Project Plan: Android Recipe Application

Jeffrey Carey, Sarah Gardiner, Seth Perts, Tiffany Lower

1. Introduction

1.1 Purpose

The purpose of this document is so the client, Mrs. Jeanne Campbell, can understand how the development team, are going to develop her application. This document will also give the client a timeline, list of milestones, and deliverables of what the team is going to produce and work on for the application to be successfully completed in the alloted time.

1.2 Scope

This software is going to be used by the client, Mrs. Jeanne Campbell, and will perform a variety of functions. Those functions include saving recipes, uploading recipes, saving recipes to a week, creating a grocery list for the week, and possibly the ability to cycle recipes on the application. In addition, the application will allow users to search recipes by tags and export recipes via print/ email. Jeanne will be the administrator of the application and she has undefined users in mind, and the development team has high hopes to publish it to the Google Play Store for other people to utilize.

1.3 References

This project plan has been based off both of the requirement documents created by sections one and two in the 430 Software Development class. It is also based off of many sets of notes provided by the client, Jeanne, and the website 5dinners1hour.com.

1.4 Overview

The remainder of this document will show how the team plans on developing the application. Section two will go into greater detail about the application. It will give more information about the client and the user, and will give a detailed overview of the functions of the system. Section three will explain and justify the project schedule. It will layout the milestones and give estimates for phase completion as well as contain multiple tables and graphs to visualize this information. Lastly, the appendix will include a glossary, author information, and the additional documents that were utilized to create this project plan.

2. Project Description

2.1 System Overview

The main user and administrator of the application is Jeanne Campbell. The application will include functions similar to those from 5dinners1hour.com and will be developed utilizing Android Studio. The administrator will have full control of the application and be able to add users and manually enter all recipes used in the application via a form. All users of the application will have an account that will be stored via a database. Users will able to browse the library of recipes through the use of predetermined tags, and quick add recipes to their shopping list. The tags will include "all", "5 ingredients", "beef", "freezable", "kid favorite", "low carb", "meatless", "one pan", "pork", "poultry", "seafood", "skinny" and "slow cooker". Each recipe will include title and tags, ingredients, ingredient amounts, serving size and assembly instructions. Pictures may be added to the recipes. When a user is finished adding recipes to their shopping list, a compiled list including all the ingredients will be generated for the selected recipes. The ingredients will then be combined across selected recipes and sorted into categories such as "canned/packaged", "dairy/meat", "fresh produce", "frozen", and "pantry staples." Users will be able to glance at their shopping list containing all necessary ingredients and "check off" the items from the list. The user may also "clear" their shopping list which will remove all items and recipes from the list allowing a new list to be started.

2.2 Client Characteristics

Jeanne Campbell is the Computer Science departmental office manager at the University of Mary Washington. As someone who values time, Jeanne is interested in a system that is able to store her favorite recipes and generate categorical lists for the ingredients for said recipes. Currently, she is paying for a service from 5dinners1hour however, the new application will incorporate the majority of the functionality that 5dinners1hour provides while being free to use.

2.3 User Characteristics

The intended end user of this system will be the client, Jeanne Campbell. Currently, there aren't any other specified users of this application but, the application will be able to accommodate additional users down the road. There will be 1 administrator and at least 1 user.

2.4 Functional Requirements

1. The application shall have a "add to list" button that adds the ingredients of the selected recipes to the shopping list.
2. The application shall generate a sorted grocery list of ingredients from the user selected recipes in the application.
3. The application shall combine all quantities on the shopping list for ingredients that are the same, even if they appear across different selected recipes.
4. The application should have tick boxes next to ingredients on the grocery list.
5. The application shall allow the user to add recipes to a grocery list.
6. The application shall allow the user to see the recipes that are associated with the grocery list.
7. The application shall allow the user to be able to clear the grocery list after they go shopping.
8. The application shall allow the administrator to add new recipes to the recipe database.
9. The application shall allow the administrator to add additional users to the application.
10. The application shall maintain a list of recipes that users can view.
11. The application shall have two types of users; user and administrator.
12. The application shall force registered users and administrators to login to access content.
13. The application shall have a system to search by recipe tags.
14. The application shall allow users to change the serving size of recipes.
15. The application shall have a main menu.
16. The application shall have a log out from account option.

2.5 General Constraints

1. The application must have >95% uptime.
2. The application will be developed for use on the android platform.
3. The application will be connected to the internet.
4. The application must respond to searches in 3 seconds or less.
5. The application will require permissions from the phone.
6. The application will utilize a postgresql database.

3. Project Schedule

3.1 Approach

The first thing the development team plans to do is create a shell base for the application to lay the groundwork for future implementation. Next, will be to create different classes, i.e. a recipe class and an ingredient class, that will store all the recipe information and then will be plugged into the application. The development team will utilize the phone and android emulator to get the application to perform and react how the client requires it to. Then, the following functionality will be implemented: ability to add recipes to the application, save them, and create grocery lists. After extensive testing and debugging of all functions and code, the team will implement the database as a storage option. The development team anticipates this part of the project will require some time thus, there has been ongoing research and additional time allocated for implementation. Lastly, final testing/ debugging will be performed and the final product delivered to the client.

3.2 Milestones and Deliverables

Milestone 1: Shell of Code
>   This is a basic view of each screen the application will need to show. Each view will not do anything at all, just be able to move back and forth between the different views.
>   Subtasks:
>   i.   Splash page creation
>   ii. Login page creation (two subpaths below)
>   1. User views:
>   a. Main Menu page creation (subpages below)
>   i.   View Recipes page creation
>   ii. Grocery List page creation
>   iii. Search page creation
>   2. Administrator views:
>   a. Main Menu page creation (subpages below)
>   i.   View Recipes page creation
>   ii. Create a Recipe page creation
>   iii. Grocery List page creation
>   iv. Search page creation
>   Deliverable: Working shell of the application will be given to the client.

Milestone 2: Recipe and Ingredient classes

These are the classes that will create the Recipe and Ingredient objects. The recipe class will be made up of ingredient objects and the recipe object will eventually be stored in the database.

Subtasks:

    i.   Recipe class creation

    ii.   Ingredient class creation

Milestone 3: Plug in the Classes to the Application

This is where the application will finally start to be able to function. The team will plug the recipe class into the code to be able to create recipes in the application. The recipes in the application will then be serialized and saved to the phone pending database setup.

Subtasks:

    i.   Be able to create a recipe as the administrator

    ii.   Be able to add a recipe to their week as a user

    iii.   Be able to browse recipes as a user

    iv.   Be able to search for recipes using tags

Deliverable: Working application showing how to create recipes as an administrator, and adding recipes to the shopping list as a user.

Milestone 4: Be able to Generate a Shopping list

This is when the functionality to generate a sorted shopping list will be implemented. The team will link this function to the Create a Shopping List button on the main page, and the shopping list will be able to be viewed on a different page in the application.

Subtasks:

    i.   Create a Grocery class

    ii.   Have the list display on the grocery list page

Deliverable: Working application with the functionality of being able to generate a grocery list from the added recipes.

Milestone 5: Login Capability

This functionality will allow the administrator to log in and be able to create recipes. Additionally, the user will be able to log in to add recipes to the shopping list and generate grocery lists.

Subtasks:

    i.   Create a Postgresql backend

    ii.   Create the application database and add a login info table

    iii.   Add users and an administrator to the table

          iv.   Plug the backend into the login page

Milestone 6: User Access to Recipes

This functionality will connect the application to the internet so that users can access recipes that the administrator creates and pushes out to the users.

<u>Subtasks</u>:

        i.   Create a recipe table in the Postgresql backend database
        ii.   Plug the backend into the application
        iii.   Add recipes to the database using the application

<u>Deliverables</u>: Fully functional application.

3.3 Work Breakdown Structure

Milestone 1: Shell of Code

Create basic views for the entire application and build out the ability to navigate to different pages.

<u>Schedule Estimate</u>:

     3/13/18 - 3/16/18

<u>Justification</u>:

     This area should proceed quickly once initial setup and understanding of the view navigation is implemented.

Milestone 2: Recipe and Ingredient classes

Define and implement the core classes (recipe and ingredient) that the application will use to store and manipulate the data.

<u>Schedule Estimate</u>:

     3/17/18 - 3/21/18

<u>Justification</u>:

     There may be minor roadblocks with implementing Java inheritance and storing pictures however, the classes themselves should be straight forward creating.

Milestone 3: Plug in the Classes to the Application

This phase will combine the recipe and ingredient classes with the corresponding buttons and views. For debugging/testing, data will be stored on the phones storage using serialization until the database is setup.

<u>Schedule Estimate</u>:

     3/22/18 - 3/27/18

<u>Justification</u>:

There is increased labor for this phase as plugging in the classes with the buttons and views of the application will require learning, trial, and error. Thus, the development team has allocated additional time.

Milestone 4: Be able to Generate a Shopping List

Implementing the shopping list of ingredients where users and administrators will be able to compile a single list of ingredients and amounts. The shopping list will be on a separate page and viewable independently.

Schedule Estimate:

3/26/18 - 4/6/18

Justification:

It is planned that this section will not require to much time and there should not be any setbacks implementing the grocery class and the shopping list view of the application.

Milestone 5: Login Capability

This phase will utilize the Postgresql database backend which will be setup and used to allow login of administrators who can add recipes to the application and users who can add recipes to their shopping list.

Schedule Estimate:

4/6/18 - 4/16/18

Justification:

This phase has been allocated additional time and resources due to the fact that the database and backend will need to be setup. Specific protocols, settings, and access will need to be created and enabled in order to have the application send and retrieve information from the backend.

Milestone 6: User Access to Recipes

This last major phase will connect the backend and database with the application and allow communication between the two. The login pages and recipes will need to be pushed from the application to the backend. As well as allow the administrator to create and upload new recipes.

Schedule Estimate:

4/10/18 - 4/17/18

Justification:

This area will require considerable time to implement as after this milestone, the application will be nearing total completion. At this point, the team should be acclimated and comfortable with the backend and database thus, integration with the frontend and user interface should be manageable.
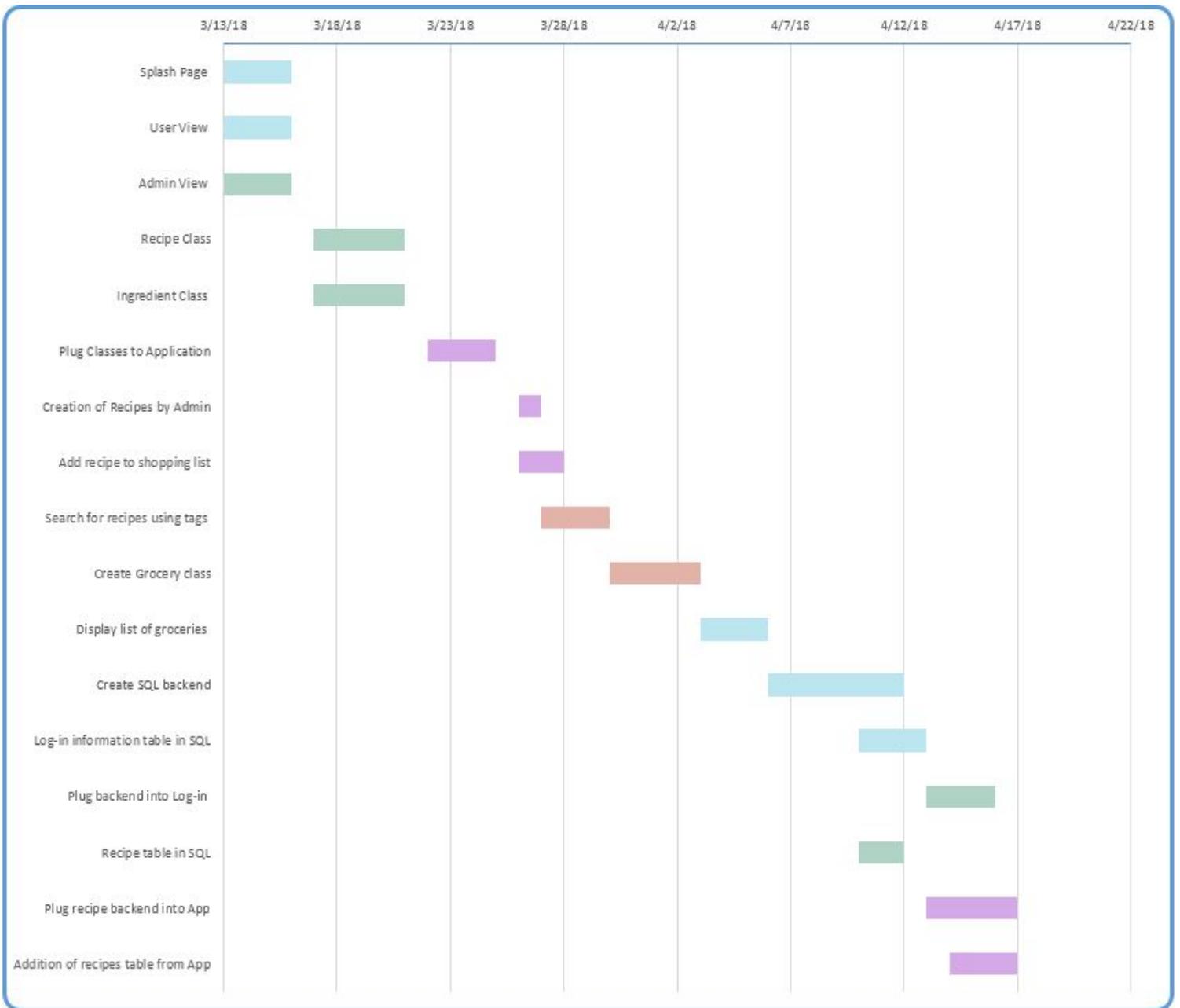
3.4 Gantt Chart

Figure 3.4.1. Visual Schedule of the Gantt Chart (1 Day ~ 3 hours)

Task Manager	Task Name	Start Date	End Date	Duration (Days)	Days Complete	Days Remaining	Percent Complete
Tiffany Lower	Splash Page	3/13/2018	3/16/2018	3	0.00	3.00	0%
Sarah Gardiner	User View	3/13/2018	3/16/2018	3	0.00	3.00	0%
Seth Perts	Admin View	3/13/2018	3/16/2018	3	0.00	3.00	0%
Jeffery Carey	Recipe Class	3/17/2018	3/21/2018	4	0.00	4.00	0%
Sarah Gardiner	Ingredient Class	3/17/2018	3/21/2018	4	0.00	4.00	0%
Seth Perts	Plug Classes to Application	3/22/2018	3/25/2018	3	0.00	3.00	0%
Tiffany Lower	Creation of Recipes by Admin	3/26/2018	3/27/2018	1	0.00	1.00	0%
Jeffery Carey	Add recipe to shopping list	3/26/2018	3/28/2018	2	0.00	2.00	0%
Seth Perts	Search for recipes using tags	3/27/2018	3/30/2018	3	0.00	3.00	0%
Tiffany Lower	Create Grocery class	3/30/2018	4/3/2018	4	0.00	4.00	0%
Sarah Gardiner	Display list of groceries	4/3/2018	4/6/2018	3	0.00	3.00	0%
Jeffery Carey	Create SQL backend	4/6/2018	4/12/2018	6	0.00	6.00	0%
Tiffany Lower	Log-in information table in SQL	4/10/2018	4/13/2018	3	0.00	3.00	0%
Sarah Gardiner	Plug backend into Log-in	4/13/2018	4/16/2018	3	0.00	3.00	0%
Seth Perts	Recipe table in SQL	4/10/2018	4/12/2018	2	0.00	2.00	0%
Jeffery Carey	Plug recipe backend into App	4/13/2018	4/17/2018	4	0.00	4.00	0%
Sarah Gardiner	Addition of recipes table from App	4/14/2018	4/17/2018	3	0.00	3.00	0%

Figure 3.4.2. Breakdown of Gantt Chart Data

The tasks for the Gantt chart were taken from section 3.3 in cooperation with section 3.5 Task Dependency Diagram. In the Task Dependency Diagram, tasks are given a certain number of hours for completion. For the Gantt chart, each day equates 3 hours of work. The development team found that 3 hours per day would allow flexibility with their schedules without overloading. Each task is given a start date and end date. The Gantt chart allows the development team to update it as work is completed as well as in the schedule changes. This provides a visual model of the current progress and how far the application is from completion. The application is scheduled to be completed on April 17th to allow for additional days of debugging and final adjustments.

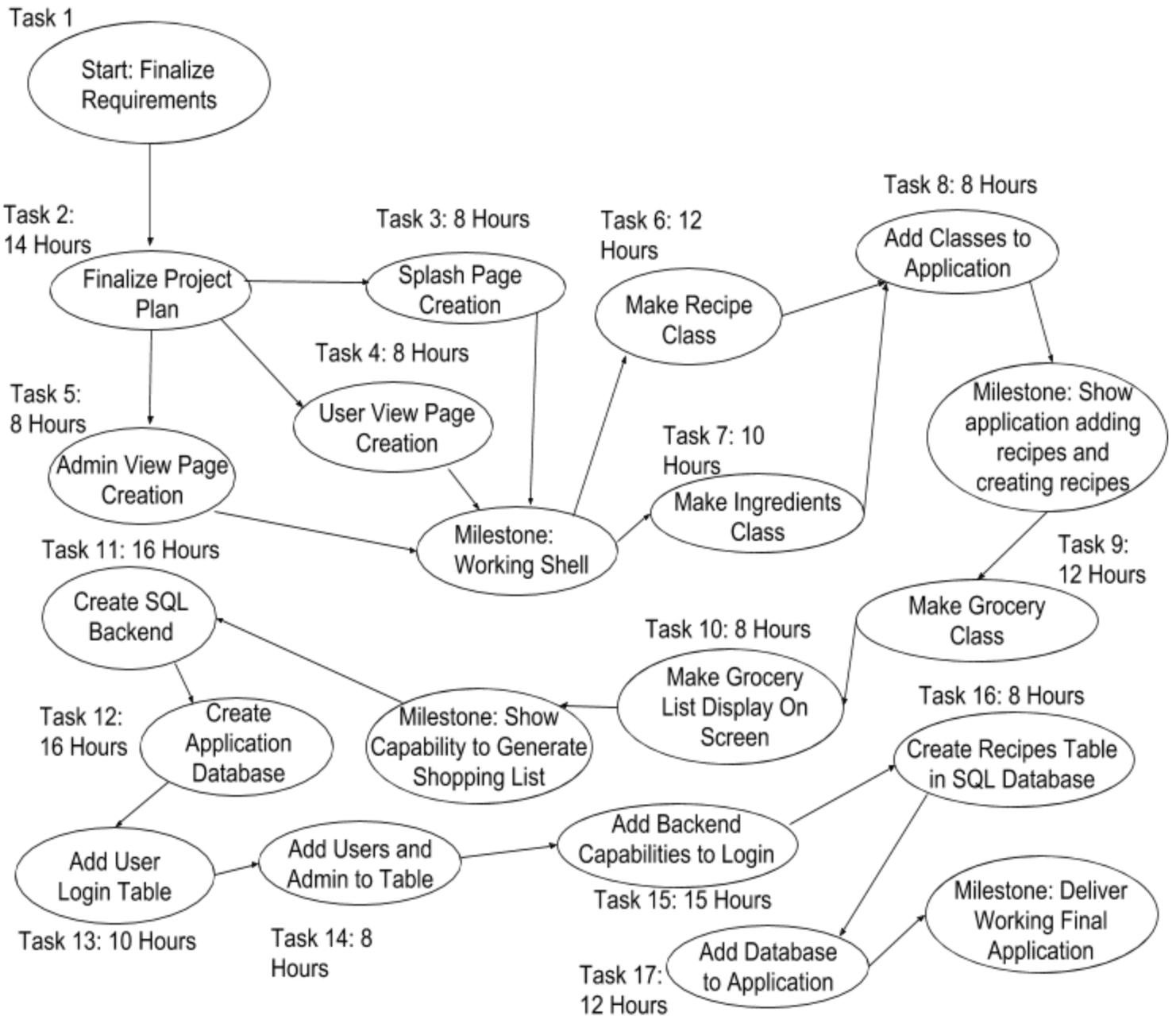## 3.5 Task Dependency Diagram

Figure 3.5.1 Task Dependency Diagram

The system has a fairly straightforward dependency path and the majority of tasks lead into the next, linearly. The dependencies start with finalizing the requirements as it is necessary to pin down what those were before planning for the rest of the project could begin. After the requirements were finalized, the next task was creating the project plan. Creating the project plan was necessary so that the development team had a schedule to work on and new what needed to be done to complete the project. After finalizing the plan for the project, work can begin on creating the code. Tasks 3, 4, and 5 can be done simultaneously as they don't need to interact with each other. After the team has completed these three tasks, the pages that are made in them will be shown to the client to make sure they are happy with the format. Once the client has approved, work on making the classes that will store the recipe data can begin. The ability to add and create recipes will be demonstrated to the client for their approval. Work on task 9 will begin once the client has approved of the adding and creating recipes works followed by task 10. At this point in the project, almost all tasks depend on previous tasks so the project becomes much more linear in its work. Once task 10 is completed, the ability to generate a shopping list will be demonstrated to the client. It will be essential that the shopping list is in the form the client wants and has all the capabilities. As soon as the client says that the shopping list is in the form they want, work on the database back end will begin. These tasks were assigned the longest time estimates as they are the tasks the group feels least confident in and has the least experience in. Creating the database can only begin after the cloud server backend to support it has been created. Once the database has been completed the user login table will be added followed by the user and administrator tables. Task 13 falls before task 14 because the ability for a user to login must be available before a user can be created, otherwise, a user could be made with an account that could never be accessed due to the fact no login exists. After the tables have been created, the actual functionality to login will be added to the application. Once the login functionality is completed and the team has finished debugging and testing, the last few tasks can begin. The team will create the recipe tables and add them to the database. After all the tables have been added to the database, the database will be connected to the rest of the application and final error checking and testing can begin. When the team has finished final optimizing, the application will be delivered to the client in its completed form.

4. Appendix

4.1 Glossary of Terms

Android Application: Mobile phone application created for devices running the Android operating system.

Class: Java class.

Database: Organized collection of data.

Dependencies: Tasks that are prerequisites to other tasks.

Gantt Chart: A type of bar chart that illustrates a project timeline.

Splash Page: First page you see when you open an application telling the user the application name.

Postgresql: A specific structured query language used in the database server.

Tasks: The separate parts of the project that will make up the final deliverable.

4.2 Author Information

Jeffrey Carey
        Section: 3.5, 2.4.1, 2.4.3, 2.4.9, and 4.1.
Sarah Gardiner
        Section: 1.1, 1.2, 1.3, 1.4, 2.4.2, 2.4.6, 2.4.8, 3.1, 3.2, and 4.1.
Seth Perts
        Section: 3.3.
        Additions, grammar & formatting adjustments to entire document.
        Final proofread.
Tiffany Lower
        Section: 2, 2.1, 2.2, 2.3, 2.4.14, 2.4.14, 2.4.16, 3.4, 4.1 and 4.3.

4.3 Additional Documents

5dinners1hour.com

2018-02-16 | Jeanne Campbell

Ingredient	Amount
Canned/Packaged	
☐ Dijon mustard (or brown mustard)	3 tbsp
☐ Orange juice	1/3 cups
Dairy/Meat	
☐ Chicken tenders	32 oz
☐ Smoked sausage	64 oz
Fresh Produce	
☐ Bell pepper ( )	2
☐ Bell peppers	4
☐ Onion	2
☐ Orange ( )	1
☐ Salad greens	18 cups
Frozen	
☐ Frozen mashed potato steamers	4 bags
☐ Steamers brown rice	2 bag
Pantry Staples	
☐ Balsamic vinegar	4 tbsp
☐ Cajun seasoning	2 tsp
☐ Dried Italian seasoning	2 tsp
☐ Dried parsley flakes (or fresh chopped)	3 tsp
☐ Honey	1/3 cups
☐ Minced garlic ( )	2 tsp
☐ Mustard (regular yellow)	3 tsp
☐ Olive oil ( )	2 tbsp
☐ Salt and pepper (to taste)	0
☐ Spicy brown mustard	4 tbsp

Figure 4.3.1 Front side of print out from client. Example of list generated from
5dinners1hour.com

Roasted Sausage, Peppers and Onions - CE
GREEN SALAD

■ CLEAN EATING
■ FREEZABLE, PORK

*You can substitute zucchini for peppers (for non pepper eaters) or add them to the recipe.

INGREDIENTS NEEDED ON COOKING DAY	INGREDIENTS NEEDED FOR ASSEMBLY	ASSEMBLY DIRECTIONS
12 cups salad greens 4 bags frozen mashed potato steamers COOKING DIRECTIONS Pour sausage mixture onto baking sheet lined with foil and sprayed with non stick spray. Bake at 400 degrees 20 - 25 minutes. Cook potatoes according to package directions. Serve with green salad.	4 lbs smoked sausage 4 bell peppers 2 onion 4 tbsp balsamic vinegar 4 tbsp spicy brown mustard 2 tsp dried Italian seasoning salt and pepper to taste	Slice sausage into 1 inch pieces. Place in large zipper bag. Slice peppers and onion into bite size pieces. Add to zipper bag. Add the next 3 ingredients into the zipper bag. Sprinkle with salt and pepper. Zip to seal and squish around to mix.

Sheet Pan Cajun Chicken - CE
BROWN RICE & GREEN SALAD

■ CLEAN EATING
■ LOW CARB, ONE PAN, POULTRY

*orange slices are primarily for flavor.

INGREDIENTS NEEDED ON COOKING DAY	INGREDIENTS NEEDED FOR ASSEMBLY	ASSEMBLY DIRECTIONS
2 bag steamers brown rice 6 C salad greens COOKING DIRECTIONS Pour chicken mixture onto sheet pan sprayed with non stick spray. Bake in a 400 degree oven for 20-30 minutes or until chicken is cooked through. Cook rice according to package directions. Serve with rice and green salad.	1 orange 2 bell pepper 2 lb chicken tenders 2 T olive oil 1/3 C honey 1/3 C orange juice 1 T dried parsley flakes or fresh chopped 3 T Dijon mustard or brown mustard 1 T mustard regular yellow 2 t Cajun seasoning 2 t minced garlic	Slice orange and bell peppers and add to large zipper bag. Add remaining ingredients to same zipper bag. Zip and shake to mix.

Figure 4.3.2. Backside of print out from client. Sample of recipes that user has selected.