaction survey itself. There will be 5 brief questions about their Meijer shopping experience. Each of the questions has a rating system going from 1 to 5. Each of the options has an associated radio button. Response options will range from very satisfied, satisfied, average, unsatisfied and very unsatisfied. Should a user not respond to a specific question, the value will be empty and the data will reflect that the question was not answered.



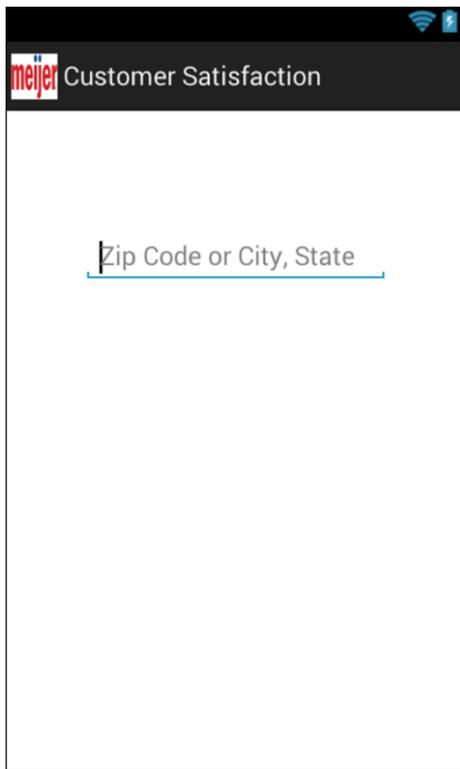
[Figure 3 – initial design of survey page]



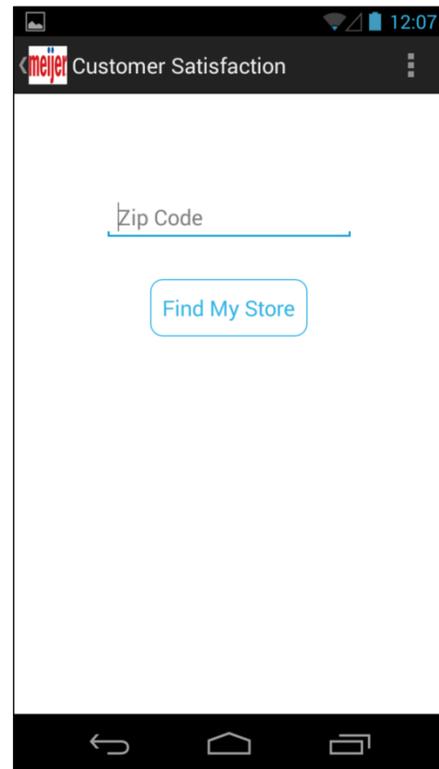
[Figure 4 – final design of survey page]

3.2.3 Find My Store

This page allows users to search for a specific Meijer Store based on zip code, city/state or latitude/longitude via global positioning system. This application only allows a zip code to search stores ordered on distance from the entered zip code. However, the existing application to which this application will be merged automatically searches based on the latitude/longitude position of the phone (assuming gps is allowed), and enables the search to be modified based on an entered zip code or city/state combination. The main purpose of the store selection option is to direct the feedback to a specific store.



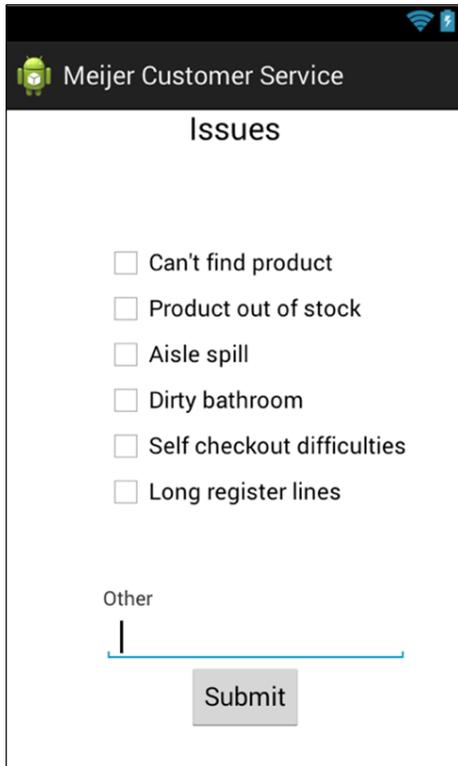
[Figure 5 – initial design of store location]



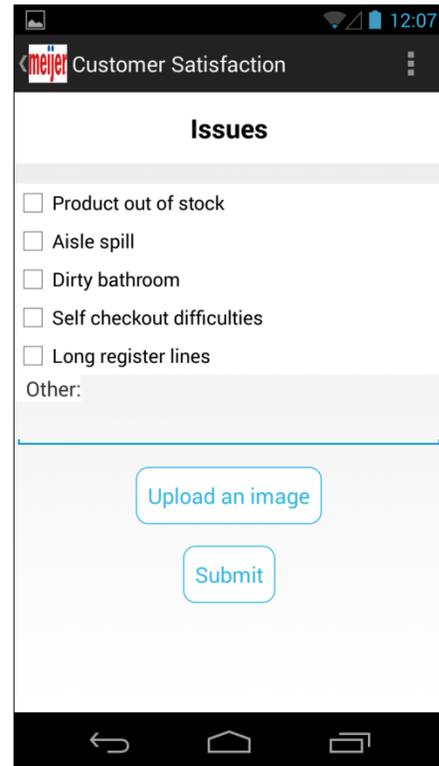
[Figure 6 – final design of store location]

3.2.4 Submit Issue

If there is a specific issue experienced by a customer in a particular Meijer store, this allows them to submit it directly. It will ask them to select a specific Meijer store and enter either a preselected list of issues or enter information in an open text box for another issue, or some combination of both. They will also be prompted to identify themselves, should they want a response from Meijer, via entering their mPerks number (which is also their phone number), their name, and email address.



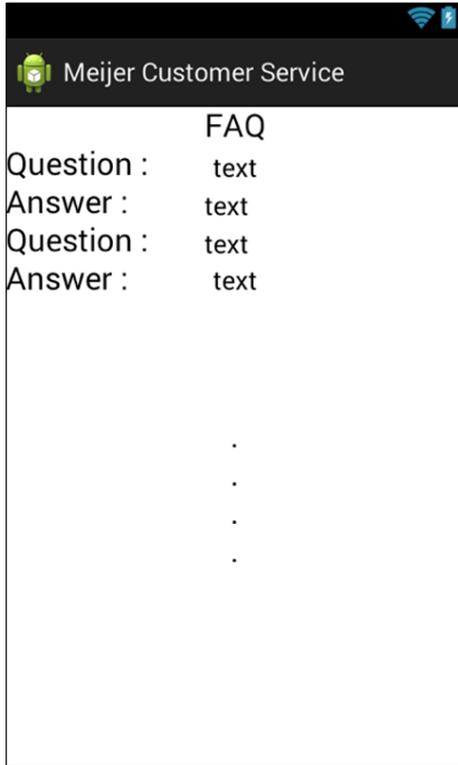
[Figure 7 – initial design of issue page]



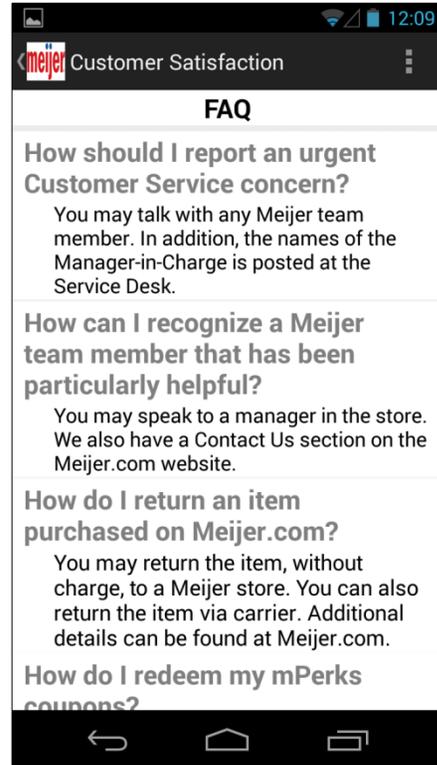
[Figure 8 – final design of issue page]

3.2.5 Frequently Asked Questions

This will be a list of frequently asked questions and their answers. This is a static list of questions and answers that can be modified by Meijer staff to better facilitate the needs of their customers.



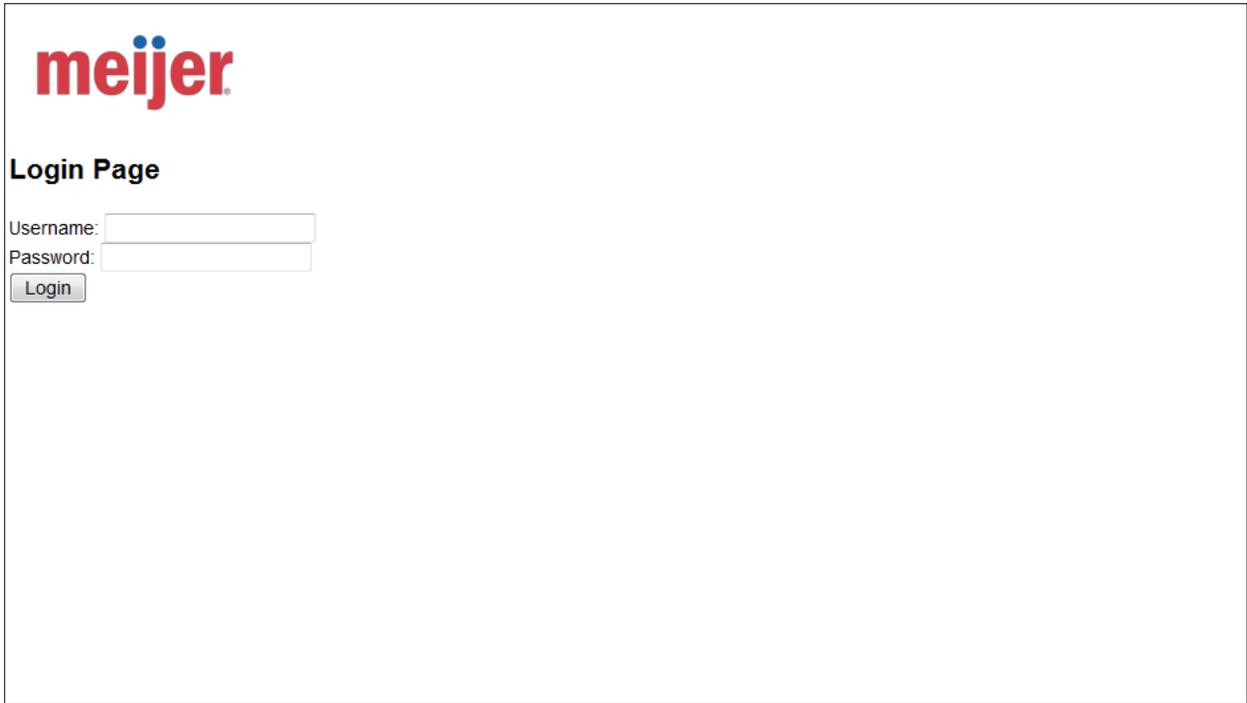
[Figure 9 – initial design of FAQ page]



[Figure 10 – final design of FAQ page]

3.2.6 Login

This will be a login page to allow only Meijer Staff to view and modify the data through the website.



meijer

Login Page

Username:

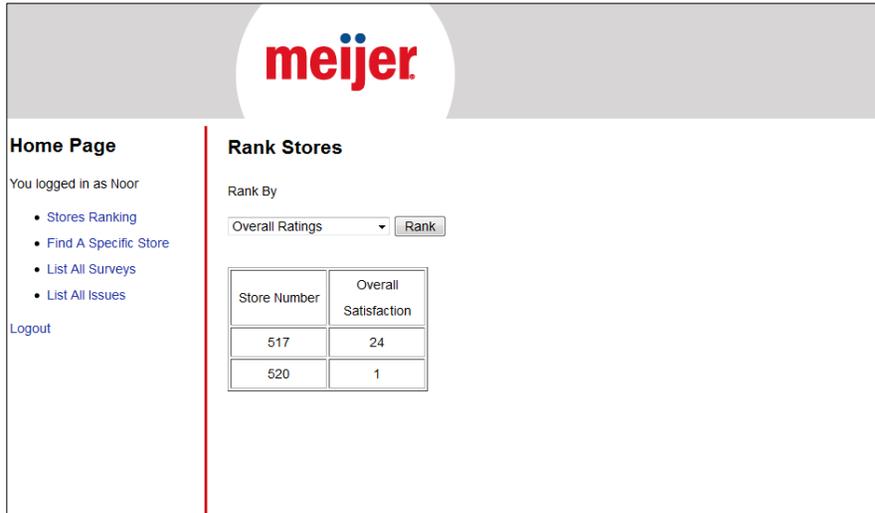
Password:

Login

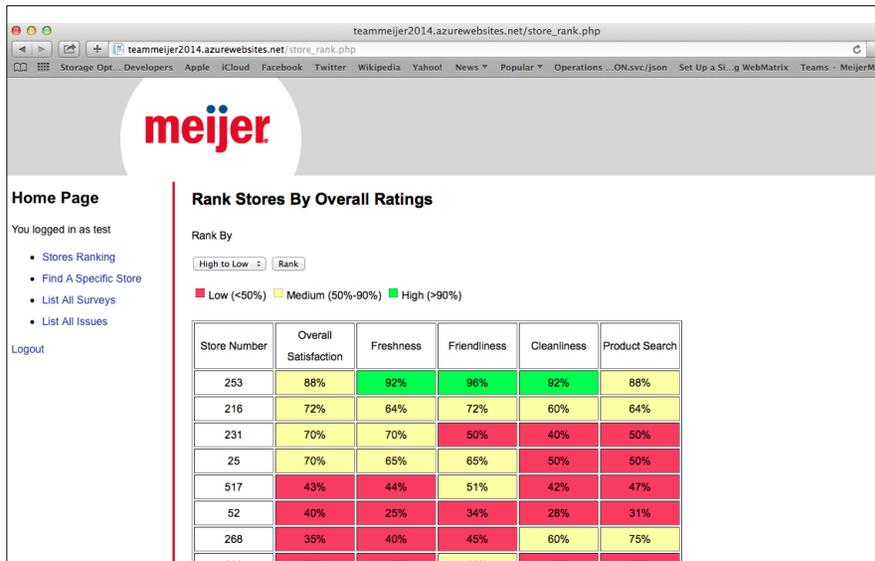
[Figure 11 – initial design of Login page]

3.2.7 Rank Stores

This will be where every store in the database is listed and compared based on the submitted surveys. Aggregate responses are color coded so that it is easy to detect which areas stores are struggling with or which areas they are excelling. Red is for values less than 50%, green are for values above 90% and yellow are in between.



[Figure 12 – initial design of Store Ranking page]



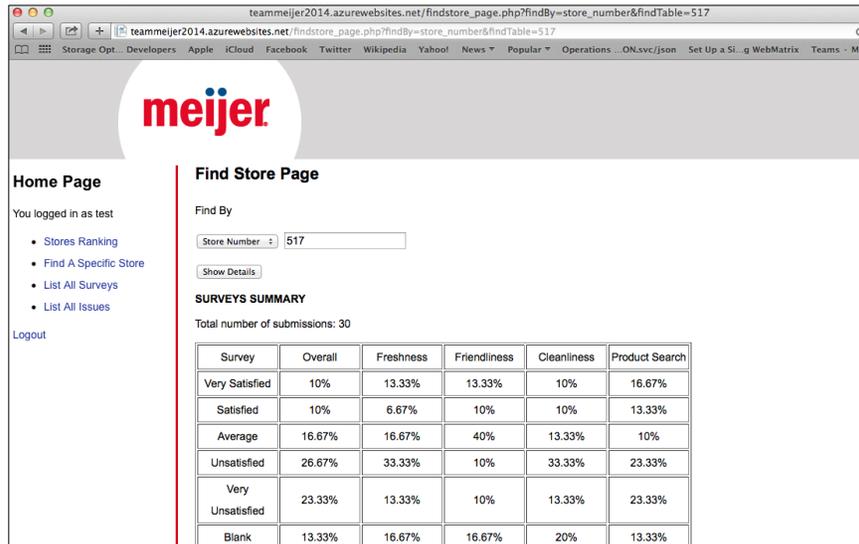
[Figure 13 – final design of Store Ranking Page]

3.2.8 Find a Specific Store

This is where, by entering a store number, the page with all of the information about that store is listed. There will be a table of the distribution of survey responses for how many of the surveys associated with this store had the value “satisfied” selected for Cleanliness for example. This way, it provides an overall aggregate look at how the store is doing. In addition, there is a complete list of every issue associated with that store.



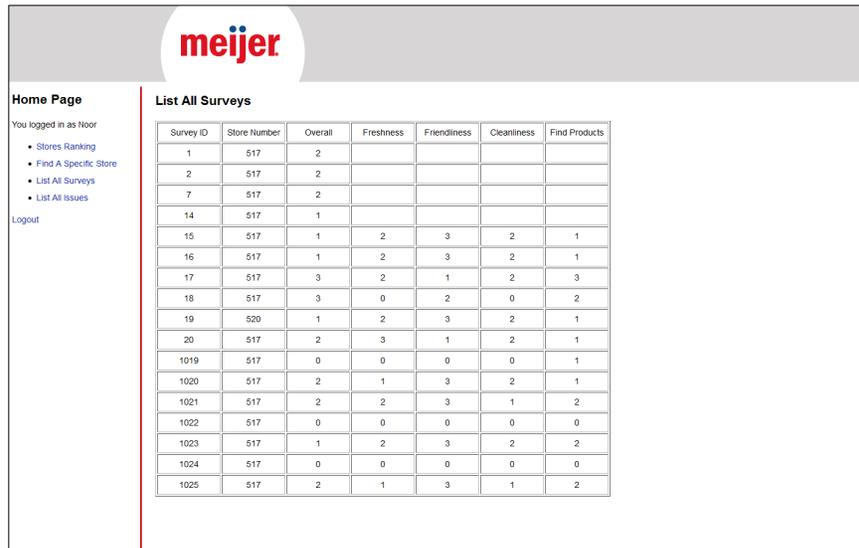
[Figure 14 – initial design of specific store page]



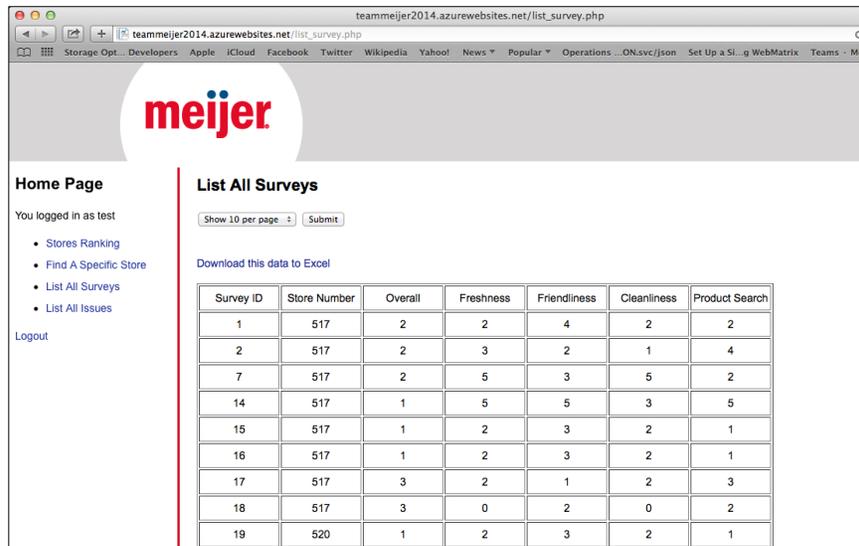
[Figure 15 – final design of specific store page]

3.2.9 List All Surveys

This is where every survey submitted will be listed. One of the fields is the store number so that a user can tell which store a survey is associated with. This allows paging so that when the number of surveys gets to be very large, a user can restrict how many are displayed on a page. This page is exportable to Microsoft Excel so the Meijer staff can further sort and display the data.



[Figure 16 – initial design of list all surveys page]



[Figure 17 – final design of list all surveys page]

3.2.10 List All Issues

This is where every issue submission is listed. Issues are color coded by their status, with open issues in green, issues in progress in blue, and closes issues in red. Various fields are clickable as denoted by their blue text which will pull up a window elaborating additional fields of information such as customer contact information if it was submitted for the issue. That is also where a text box pops up to allow the actions taken field to be entered and edited.

Action	Issue ID	Store Number	Messy Aisle	Dirty Bathroom	Long Register Lines	Self Checkout	Out Of Stock	Other	Severity	Status
View	9	517	0	0	0	0	0	0	0	Open
View	1010	517	0	0	0	0	0	0	0	Open
View	1011	517	0	0	1	0	0	1	0	Open
View	1028	517	0	0	0	0	0	0	0	Open
View	1032	517	0	0	0	0	0	0	0	Open
View	1034	517	0	0	0	0	0	0	0	Open
View	1036	517	0	0	0	0	1	0	0	Open
View	1039	517	0	0	0	0	0	1	0	Open
Edit	5	577	2				4	1	1	In Progress
Edit	7	517	0	0	1	0	1	0	2	In Progress
Edit	8	517	0	0	0	0	0	0	1	In Progress
Edit	6	543	2				1	1	2	Closed
Edit	4	517	1				1	1	1	Closed

[Figure 18 – initial design of list all issues page]

Action	Issue ID	Store Number	Messy Aisle	Dirty Bathroom	Long Register Lines	Self Checkout	Out Of Stock	Other (click for details)	Severity	Status	Actions Taken	Images
Edit	4107	52	F	True	F	F	F	True	Low	Open		
Edit	4108	52	F	F	F	F	F	F	Low	Open		
Edit	4109	52	F	F	F	F	F	F	Low	Open		
Edit	4110	52	F	F	True	F	F	F	Low	Open		

[Figure 19 – final design of list all issues page]

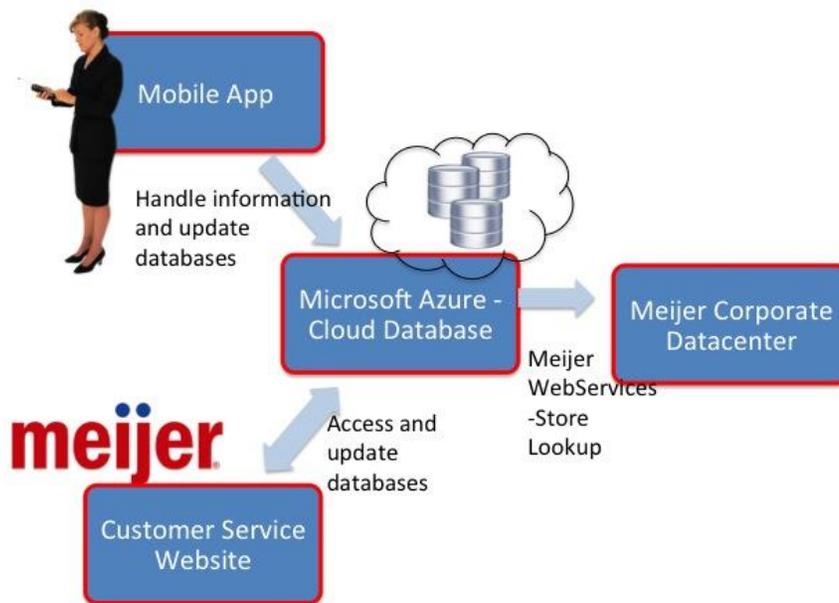
4. Technical Specification

4.1. System Architecture

The main interface for the Meijer customer is the Android downloadable application. When the customer submits an issue or survey, the application inserts a new row into the relevant table to the SQL database hosted by Microsoft Azure. In order to find a store, the application queries the Meijer Restful WebService housed in the Meijer Corporate Datacenter with the zip code the user inputs, and returns the relevant stores. The website is hosted on Microsoft Azure and is written in html php. The website only interacts with the database and queries tables multiple times based on the information to be displayed and analyzed by Meijer employees.

The three main components for the system are

1. Java app run on android devices
2. A website written in HTML/PHP
3. An SQL database



[Figure 20 – diagram of system architecture]

Customers interact with the system using the android application. Users would download the application from Google Play and launch the application by clicking the application icon, which is the Meijer logo. As the user fills out either a survey or an issue and submits, the application sends submitted data to the database hosted by Microsoft Azure. Azure hosts a SQL database with multiple tables; most importantly, a table for all issue data with a randomly generated id, a store number, Boolean issue options, a text field for open input, and email, name, and mPerks fields, if the user provides those values. The survey table stores a randomly generated id, an associated store number, and integer values for each of the five questions. A Meijer employee website pulls data from the database to aggregate and

enable access by Meijer staff. The website can filter data to show particular store results and also to compare results between different store locations. The website is built with HTML and PHP and has a login system to ensure access is only to Meijer employees.

The general overview of the application is a series of pages with text fields and clickable fields in the form of checkboxes, radio buttons and text fields. Upon clicking such a field, it will call a function in Java which will take in any data submitted. For example, on the survey page, each radio button selected will be saved as a member variable via a specific function. The variable will exist in the activity, and upon hitting the submit button for the activity, it calls another function which will map the member variables to the variables of a class for the relevant table in our database. For the survey, it is a table with a primary key, and 5 questions represented with integers. Thus, the survey class as created in the application has 5 integer values, and those values will be mapped to the member variables for the radio buttons. Upon opening an http connection it will send those values in the class object as strings or other relevant type for the data and store them in the database. It will then launch a new intent and thus direct the user either back to the home page or the next page in a chain, should they be linked. This allows users to be navigated where they need to go without having to intentionally direct themselves to that page. However, should they wish to, the menu accessible from the top right corner allows instant movement to any of the main four pages. While the specific fields for data collection or submission may change by page, they all follow this general format. If there is no data being associated with a database, the function only needs to launch a new page.

The website is hosted on Microsoft Azure and connected to the Azure SQL database. Upon searching the website, a login page will appear requesting the username and password. As this website will only be used by the Meijer staff, any kind of registration was not implemented for the outside users. Once logged in, a welcome message appears with 4 options on the left of the screen. Each of these options is a link associated with a php page. The first option is the Stores Ranking. This option shows the summary of how each store was rated by customers in 5 areas, which are overall satisfaction, freshness of the products, friendliness of the staff, cleanliness of the store and the ease of finding products. The data is pulled from the database by calling an SQL query and is aggregated for observation purposes. The data is shown as a percentage as it is calculated by the sum of ratings divided by total number of survey submitted for a particular store. It can be sorted based on the overall ranking and it is also color coded for easy viewing, by calling a function that checks the value handed in and setting the color appropriately. It is given a green background when it is 90% and above, yellow when it is between 50% and 90%, and red when it is below 50%. The second option is Find A Specific Store, which will display a summary of the survey data and all issues associated with the store number that was entered. The survey summary is displayed as a percentage, calculated by the number of ratings for each feedback option divided by the total number of submissions. The third option is List All Surveys, which will display all of the surveys that was submitted. It can be viewed 10, 30 or 50 rows per page, with 10 rows per page as the default. The last option is List All Issues. This displays all the issues depending on what the staff picks from the drop-down menu. The issue also displays small image associated, and can be enlarged by clicking on it. When a customer submits an issue, the status and severity are set to be open and low respectively by default. Meijer staff can edit each issue submitted to update status and severity, and also can add any actions taken, which will then be send to update the database. The logout system is also implemented and once logged out, it will automatically redirect to the login page.

5. Risks

Extracting data

- We need to move data from the application to database.
- Mitigation: Doing research into how to connect app to database. Sending data from an application to an online database is something many applications do and, as such, finding examples for how to establish an http connection were readily available.
- Status: Data is successfully sent from the application to the database.

Code versioning

- We will need to host versions of our android code that multiple people can access while also allowing us to go back to a prior version in case of emergency.
- Mitigation: Researching how to host and rollback to working version, looking into GitHub.
- Status: GitHub is being utilized to version code as well as to facilitate communication between MSU Team Meijer and clients from Meijer.

Store locator

- We will need to be able to discover which store a customer was in when they submit an issue or survey.
- Mitigation: Contacted client, looking into JSON, RESTful Webservice to mimic how this is done with the existing application.
- Status: Successfully submitting an HTTP POST to send data to the RESTful webservice and receive the response back in the application.

Feature creep

- Throughout planning and design, we have many opportunities to add more features to the product.
- Mitigation: Time management, prioritizing, and compromising with our client.
- Status: Tasks have been assigned a priority. Team is implementing features from most important/central to the purpose of the project to more auxiliary features.

6. Timeline

Week 1 (1/6 – 1/10)

- Met with team, established meeting times with course, clients and within team
- Downloaded basic OS and some software
- Initial client conference call

Week 2 (1/13 – 1/17)

- Downloaded GitHub, Eclipse, Visual Studio 2013
- Began project plan
- First drafts of layouts for application and website
- Made test application and basic website

Week 3 (1/20 – 1/24)

- 1/22 – Status Report Presentation
- Establish Microsoft Azure

Week 4 (1/27 – 1/31)

- 1/27 – Project Plan Presentation
- Get data from app to database

Week 5 (2/3 – 2/7)

- Full skeletons for every page of app
- Fleshed out website options allowing different tables to be viewable and color coded

Week 6 (2/10 – 2/14)

- Get Store Locator Working
- Made issues editable from website

Week 7 (2/17 – 2/21)

- Get Pop-up Windows working on application
- Figure out how to cache information locally on device
- Parse JSON response from RESTful webservice for store location
- Edit severity and status, pop-up window for text for “other” issues on website

Week 8 (2/24 – 2/28)

- 2/24 – Alpha Presentation

Week (3/3 – 3/7)

- Spring Break

Week 9 (3/10 – 3/14)

- Caching store information and associating surveys and issues with stored store id
- Aggregate of data on website
- Thank you message for submission
- Applied Crashlytics

Week 10 (3/17 – 3/21)

- Improved navigation of app so flow is more intuitive and easy to use
- Updated UI for app, changed text, formatting and display

Week 11 (3/24 – 3/28)

- Image submission on Issue page
- Bug fixing
- Start project video
- Start user and administrator manual

Week12 (3/31 – 4/4)

- 3/31 – Beta Presentation

Team Meijer: Mobile Customer Satisfaction Application

Debugging

Week 13 (4/7 – 4/11)

Initial aim at product completion to be moved as needed

Debugging

Week 14 (4/14 – 4/18)

Continued debugging, additional final modifications and wrap up

Week 15 (4/21 – 4/25)

4/21 – Project Video due

4/23 – All deliverables due

4/24 – Design Day Setup

4/25 – Design Day