Project Specification Document

Team Red (Student Nutrition Mobile App)

Paul Abts
Gage Askegard
Donovan Beckmann
Brett Chastain
Jordan Falcon
Connor Fradenburgh
Zach Grosz
Tyler Johnson
Victoria Kyereme
Grant Moe
Sethu Monick
Ryan Nelson
Mitchell Olson
Jaron Pollman
James Raboin

Date: 05 December 2016

1. Project Vision and Objectives

1.1 Project Scope and Vision

The foods we eat play a crucial role in determining individual health on both a short term and long term basis, yet to the general public, nutrition remains a complex and impersonal branch of knowledge. Many people struggle to consider nutrition in their food choices, and this can lead to a less healthy lifestyle. We intend to provide a solution to change this and help users achieve a healthier lifestyle by considering nutrition in their food choices.

We intend to provide an Android mobile application that focuses on the experience of diet tracking, while also delivering the tools necessary to help users make better choices about nutrition. These tools include a foods database to display relevant nutritional information with an API that allows for the entry and maintenance of data, a dietary analysis system to calculate a user's nutritional health, and a recommendation engine to guide users toward healthier choices.

We intend to focus our mobile application on the dining options available on the campus of NDSU, while allowing for the possibility of expanding beyond campus in the future. A college setting provides a great opportunity to help students develop habits they can carry throughout their lives. At the same time, the limited number of dining options will allow us to focus on improving the experience of diet tracking and perfecting the tools we will offer. The final delivery of our mobile application will be at the final presentation in early December.

1.2 Project Goals and Objectives

#	Goals
1	Make the system easy to use by minimizing the amount of training necessary to use the application. Users should agree that minimal or no training is required to use this app.
2	Make the system convenient to use by minimizing the amount of time needed to enter in a food item. It should take less than two minutes to enter a food item, when starting from main screen of the phone app. This includes all interactions including searching and scrolling through results.
3	Make realistic / relevant recommendations that will help users make food choices. Users should select one of our recommendations at least 50% of the time.
4	Provide a comprehensive nutritional list of all current food choices available on NDSU campus, including all venues and both fixed and rotating menus of the dining centers.
5	Provide a dietary analysis / nutritional calculation system that follows the best practices in the industry. Calculating and analyzing Daily Recommended Values (DRV) are based on the standards and guidelines of the USDA and FDA. (How to determine an individual's DRV, how to calculate percentages the individual has achieved for the day)
6	Make the system easy to maintain. – As an example, if the database is changed make it simple to re-access the information in the server.

7	Build a prototype that demonstrates the application by October 31. – This for demo purposes and for showing the professor our progress.
8	Design the database to be extensible, with the ability to add venues like local restaurants and home recipes. It should be easy for other developers to add on to this application.
9	Keep a constant velocity of progress. Avoid sleepless nights of coding as much as possible.

2. Project Planning

2.1 Project Lifecycle

The team will use an agile approach in two-week sprints while keeping in mind a high-level vision of schedule and deadlines. Within the sprint, our team will subdivide into smaller groups to take on parallel tasks. Group one will focus on using repeated iterations and prototyping to develop the concepts and functions of the application, while group 2 will focus on developing requirements and system / program design. Each smaller group will collaborate between themselves to complete assigned tasks, and the groups will come together weekly to demonstrate progress and elicit feedback. Each sprint some team members will change teams in order to gain a better understanding of the project.

2.2 Project Setup

#	Decision Description
1	OS: Windows 7, 10, Mac OS for coding
2	Coding Language: Java, MySQL
3	Database: MySQL database Server
4	IDE: Android Studio
5	Code Repository: GIT
6	Team Communication: Slack
7	Project Management: GITHub
8	Document Repository: Google Drive, GITHub

2.3 Stakeholders

Stakeholder	Role
Alex Rademacher	Instructor, Computer Science (NDSU)
Dr. Sherri Stastny	Consultant from Health, Nutrition, and Exercise Sciences (NDSU)
Dr. Amelia Asperin	Consultant from Hospitality and Tourism Management (NDSU)
Paul Abts	Developer
Gage Askegard	Developer
Donovan Beckman	Developer
Brett Chastain	Developer
Jordan Falcon	Developer
Connor Fradenburgh	Developer
Zach Grosz	Developer
Tyler Johnson	Developer
Victoria Kyereme	Developer
Grant Moe	Developer
Sethu Monick	Developer
Ryan Nelson	Developer
Mitchell Olson	Developer
Jaron Pollman	Developer
James Raboin	Developer

2.5 Small Group Assignments for Sprints

Member	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	End
Paul Abts	REQ	GUI	REQ	SPEC	REQ	SPEC	-
Gage Askegard	PM	GUI*	GUI*, PM	GUI*	GUI*	GUI	P
Donovan Beckmann	REQ	REQ	PD	ENG	ENG	ENG	-
Brett Chastain	REQ	DB	SPEC	PD	TEST	TEST	-
Jordan Falcon	PM	REQ	SPEC*	ENG*	ENG*	ENG*	P, POST
Connor Fradenburgh	REQ	REQ	GUI	GUI	GUI	PD	P
Zach Grosz	REQ	PM, SD	PM, PD*	DB	TEST	TEST	P
Tyler Johnson	PM	DB*	DB*	DB*	DB	DB, CI	P, POST
Victoria Kyereme	PM	REQ*	PD	PB	PB	PB	-
Grant Moe	SPEC	REQ	GUI	GUI	GUI	GUI	P
Sethu Monick	REQ	REQ	SD	PB	PD	SPEC	-
Ryan Nelson	REQ, SPEC	PM, REQ*	PM, SD*, PD	PM, SPEC, PD, REQ	TEST PD, PM	SPEC*, PD, D, PM, P	P, POST
Mitchell Olson	REQ	GUI	SD	ENG	ENG	ENG	-
Jaron Pollman	REQ	GUI	GUI	GUI	GUI	GUI, D	P
James Raboin	REQ	GUI	SD	REQ	SPEC	CI	-

*: group leader
PM: Project Management
DB: Database (SQL server)
PB: Phone database
REQ: Requirements
SD: System design
PD: program design
GUI: app prototyping
ENG: app engine
SPEC: spec document
CI: Continuous Integration
D: Delivery
P: Presentation
POST: Postmortem document

Sprint 1: September 11 – September 25
Sprint 2: September 26 – October 9
Sprint 3: October 10 – 23
Spring 4: October 24 – November 6
Sprint 5: November 7 – November 20
Sprint 6: November 21 – December 4
End: December 5 – December 12

2.4 Project Resources

Resource	Description	Quantity
Database Server	A MySQL database server provided by NDSU IT Department	1
Code Repository	A private GITHub	1
Dr. Stastny	The stakeholder that has studied similar nutrition tracking applications and will provide expertise in nutrition and dietary analysis.	1
Dr. Asperin	The stakeholder that has studied restaurant operations.	1
Alex Radermacher	The mentor who will be able to provide us with technical assistance	1
Capstone Team	Our team of students who will be the primary developers of the project	15
Windows Workstation	A windows workstation with Java code for developing the Android version of the software	15
Android Phone	An android phone to be used as test hardware for the mobile version of the software.	1

2.5 Assumptions

#	Description
A1	The entire team will be able to meet face to face once a week during class.
A2	The development test data and student demos will be sufficient to create an accurate prediction of user actions.
A3	Team will have sufficient time to complete a prototype of all features integrated together to present by the end of October.
A4	It will be possible to integrate an Android application and mySQL server.
A5	It will be possible to get nutritional data for all dining venues at NDSU, and add it to the database manually.

3. Project Tracking

3.1 Tracking

Item	Description	Link
Code Storage	Project code will be stored in a GIT repository	https://github.com/TeamRedRocks
Project Doc's and Assignments	Reports, spec and design documents will be stored in GITHub	https://github.com/TeamRedRocks/NDSUNutrition/tree/Development/NDSUNutrition/Documentation
Continuous Integration	Jenkins	??

3.2 Communication Plan

3.2.1 Regularly Scheduled Meetings

Meeting Type	Frequency	Who Attends
Sprint start Meeting	Every two weeks in class (30 min)	Project team
Team meeting / status update	Every week in class (30 min)	Project team
Sprint Planning Meeting	End of previous sprint	Team leaders
Small group meeting	Weekly in class (30 min)	Small group team members

3.2.2 Information to Be Shared Within Our Group

Who	What info	When	How
Team leaders	Task definition, task responsibilities, task deadlines	Sprint start meeting	Word document
Project team	Individual task status, roadblocks	Status update meeting	Verbal
Any smaller group within the project team	Meeting minutes from meetings of smaller groups (what discussed, achieved)	as they occur	Google drive word document with current date
Project team	domain research	as completed	Google Drive word document with current date
Team leaders	High-level project schedule	Sprint start	Google Drive word document with current date
Project team	Documents and Code produced	as completed	Google Drive word document with current date

3.2.3 Information to Be Provided to Other Groups

Who	What info	When	How
Professor Radermacher	Final deliverables	At completion of project	Project Spec. Document, Presentation, Code

3.2.4 Information Needed from Other Groups

Who	What info	When	How
Professor Radermacher	Availability of server, GIT, Trac, date of final presentations, date of any class-wide deadlines	As early as possible	Email Ryan Nelson or Zach Grosz

3.3 Deliverables

#	Deliverable
1	Code
2	Project Specification Document
3	Requirements Document
4	System Design Document & UML Diagram
5	Program Design Document & UML Diagram
6	Server API Documentation
7	Testing Document
8	Postmordem Document
9	Final Presentation (PPT)

4. Requirements (User Stories)

4.1 Overall Description

This project, on its surface, is meant as a way for a user to keep track of their meals. It shows nutritional information for meals in such a way that the user can more easily see their diet. On a deeper level this project will hopefully help the user make better dietary decisions. Overall, the user has all the control with what data the app must gather (from the database) to display.

The app will provide many means by which the user can gather more information on the foods which they eat. The app does this through several different means, like allowing the user to update meals as much as they want to allow for better accuracy. The app displays all vital information in a clear and concise manner for easy understanding by the user. When the user navigates through the app looking for specific information the app will gather all needed information from either the main database of the phone database for the user to sift through.

4.2 Users and Roles

Developer- A group member that works on creating the app and makes sure it works the way the group intended. There are multiple parts of the app being developed (GUI, Database, Phone Database, etc).

App User- An end user of the app who will be utilizing the app to track their meals.

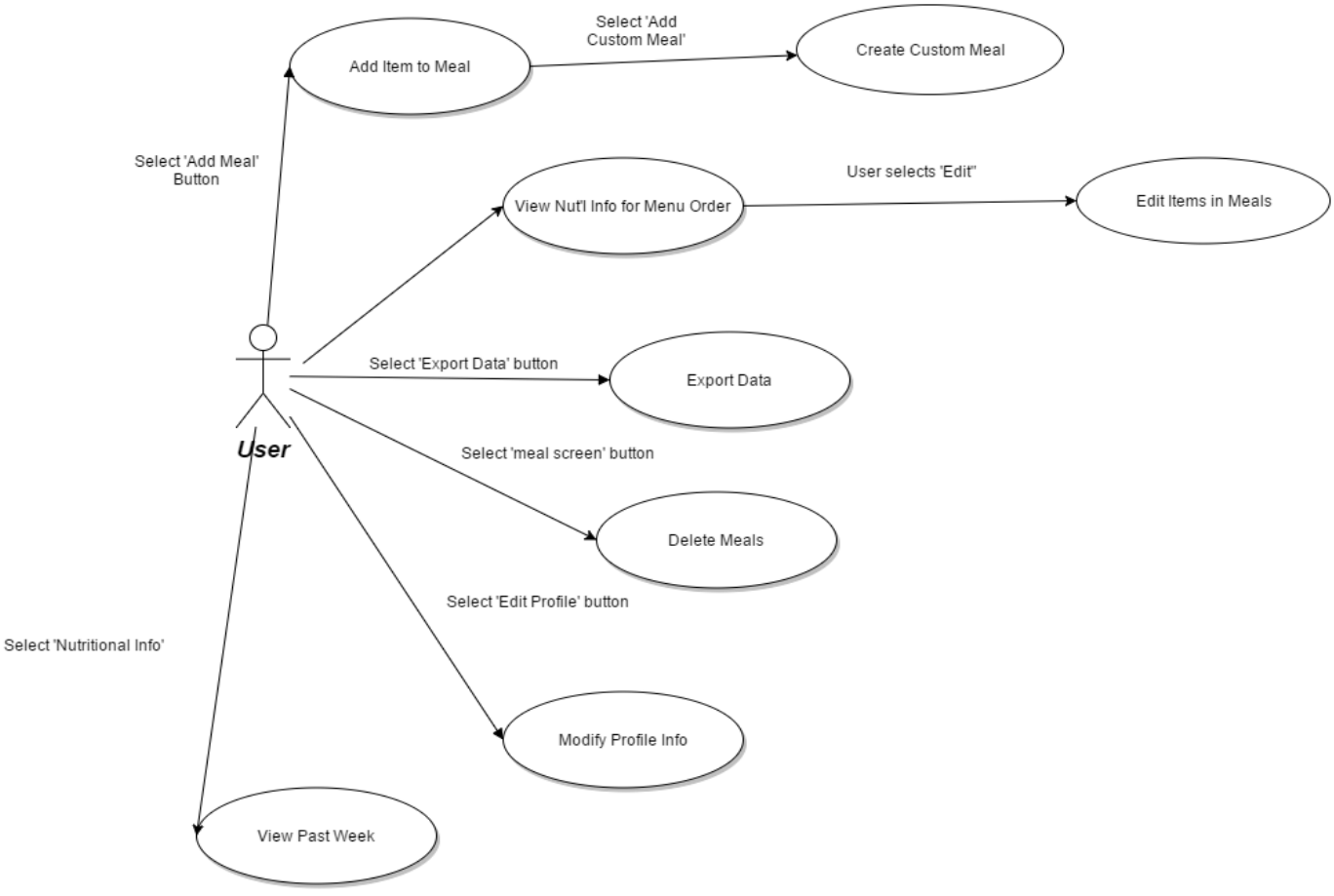
4.3 Use Case

4.3.1 Table of Use Cases

Use Case ID	Use Case Name	Actor	Priority
1	Add Item to Meal	User	1
2	Create Custom Meal	User	2
3	View Nutritional information for a menu item	User	1
4	Users should be able to edit items in meals	User	1
5	Users should be able to export data	User	2
6	Deleting Meals	User	1
7	Modify profile information	User	2
8	View past week	User	2

4.3 Use Case

4.3.2 Diagram



4.3.3 Descriptions

Name	Add Meal Item
ID	1
Description	User should be able to add a food item to a given meal.
Actors	User
Main Course	1. User clicks on the Add Meal button 2. User chooses from a list of meal Items from a certain venue 3. User chooses the portion of the meal item 4. User enters the date, time, location of the meal 5. User confirms meal selection via a “Next” button
Alternate Courses	2a. 1. User decides to add a custom meal 2. User chooses certain food items to add to a meal 3. User chooses portions for each item 5a. 1. User selects the “Home” button which cancels the meal input
Exceptions	

Name	Create Custom Meal
ID	2
Description	User should be able to add a custom food item to a given meal
Actors	User
Main Course	1. User clicks on the Add Meal button 2. User selects to add a custom recipe 3. User selects from a list of custom recipes 4. User selects the portion of the meal, the location, date, and time 5. User selects “Next” and the meal is added
Alternate Courses	3a. 1. User needs to enter in the custom recipe 2. User is directed to the custom recipe screen 3. User enters all food items contained in the recipe 4. User selects “Next” and the meal is added to the custom recipes 3a(a) 1. User selects “Home” and the meal entry is canceled 5a. 1. User selects “Home” and the custom item entry is canceled

Exceptions	

Name	View Nutritional information for a menu item
ID	3
Description	User should be able to view nutritional information for any meal item
Actors	User
Main Course	1. User navigates to the meal screen button 2. User selects an Item from the meal screen 3. User is directed to the nutritional information screen
Alternate Courses	2a. 1. User selects “Home” button and is directed to the home screen
Exceptions	

Name	Users should be able to edit meals
ID	4
Description	User should be able to edit items in meals
Actors	User
Main Course	1. User navigates to the meal screen 2. User selects the “Edit” button for a meal 3. User can change the meal items, date, location, and time of a meal 4. User selects “Next” button to confirm change 5. User is returned to the meal screen
Alternate Courses	3a. 1. User chooses to delete a meal 4a. 1. User selects “Home” button and changes are discarded
Exceptions	

Name	Export User Data
ID	5
Description	User should be able to export nutritional data from the app
Actors	User; System
Main Course	1. User clicks on the “export data button” 2. System prompts the user to choose how much data to export 3. User clicks on the length (days, week, month) 4. User chooses what type of meal to export (breakfast, lunch, dinner, snack, all)
Alternate Courses	3a. 1. User selects day 2. System prompts user to select day or days 3. User selects amount and specific days 4. (back to Step 4 above) 3b. 1. User selects week 2. System prompts user to select amount of weeks 3. User selects amount and specific weeks 4. (back to Step 4 above) 3c. 1. User selects month 2. System prompts user to select amount of months 3. User selects amount and specific months 4. (back to Step 4 above)
Exceptions	

Name	Deleting Meals
ID	6
Description	User should be able to delete unwanted meals or incorrect meal information
Actors	User; System
Main Course	1. User clicks the “Meal Screen” button 2. User navigates to the up or down to find the meal they wish to delete. 3. User clicks on the meal and is transferred to the “Meal Description Screen.” 4. User selects the delete meal button 5. System prompts user to confirm deletion 6. User confirms and the meal is deleted.
Alternate Courses	5b. 1. User clicks no and does not delete meal. 2. System kicks user back to the main menu
Exceptions	

Name	Modify Profile Information
ID	7
Description	User should be able to edit their profile information
Actors	User; System
Main Course	1. User clicks on the “edit profile information button” 2. User is brought to the “profile information screen” 3. User is able to click and edit their age, weight, height, and other vitals 4. Once user finishes changing all vital information they want changed, user can click the save button. 5. System prompts user to confirm that they want to save their information or cancel it.
Alternate Courses	
Exceptions	

Name	View Past Week/Month Information
ID	8
Description	User should be able to view the past week/month of information.
Actors	User; System
Main Course	1. User clicks the “Nutritional Information Screen” 2. System prompts user to select what day/week/month the user would like to see information from. 3. User selects amount of time. 4. System shows a recap of nutritional information.
Alternate Courses	
Exceptions	

4.4 User Stories

1. As a user, I want access to my previous meals at any given time with specifics such as calories, dates, and the like

2. As a user, I want to be able to update my meals that I’ve eaten in real time

3. As a user, I want to know my nutrient intake for today and see if/where I am deficient.

4. As a user, I want to be able to track my water intake.

6. User Interface

6.1 UI Description

We wanted the UI to be very simple and user friendly so that little to no tutorial would be needed for new users to understand and use the system to the best of their abilities.

- When the app is first opened the user sees the home screen (1), this first screen displays their current calorie count using a circular graphic to help show whether their intake is normal. The home screen also will show some suggestions for the user to normalize their intake for the day.

- The next screen is the Nutrients Screen (2) and the user gets to it by swiping right while on screen 1. Screen 2 gives the user the option to look at all the different nutritional information about their day based on the different level of nutrients in their meals for the day. From screen 2 the user can select which nutrient they want to inspect.

- Screen 3 is the “Today’s Meals” screen and the user gets to it by swiping right from screen 2, this screen displays all the meal information for the user based on their meal inputs for the day.

- The last of the main screens is the “Recap Meals” screen (4), again this screen is displayed when the user swipes right on screen 3. Screen 4 displays all the user’s meals in past days for the past week, Additional information can be brought up on these meals by the user.

On all the main screens (1-4) there will be a plus button displayed at the bottom of the screen, this is the “Add Meal” button and when pressed by the user it takes the user to the “Pick A Meal Type” screen (5).

- In screen 5 the user can select what type of meal they are having, a meal from a specific location or a homemade custom meal. When the user selects the Restaurant Meal button on screen 5 it takes them to screen 6, the “Pick A Restaurant” screen.

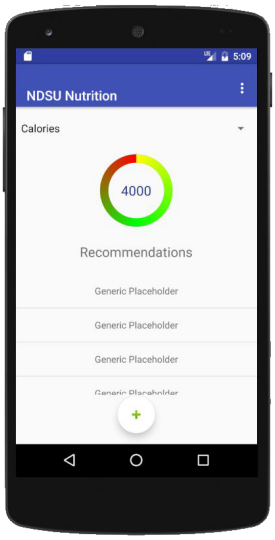
- From screen 6 the user is prompted to add a time, date(which defaults to the current date), and a restaurant which is selected from a drop down menu. Once all of these options are made the user hits the “Select Restaurant” button on screen 6 and the system takes them to screen 7, “Pick A Menu Item”. on screen 7 the user selects a menu item from a drop down menu, these options are based on their restaurant selection from screen 6.

- The user can keep adding items on screen 7 until their meal is complete, once they have all the information they need about their meal they can hit the “Review Meal” button which will bring the user to screen 8, “Review Meal”.

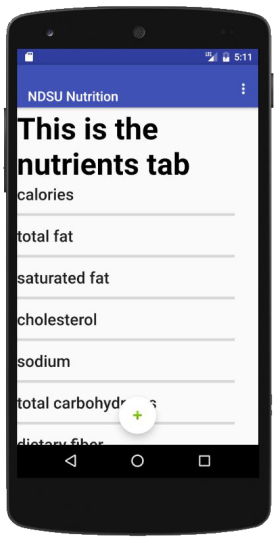
- Screen 8 shows all the information about the meal including, location, date, time, and meal items. From screen 8 the user can hit the “Submit Meal” button the system will take the user back to whichever of the main screens (1-4) they hit the “Add Meal” button on.

6.2 UI Actual (images)

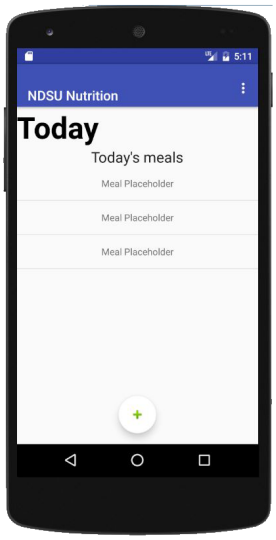
1



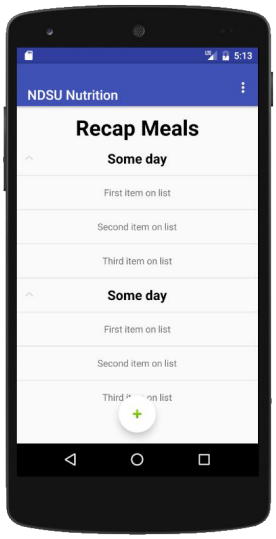
2



3

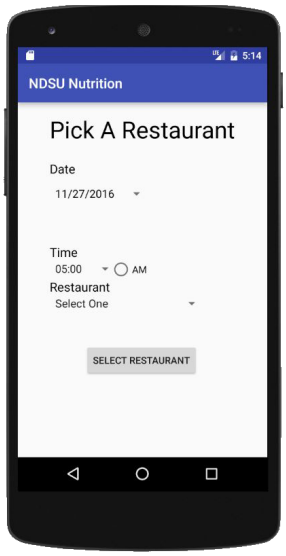


4

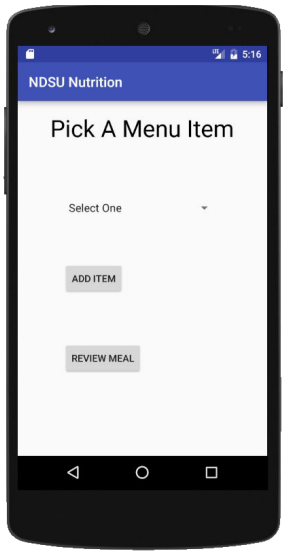


5

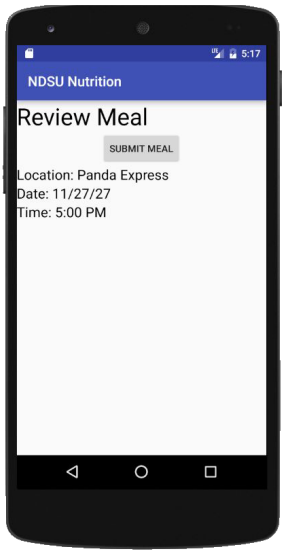
6



7



8



7. Project Closure

7.1 Goals / Vision (revised)

Our original goal was to (1) create an Android mobile application in which users could record meals, view nutrient levels, view past meals, and get recommendations; (2) create a SQL server to store venue and menu item information; (3) create a phone database where user profile information, including all recorded meals, could be stored; (4) include all venues on campus in the SQL server.

Through the course of the project, multiple challenges led us to narrow our focus to the following: (1) create a functioning version of the app that provides the minimum GUI and App Engine functionality to record meals, view nutrient levels, and view past meals; (2) create a fully functioning SQL server to store venue, menu items for each venue, and recommendation information; (3) provide a single venue on campus on the SQL server. In this way, we could focus on providing a complete product with a solid foundation to be built upon by future classes.

7.2 Delivered Solution

By the end of the project, we will have delivered (1) a functional SQL server with data ready to use and the functionality to add information from third parties; (2) working code that implements an Android mobile app with basic functionality for connecting to the server and getting a list of venues; (3) a functioning app engine that communicates between the server and app GUI; (4) high level of documentation, including a Specification Document, Requirements Document, System Design document & UML class diagram, Program Design document & UML class diagram, Testing document, Server documentation, and Postmortem document.

7.3 Remaining Work

This is a list of the major work remaining to be done:

1. Mobile Nutrition Tracking app
    a. Finish communication between server and app
        i. ConstructListOfMenuItems() method
    b. Create working phone database in SQLlite
        i. we have some code and understanding of what needs to be done, but we haven't been able to get it to work with the app as needed.
    c. Create an interface for uploading new server information
        i. So that third parties can submit nutritional information and venues to be added to the server database
    d. Add recommendation function to app
        i. get recommendation listings for SQL Server by working with Nutrition department at NDSU to write appropriate recommendations for all nutrients
        ii. add GUI recommendation feature
    e. Implement remaining "critical" and "medium priority" requirements, which include:
        i. Add additional venues near campus to the server
        ii. Improve quality of user interface in the App GUI
        iii. Add additional screens and functionality to GUI
        iv. User can edit previously recorded meals
    f. Implement "low" requirements and "nice to have" requirements
        i. adding custom meals

   ii. adding user information to calculate a custom calorie level
   iii. export recorded information to a readable, printable format
g. Add in login functionality and the ability to save user profile data in the SQL server

2. Work with Dr. Stastny (NDSU) from the Health, Nutrition, and Exercise Science department to use the app for her restaurant class in the Spring
   a. Her class designs meals, and opens a temporary on campus restaurant (the "800 cafe") with meals of 800 calories or less
   b. She would like to be able to upload venue and menu item data based on the items in these meals
   c. She would like student guests who eat at the restaurant to be able to enter their meal into the Nutrition App to track nutrient levels of what they ate
3. Work with the Dr. Asperin (NDSU) from the Hospitality and Tourism Management department to design and build a second component to the Nutrition Tracking System, which includes cost of items and a way to calculate profit and margins. (see Appendix 1 for more information)

Appendices

Appendix 1 : Restaurant Operations module description

16_1018 Meeting with Dr. Asperin

Participants:
Dr. Asperin, NDSU
Gage Askegard
Ryan Nelson

What was discussed:
Dr. Asperin teaches a restaurant operations course. One part of that course involves determining the menu price of a meal. These are the steps it involves:

1. Students develop a recipe
   a. Choose Ingredients
   b. Determine amounts of ingredients
   c. Determine number of servings recipe makes
2. Students determine cost of raw ingredients
   a. Calculate number of servings needed
   b. Calculate amount of ingredients needed in purchasable units
   c. Choose supplier (grocery store)
   d. Choose brand of the ingredient to purchase
   e. Summation of cost of all ingredients
3. Student determines the menu price of the recipe / meal
   a. Determine cost per customer serving
   b. Determine percentage of profit / overhead
   c. Calculate menu price for the menu item
   d. Summation of all menu items selected to determine cost of full meal

Currently, Dr. Asperin uses an excel spreadsheet to manually enter all values. There are some common issues her students run into:

- Not using the correct units for ingredients for purchase. Some ingredients are sold by the pound, some by the fluid ounce. Getting this wrong means you don't have enough for making the meal.
- Recipes often call for smaller ingredient amounts, such as teaspoon or tablespoon. Need to convert that into standard purchase units (tsp à fluid ounces)

Some other thoughts she had

- Way to switch out ingredients, such as if someone has a gluten allergy, what ingredients are removed and what are added
- How does that change the overall menu price for the meal (upcharge)

How it ties to our project

- This is a second part to a larger food database project. Our part of the project is developing an app for nutrition tracking, accessing the nutrition information of the database. Part 2 would be creating a restaurant ingredient costing desktop application which performs the features listed above. It could use the same database, once costs were added to items. This project could be developed by next semester's group of students to tie into our project.
- Our project won't change scope, except for designing the database to be able to be extended by future classes.

Her ideas for our project

- She likes the dashboard icon and how it presents a snapshot of your current health
- She would like to see the circle be divided into categories of where those calories came from (fats, protein, carbohydrates, etc.)