CS 389 PROJECT DOCUMENTATION

Student takes full advantage of mature software frameworks; and exhibits itself in a separate public GitHub site with

(a) **project requirements and user interface specification** (chapter 4.1,4.2,4.3&4.5)

**Requirements:**

The application shall deliver ideas for meals with user-specified ingredients.

The application shall provide various recipes from one platform.

The application shall save time in finding a snack to make.

The application should supply a professional but simple interface for the user.

**Specification:**

|  |  |
| --- | --- |
| **Function** | Provide possible recipes for the user to make. |
| **Description** | Provides various recipes for the user based on the inputted ingredients. |
| **Inputs** | One or more ingredients. |
| **Source** | Our database. |
| **Outputs** | Recipes that match the inputted ingredients. |
| **Destination** | The screen with the returned recipes with the matching ingredients. |
| **Action** | A string of ingredients are searched within the database to return recipes that have the specified ingredients. Results are returned onto the screen in a list-view to be selected. Once a recipe is selected, information on the recipe is fetched in the database and returned. |
| **Requires** | At least one ingredient. |

Project proposal: Our team plans to create an Android application, called "What's Cooking?" using Java and XML languages. This application's purpose is to provide the user with simple recipes using the basic ingredients available to them. The user will be able to input five or less ingredients in their household and then search through a manually made SQL database for foods that match the requirement. This can prevent searching the browser for complicated recipes and ultimately make cooking quick and easy.

Names: Angela Fong, Jackie Shao, Ryan Buraus, Sebastian Debrowski, Wissam Mateen

Pictures:



Roles:

Sebastian: Functionality, Design Documents- UML user cases, Activity/Sequence diagram

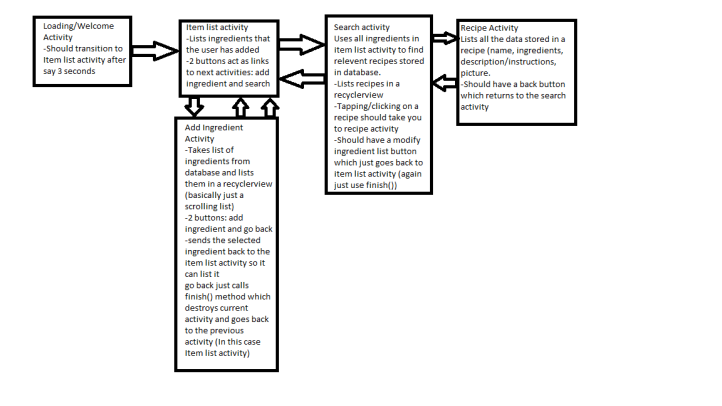
Ryan: Scrum Master, GIThub manager

Wissam: Testing, debugging

Angela: Visual presentation

Jackie: **User Interface**, **User Manual**, Documentation (**specifications**, **goal**, **milestones**)

(b) **design documents** (architectural, interface, UML user case, sequence, or class) chapter 5, 7.1,7.2)



Client-server android application

Object-oriented

(c) source code,

(d) project build deployment instruction

(hint:<https://www.toptal.com/freelance/why-design-documents-matter>)

Testing (chapter 8) (what testing methods you plan to use)

(e) project user manual:

Upon opening What’s Cooking, a title screen with the version number in the top right corner will appear. When it loads the main screen in 2 seconds, there are options to input ingredients along with three buttons; the *add* *ingredient* button appends an ingredient to be searched, a *reset* button to restart the search and the *search* button looks through the database of recipes based on the already inputted ingredient(s). To add an ingredient, there is a full screen menu thatlists the ingredients available in the database. From here, there are two buttons; the *add* button sends the selected ingredient information to the main screen and the *back* button switches back to the main screen, ideally used after the user is finished inputting the desired ingredient.

When the user is finished appending all the ingredients to be searched, a second screen displays the matching recipes for the user to choose from. A *modify search* button is included if the user wishes to change the ingredients searched. If a recipe is chosen, a third screen displays the name of the recipe, along with a picture, an estimated time for cooking, a description of the meal and instructions. There is another *back* button that brings the user to the second screen. There is also a *finish* button that brings the user to the main screen, ideally used when the user is finished cooking with the recipe.

(f) project course report clearly specifying the role of each project team member,

(g) instructor assessment of the project with a rubric, and

(h) a video file of student project presentation

This is just a sample PowerPoint slide deck

(or you could use a similar to this one

https://docs.google.com/presentation/d/1aeNwKil86FSB9InsQpwFA2e3Up8XCd47uG6AJ74rc-M/edit?usp=sharing)

PowerPoint slides

**FUNCTIONALITY:**

What does the application do, and how quickly does it do it?

What are possible failure conditions and how are they handled?

What one-time operations are done at the first execution (i.e., after installation)?

If the user creates entries of any kind (e.g., bookmarks), what are the limitations?

**MILESTONES:**

1: We came up with a list of commonly used ingredients and searched for recipes that involve those specific ingredients. We organized the recipes that involve the list of commonly used ingredients into categories: breakfast, lunch, dinner, snack. All of the data-entry is done in firebase, an online shareable database.

2: We started the coding process for the splash screen to show our logo and app name. After 3 seconds, the splash screen disappears, showing our main screen where we have the search bar.

3: We finished our first activity for the app. The *ingredient* activity reads from the database to display the ingredients we’ve inputted to be added into the searching.

4: We finished our second activity for the app. The *search* activity searches the implemented database for ingredients. We have put in buttons for each of the activities into the app. We also included buttons that will allow the user to easily navigate the app. The buttons are simple, rectangular boxes with their function written on it. The *back* button goes back to the previous screen while the *modify* button changes the ingredients to be searched.

5: We finished the last activity for the app. The *recipe* activity reads from the database to display recipes that match the ingredients searched. Information on the recipe will be displayed along with the instructions, and description.

**GOALS:**

What’s Cooking is an Android application that targets to help the user come up with simple snacks. The user will be able to search through our database of snack recipes based on the few ingredients they input. The goal of the app is to provide convenience to the user, especially for those in a rush. Instead of searching recipes online for each readily available ingredient, the user is able to narrow the results specifically to the recipes that match the ingredients they have.

**USER INTERFACE:**

All of the buttons are enabled and visible on the screens they’re displayed on.

On the main screen, there are three small rectangular buttons on the bottom center of the screen that will remain visible at all times. When the *add ingredient* button is pressed, another screen will transition from the right side, completely overlapping the main screen. Here, there will be multiple rectangular blank fields where the user will fill with ingredients. When an empty field is clicked, another screen will appear from the right side, completely overlapping the current one, with various icons of ingredients with their respective names on them. After a user clicks an icon, the screen will disappear and go back to the previous screen where the clicked ingredient will appear in the first blank field. Next to that is the *reset* button will erase all the already selected ingredients from the fields. Next to the *add ingredient* button is a *search* button that searches for recipes containing the specified ingredient(s).

After the search is complete, a third screen will appear with the resulting recipes. On the bottom is a *modify* button that lets the user change the chosen ingredients. After clicking on a recipe, there is another screen that displays the recipe along with the name, instructions, ingredients, and details. On the bottom will be a *back* button that returns to the previous screen that holds the list of recipes. Next to that is a *finish* button. The button takes the user back to the main screen where the search will begin again.

Each control, including states (enabled/disabled/highlighted) and operations.

Supported orientations and transitions between them.

Functionality represented.

Error handling.

Dimensions and constraints.

**First Slide**

Project Title:

Name of the team members

assign Role

**Second Slide**

Project description

Describe the need

**Third Slide**

Project Requirements

Project Design

User Interface Design

**Fourth Slide**

**Fifth Slide**

**Sixth Slide**

**Seventh Slide**

**……….**

