Meal Helper



User Manual and Project Report for COSC 400
April 27, 2012
Version 2

Project Team
Matthew Burnett, BS Computer Science 2012, University of Tennessee
Ian Harmon, BS Computer Science 2012, University of Tennessee
Jacob Peek, BS Computer Science 2012, University of Tennessee
David Prenshaw, BS Computer Science 2012, University of Tennessee

Project Mentor
Dr. Brad Vander Zanden, EECS, University of Tennessee

Summary. This document describes Meal Helper, a mobile phone application for the Android platform. This application can be used to find specific recipes from the world-wide-web based upon user-specified constraints.

Contents

Design Team 3
Patents, Patents Pending, and Patents in Preparation 8
Specification 8
Claims 9
Drawing 9
Overview of Meal Helper 10
Brief Overview 10
Feature Overview 10
Recipe Info Overview 10
What Meal Helper Can Let You Do 11
Why Meal Helper is Needed 14
Target Market—Who Needs Meal Helper 15
Glossary 17
Warnings and Hazards 18
Warnings 18
Disclaimers 19
Safety Precautions 20
User Agreement 21
User Manual 22
Features 22
Instructions 22
Build Pantry 24
Find Recipe 26
Favorite Recipes 30
Surprise Me 32
Appendix A: Technical Specifications 34
Appendix B: Measurement of Prototypes and Final Product 35
Appendix C: Measurements of Similar Products 38
Appendix D: Design Decisions 41

Design Team

The project design team for Meal Helper includes Matthew Burnett, Ian Harmon, Jacob Peek, and David Prenshaw. Their resumes are attached on the following pages.

Matthew Burnett

2055 Grand Ave. #303
Knoxville, TN 37916-1295
T 731-6140936
lextown38351@hotmail.com

PROFILE

I am currently an undergraduate student in Computer Science at the University of Tennessee. I am seeking a full-time or internship position in Software Engineering. In my free time, I have been researching and developing Android and iOS applications.

EXPERIENCE

Associate, Sam's Club; Knoxville, TN — 2008-2012
Started as cashier, transferred to night cafe position, and left as night merchandiser. As a cashier, I was responsible for member service and maintaining potentially thousands of dollars in my cash drawer. In the night cafe position, I was responsible for producing a set amount of product each night as well as cleaning the area, and putting supplies in their proper place. As a night merchandiser, I was responsible for organizing and stocking several large areas each night by myself.

Manager, WingZone; Knoxville, TN — 2006-2008
As a manager, I was responsible for taking, prepping, and packing orders, answering the phones, resolving customer complaints, distributing orders reasonably to delivery drivers, maintaining a clean and safe work area, and at the end of each night; responsible for counting each cash drawer and depositing potentially several thousand dollars each night.

Manager, McDonald's — 2003-2005
As a manager, I was responsible for a crew of seven to eight members to ensure proper company standards were followed and food was prepped according to company policies as well as resolving customer complaints and issues, and counting each cash drawer before I left each day or night.

EDUCATION

University of Tennessee, Knoxville, TN — B.S. Computer Science, 2012

Pellissippi State Technical Community College, Knoxville, TN — A.A Computer Science Prep, 2010

SKILLS

Fluent in C and C++

Proficient in Java

Experience with Pthreads, OpenMP, Perl, mySQL, and Python

Experience with Xcode, Eclipse, Visual Studio, and similar IDEs

Familiar with Android SDK (up to 4.0 ICS) and iOS SDK (up to 5.0)

RELEVANT COURSEWORK

Database Algorithms I & II — C and C++

Parallel Programming — Pthreads and OpenMP

Database and Scripting Languages — mySQL, Perl, and Python

Software Engineering - Visual Studio and Eclipse

Systems Programming — C

Senior Design - Android and iOS

223 Country Oak Circle
Knoxville, TN 37909
865-226-9248
ianmharmon@gmail.com
ianharmon.com

Ian Harmon

Skills

Technical

Capable in Python, Java, and PHP; have used each for personal projects following proper design techniques.

Familiar with C and C++ through three years of coursework.

Proficient in Windows (XP, Vista, 7) desktop configuration and troubleshooting.

Experienced in Ubuntu and other Linux installation, use, and maintenance; comfortable with CLI commands and workflows.

Adept learner; quick to learn and master new technologies.

Creative

Competent with standards-compliant HTML/CSS web design.

Skilled in Photoshop and Lightroom with image manipulation and photo processing.

Education

Sep. 2008 — Dec. 2012

BS in Computer Science, *University of Tennesee*, Knoxville, TN.
3.62/4.0 GPA.
Recent coursework includes Systems Programming, Software Engineering, and UI Design.

Experience

Jan. 2010 — Oct. 2010

Web Developer and Designer, *University of Tennessee, EECS*, Knoxville, TN.
Designed and implemented a frontend overhaul of department website. Developed an asset-tracking system for internal use in PHP/MySQL. Managed photography and other media used in publications.

Jun. 2010 — present

Photographer, *The Daily Beacon*, Knoxville, TN.
Photography of sporting events, local happenings, and other newsworthy subjects.

Aug. 2006 — May 2007

Audio Technician, *Daniel Boone High School, Theater Department*, Gray, TN.
Co-managed sound systems and mixing board, and assisted in equipment setup.

Apr. 2006 — Aug. 2006

Service Project Manager, *Gray Elementary*, Gray, TN.
Envisioned, secured the resources for, and oversaw the construction of a picnic area by a group of around 30 volunteers. Eagle Scout project.

Miscellany

Earned the rank of Eagle Scout from the Boy Scouts of America.

Former vice president of the UT Student Photography Club.

Jacob T. Peek

University of Tennessee, Knoxville
Honors Computer Science / Mathematics, May 2012

2304 Scarlett Oak Ct. 865.806.0813
Knoxville, TN 37909 jpeek001@gmail.com

Jacob Peek is a computer scientist with experience in parallel programming for systems at large scale, computer security planning, implementation, and monitoring, and system administration and maintenance. His interests are in web design, program development, and cyber security. An experienced team player, his various projects, environments, and concerns lend to his adaptability and utility in any workplace.

Employment History

Extreme Scale Systems Center, Oak Ridge National Laboratory, Oak Ridge, TN January - (June) 2012
ORISE Summer Research Intern July - August 2011
ORISE Summer Research Intern June - August 2010
Collaborated with ESSC staff and interns on *SystemBurn*, a hybrid MPI/Pthreads scalable parallel execution framework and a library of micro-benchmark plug-in components used for testing system functionality and stability at maximum power and cooling load, and qualifying the facility's power and cooling infrastructure for extreme scale systems. Designed and developed software, prepared documentation and technical notes, presented results and demonstrated software to U.S. DoD customer. ORNL and the DoD customer have adopted SystemBurn as key infrastructure for recent and future evaluation, comparison, and acceptance testing of extreme scale computing systems critical to mission. Systemburn has been used to evaluate unclassified systems/facilities exceeding 10MW and classified facilities.

Dunmore Lang College, Macquarie University, Sydney, AU Feb 2011 - June 2011
Computer Support Officer
Responsible for securing, monitoring, troubleshooting, and maintaining academic computing systems and infrastructure. Provided support to team members and end-users. Traced network connectivity issues and helped plan the redesign and upgrade for the computer lab, granting greater hardware, functionality, and support to future residents.

Regal Entertainment Group, Knoxville, TN 2008 – Present
Floor Staff (part time during school)
Working as a senior member of the staff, dealing with customer service and contributing to discussions on marketing and advertising. Previously trained coworkers and currently help delegate responsibilities and workloads.

Computer Skills

Languages: Java, C/C++, MPI, Pthreads, OpenMP, Python, Bash, MySQL, HTML, CSS, Javascript
Operating systems: Windows, Linux, Mac OS

Education

University of Tennessee Knoxville; Bachelor of Science, May 2012 (expected)
Macquarie University; Study Abroad, 2010-2011 Academic Year
Computer Science with a minor in Mathematics. Degree program focused on algorithm and program development, with special interest in parallel computing.

Societies, Honors, and Awards

National Society of Collegiate Scholars
Phi Sigma Theta National Honor Society
Boy Scouts of America, Eagle Scout , October, 2007

Publications

"SystemBurn: Principles of Design and Operation, Release 1.1", Jeffery A Kuehn, Stephen W Poole, Stephen W Hodson, Jonathan D Dobson, David B Reister, Nicholas R Lewkow, S Ross Glandon, Jacob T Peek, ORNL/TM-2011/119, 2011.

David Grantham Prenshaw

Phone (865) 661-8629
E-Mail dprensha@utk.edu
1100 Albemarle Lane
Knoxville, Tennessee 37923

Objective

Searching for internship opportunities for the summer of 2012

Education

University of Tennessee, Knoxville

- Bachelor of Science in Computer Science May 2012
- Overall GPA – 3.87/4.00
- Recipient of Min H. Kao Scholarship
- Member of Phi Kappa Phi

Work Experience

Student Intern, May 2010-July 2010, May 2011-July 2011
B&W Y-12, Oak Ridge, TN

- Worked on development team for web-based application
- Wrote and edited application code
- Helped to draft application's user manual
- Composed application's system state diagram

Office Assistant, June 2005-September 2008
NewLight Presbyterian Church, Knoxville, TN

- Responsible for typing and publishing weekly worship bulletins
- Designed and prepared special bulletins for special services
- In charge of database of church members and church directory pub
- Performed administrative tasks at pastor requests

Computer Skills

Tools: Word, PowerPoint, Excel, Access, Visio, Publisher, Crystal Eclipse, Microsoft Visual Studio, SVN
Languages: C, C++, C#, Java, HTML, CSS, SQL, Perl, Python, PHP, Ja ASP.NET
Moderate Remote Support Experience

Awards and Honors

Eagle Scout (2006)
UT Outstanding Computer Science Senior (2011)
UT Outstanding Computer Science Junior (2010)
Billy J. and Sylvia F. Moore Scholarship (2011)
Charles and Martha Sprankle Scholarship (2009, 2010)
C.W. Smith Kiwanis Club Scholarship (2008)
Service to Humanity Award by the Optimist Club of Knoxville (2007)

Activities / Involvement

Empty Stocking Fund Volunteer (7 years)
Calvary Baptist Church College Tech Team
Pride of the Southland Marching Band
UT Lady Vols Pep Band

Interests

Computers and Programming
Music
Golf

Patents, Patents Pending, and Patents in Preparation

Specification

1. **Title of the invention.** Finding a recipe from the world-wide-web based on ingredient constraints.

2. **Cross-reference to related applications.** No related applications.

3. **Statement regarding federally sponsored research.** The invention will not be made under any sort of government contract and will not be using any federal grant money.

4. **Background of the invention.** It can be difficult at times to come up with recipes based on what food one has on hand. Sometimes it can be difficult or take too much time to run to the grocery store, so one must cook only with what one has in his or her pantry. To do this, it would be possible to look through various cookbooks or recipe collections by hand. Ideally, a computerized method should exist that could find such recipes automatically, and from a much broader database—a database consisting of the entire world wide web, as opposed to a few cookbooks or recipes.

5. **Summary of the invention.** The method for finding recipes involves two main parts. The first part involves gathering the ingredient constraints that are used to search. The next part involves searching the web for recipes.

6. **Description of the drawings.** The drawing below shows the data flow process of the method. First, the ingredient constraints are specified in the application/algorithm. Next, a query on the world-wide-web is performed, and the world-wide-web is searched. The query will return results, which will be processed by the application/algorithm, and will be processed into a recipe format.

7. **Detailed description of the invention.** Finding the ingredient constraints is done by the user. The user will indicate which ingredients are on hand using a GUI on a computer or mobile phone application. The next part involves searching the web for recipes. To do this, certain web APIs may be used. Also, writing a web search algorithm to search predetermined sites may be used.

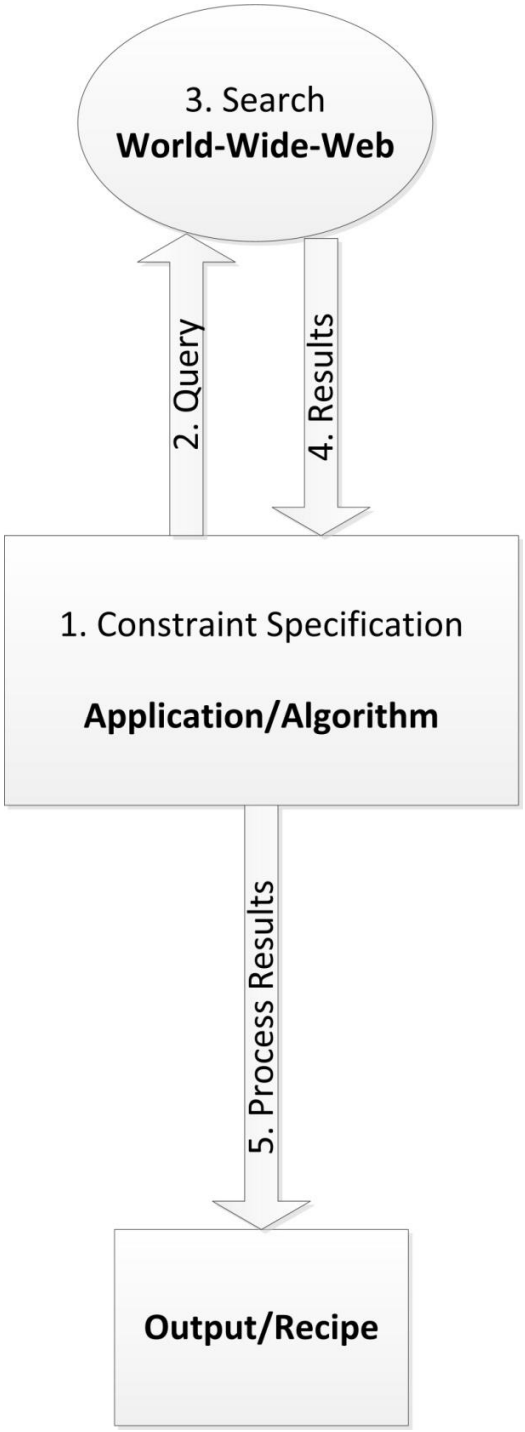
8. **Sequence listing.** Not applicable.

9. **Abstract.** This invention will aid users in finding a recipe using predetermined ingredients. The database for recipes will be the world-wide-web. The medium for this process will be a mobile phone.

Claims

1. Algorithm to search web for recipes, given ingredient constraints.
2. Method to specify ingredients the user has on hand.
3. Algorithm to turn information gathered from the web into a recipe format.

Drawing



Overview of Meal Helper

Brief Overview

Meal Helper is a recipe database application in which a user is able to store various ingredients in his or her physical pantry into the application. After doing so, the application will display the recipes that can be made with only the ingredients in the pantry, unless the user specifies to include recipes with up to a certain number of unowned ingredients.

Feature Overview

Meal Helper is designed with ease-of-use in mind. The pantry is persistent, and will remain stored in the phone's memory (even through reboots and application exits), unless the user deletes a specific ingredient. The recipe feature is based on the ingredients included in the user's pantry. Upon selecting a type of recipe to search for, Meal Helper will display only the courses that can be made with the user's ingredients. Meal Helper includes a "surprise me" feature, which will choose a random recipe based on the time of day. Upon viewing a recipe, the user can mark that recipe as a favorite, and can quickly access it through a favorites menu. Furthermore, the user may indicate a kosher, vegan, vegetarian, and/or gluten-free recipe search.

Recipe Info Overview

Recipes are provided by RecipePuppy.com using the developers' unique search algorithm to filter the results to only show the recipes the user could make. The filter assumes that most users will have common staples such as milk, butter, flour, sugar, etc. Using that assumption, the algorithm combines the pantry and the staples to form a list of ingredients that must be in a recipe before it is displayed to the user.

What Meal Helper Can Let You Do

The following story illustrates an example of where having the Meal Helper application would be beneficial and able to accomplish a task.

Senior Trip

A group of college seniors decide to take a trip to the Smoky Mountains. Being broke college students, they don't have any money to go out to eat, but they do have a fully stocked cabin that has plenty of ingredients with which to cook. They have another problem; they cannot find any recipes in the cabin. So someone pulls out their phone and checks for meals they can make. After finding a recipe they all agree on, they look at the required ingredients and start to look for them in the cabin. Uh-oh! They do not have everything they need. What are they going to do? The closest grocery store is a thirty minute drive. As they talk about what to cook, someone else decides to look for other recipes on their phone. He pulls up the Meal Helper application.



Next, he enters all the ingredients they have in the cabin, then hits “Surprise Me!”



The application pulls up the best sounding Crunchy Cauliflower/Water Chestnut Salad that anyone has ever heard of, he says that sounds good and the application displays the full recipe.



As they start to get the ingredients out, someone notices that they have every ingredient already, so they don't have to go to the store. After cooking the Crunchy Cauliflower/Water Chestnut Salad, they are so happy with how easy it was to find a recipe without having to go to the store, and they tell everyone they know about the Meal Helper application.

Why Meal Helper is Needed

Everyone is looking for new recipes to try, but how do you know what recipes you can make with the ingredients you already have? That is where Meal Helper is needed. By storing your pantry information in the Meal Helper application, Meal Helper allows you to search for new and interesting meals that you can make without going to two or three different grocery stores to buy ingredients. By using Meal Helper, you can save time and gas by not having to go to the grocery store to try something different. Whether it is a quick snack or a full entree, Meal Helper helps you with your meal.

Target Market—Who Needs Meal Helper

In 2009, there were 20.4 million university students in the US. For many of them, this is the first time they have lived away from home, and had to provide for themselves. Coming from this background, the developers of Meal Helper know exactly how this feels. One of the biggest challenges is finding food, as eating fast food is both unhealthy and expensive over a period of time. These convenient meals are loaded in calories, fat, sodium, and other harmful substances. Also, while the average American spends $500 a year on fast food, university students can wind up spending two or even three times that amount. Having just left the house, students are unlikely to have much cooking experience or recipes to use, and many will not bother with finding dishes to prepare with what ingredients they may have picked up at the store.

Built by university students, Meal Helper will do the legwork for the chef-to-be, finding the recipe faster than they could and presenting it in a neat, mobile manner for easy cooking. Being on the user's phone, rather than on a laptop or in a book, makes it much easier to maneuver in the kitchen; this not only expedites the culinary process but also makes for a safer environment to cook.

Another market includes anyone who cannot leave the house for new ingredients. For homebound users, regardless of the reason, Meal Helper turns their stocks of supplies into a veritable cornucopia of choices; with the same list of items, they could make something new every time. While increasing morale, this would also add variety to their tastes and diets.

References cited

National Center for Education Statistics, http://nces.ed.gov/fastfacts/display.asp?id=98

Tsakos, Lisa. "The Real Cost of Fast Food". http://www.naturallysavvy.com/healthy-eating/the-real-cost-of-fast-food

Developers' University Experience

Glossary

Drop Down List	A list of items that cascades vertically on the application screen.
Gesture	A gesture, in reference to an Android application, referrers to the hand movements that result in a zoom action or scroll. To zoom in, push 2 fingers apart. To zoom out, bring 2 fingers together. Scrolling is achieved by simply sliding a finger across the screen.
Ingredient	An item that is stored in the pantry. Recipes contain ingredients, which are combined to produce a food item.
Pantry	This normally refers to a room in which food is stored. With respect to the Meal Helper application, the "pantry" refers to the component of the application in which the user enters his or her available ingredients.

Warnings and Hazards

Warnings

1. Risk of electrical shock. Do not break into phone wiring.

2. Biohazard. The phone may expose the user to microbes and other microscopic organisms. These organisms may cause disease. Wash your hands thoroughly before and after using the phone.

3. This phone contains a battery which may leak if used incorrectly. Follow manufacturer's instructions for care and charging instructions for the battery.

4. This phone contains class which may shatter and break. Do not apply too much pressure to the screen.

5. Undercooked food may cause sickness or food poisoning. Be careful to follow all recipe instructions regarding cook time and temperature.

6. Do not use phone as a projectile. Damage may occur.

7. Do not expose this phone to water. Such exposure may render the phone unusable.

8. Using this application while driving may result in serious injury or death

9. This program may recommend recipes that contain allergens. Consult with a doctor if you are unsure about any allergies you may have.

Disclaimers

1. Recipes provide by RecipePuppy.com and aggregated by Meal Helper.

2. This application requires a data connection, such as Wi-Fi or 3G network. Meal Helper will not function correctly without such connection.

3. The makers of this application are not responsible for incorrect information in the recipes or for improperly cooked or chilled food.

4. The makers of this application own the rights to this application, and do not intend to sell this application directly.

5. The makers of this application are not responsible for any allergic breakouts due to food allergies.

6. The writers of the application hold no responsibility for any health issues that may result from consuming undercooked or raw meat, spoiled ingredients, infected ingredients, or foods that may cause an allergic reaction. Likewise, the writers hold no responsibility for the improper use of any cooking utensils and/or apparatuses used in the preparation of food. However, the writers *will* claim responsibility for introducing any delicious, easy-to-make, inexpensive, and/or creative recipes to the user.

7. None of the recipes or other information gathered using Meal Helper represent professional health advice, and a medical professional should be consulted before consuming any edible items.

8. The information provided by Meal Helper may be updated periodically. Additionally, the information presented to the user may be out of date, and should be independently verified by the user.

9. The information contained in and presented by Meal Helper is provided “as is”, without any warranty expressly given or implied. Any damages to health or property having to do with the usage of this application are herein disclaimed and excluded.

10. Please abide by all local and state laws for food preparations.

Safety Precautions

Please prepare all meals to proper food temperatures:
Cook hot to minimum temperatures of:
- Poultry and Stuffing: 165º F
- Pork: 145º F
- Beef, Lamb and Seafood: 145º F
- Rare Beef: 130º F
- Hamburger (ground beef): 155º F

Serve hot foods above minimum temperature of 140º F.
Serve cold foods below maximum temperature of 41º F.

User Agreement

The recipes displayed in Recipe Helper are property and copyright of their respective owners. As such, Meal Helper cannot be held liable for any damages that result from the preparation of these recipes. To use this software, the user must agree not to hold Meal Helper responsible in any such scenario. Always remember to use proper safety precautions and take consideration of allergies when preparing food.

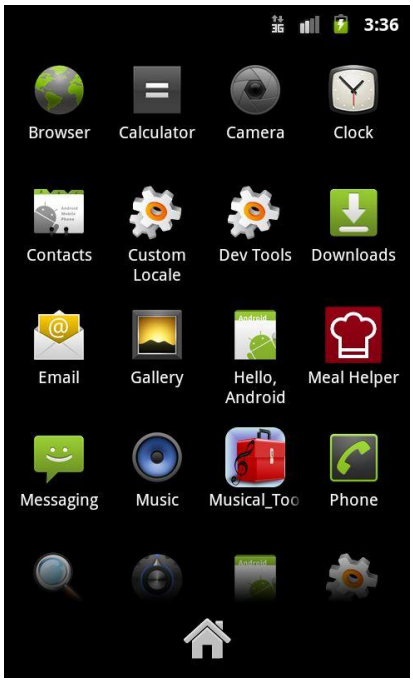
User Manual

Features

The application differs from other recipe applications by the "Pantry" feature. The ***pantry*** maintains a list of ***ingredients*** the user has on hand. After building the pantry, the user can search for recipes based on type of meal or dish the user would like to prepare. The application will display recipes found that can be made using only the ingredients on hand. The other feature available is the "Surprise Me" feature. After the pantry is built, upon selecting the "Surprise Me" feature, Meal Helper will display a random recipe based on the time of day; the user will then have the option to use the recipe or to have it select another recipe. Finally, the application stores a list of user-specified favorite recipes that can be accessed from the favorites menu.

Instructions

The first step to using Meal Helper is to install the application on an Android phone. The Android version of the phone should be 2.2 or 2.3. Open the application by tapping the application icon.



The application will then display the home screen:



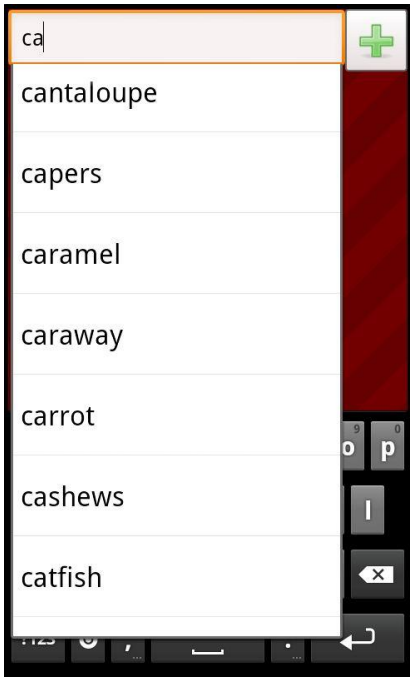
In order to give good search results for a recipe, the user first needs to enter ingredients into the pantry. Then, the user may either browse for recipes, or access the "Surprise Me" feature of the application. The user can view favorite recipes at any time. See below for instructions on how to use these features.

Build Pantry

From the application home screen, select “Edit Pantry”.



To add an ingredient, tap the green plus (+) icon at the top of the screen. This will pull up a keyboard with which the user can use to enter the ingredient. After typing 2 characters, suggestions will populate a ***drop-down list,*** from which the intended ingredient can be selected:



If the ingredient is not in the drop-down list, enter the full ingredient name using the Android keyboard. Tap the button again to add the ingredient. Repeat for each item available.



To remove an ingredient, press and hold (usually around 1-3 seconds) the item in the list until it has been removed.

After all ingredients are entered, hit the “Back” button on the phone to return to main menu.

Find Recipe

From the application home screen, select “Browse Recipes”. A list will display with the possible types of meals to choose from:

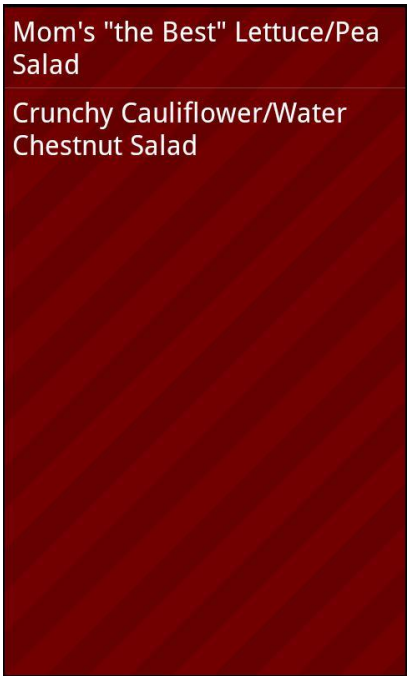


Press the settings icon to bring up the search settings menu:

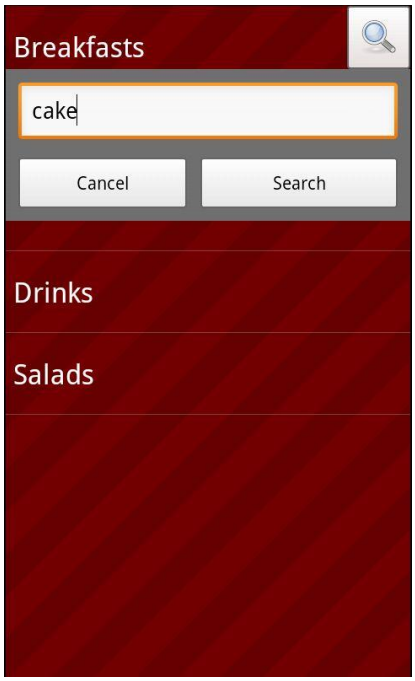


This menu will apply vegan, vegetarian, gluten-free, and kosher filters to the search query, which will force the application to display recipes that adhere to the checked boxes. When a box is checked, the check mark will turn green. The menu also allows the user to specify how many extra ingredients to include in the recipes. Including 5 extra ingredients will mean that recipes may contain up to 5 ingredients that are not included in the pantry. This number by default is 5.

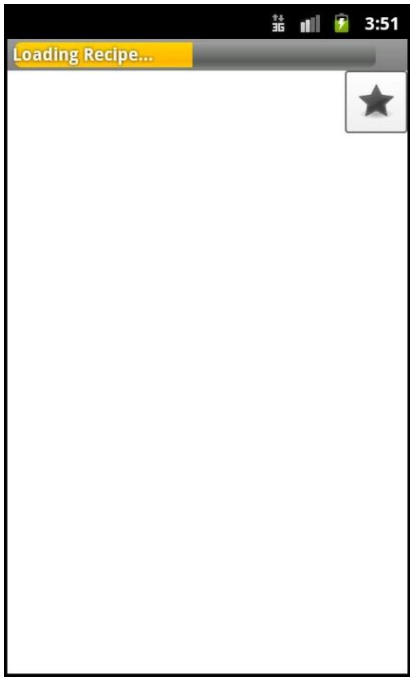
After specifying any search settings, select a type of meal. After selecting a type, a list of found recipes will display:



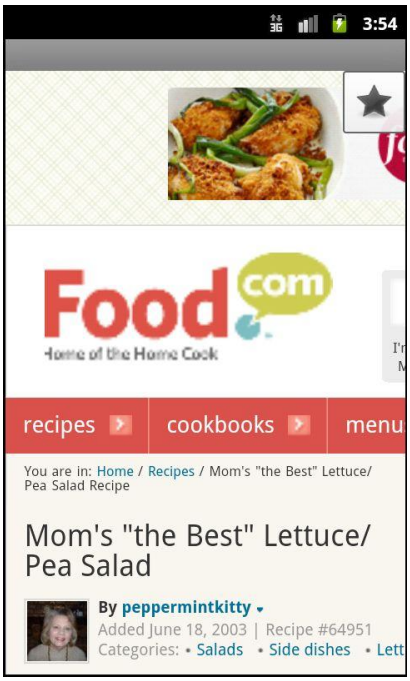
Alternatively, the user can tap the search icon to manually enter a search query for a specific type of recipe that uses the ingredients in the pantry.



Tap the desired recipe. The recipe will then begin to load. A status bar at the top of the screen will show the progress.



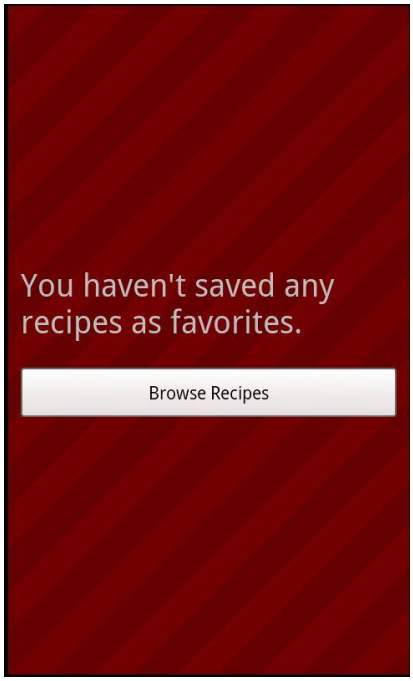
Once loaded, the user can zoom in or out and can scroll using the standard Android ***gestures***.



To save a recipe as a favorite, tap the star icon. If the star is yellow, this indicates that the recipe has already been saved as a favorite. If it is black, it is not a favorite.

Favorite Recipes

From the application home screen, select “Favorite Recipes”. If no recipes have been “starred”, a message will display, saying “You haven’t saved any recipes as favorites.



If recipes have been “starred”, they will be displayed here:



Select the desired recipe, and it will be displayed.

Surprise Me

From the application home screen, select “Surprise Me”. If the pantry is empty, or if the application can’t find any recipes, a message is displayed.



If the number of ingredients is sufficient, a message will be displayed saying “It looks like it’s time for_____.” The blank will vary depending on the time of day. If it is morning, then breakfast; if afternoon, then lunch; if evening, then dinner. The titl=e of the recipe it found based on the time of day is also displayed.



If the recipe sounds good, the user can select “That sounds good”, to display the recipe. If not, the user can select “No, something else please,” and an alternative recipe is displayed, provided there is one.

Appendix A: Technical Specifications

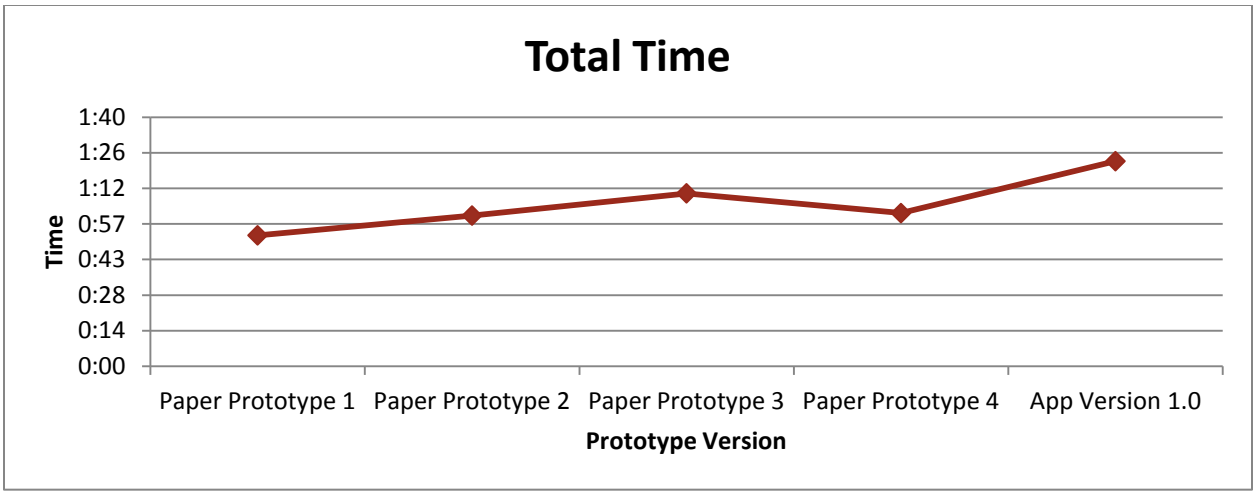
Meal Helper is built using the Android Software Development Kit(SDK) and written in Java. The application is built for Android version 2.2, but is compatible with Android versions up to 2.3.3. It incorporates a SQL database to keep a persistent pantry that is available even after reboots and application exits. Android is an open source project provided by Google. The Java programming language is an open source language provided by Oracle. The recipes used in Meal Helper are results of a search using an application programming interface(API) provided by recipepuppy.com.

Appendix B: Measurement of Prototypes and Final Product

The following table and graphs summarize the measurements of the project during various stages of development. The four paper prototypes indicated 4 different tests using our paper prototype. The App Version 0.1 was a skeleton program that did a few basic things, but could not pull up a recipe. It laid the foundation of App Version 1.0, which is the latest version of our application.

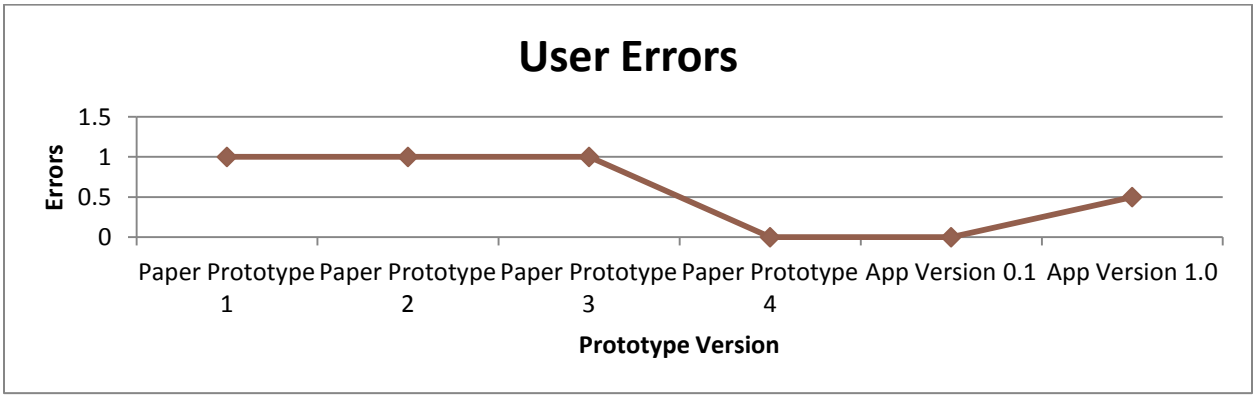
	Paper Prototype 1	Paper Prototype 2	Paper Prototype 3	Paper Prototype 4	App Version 0.1	App Version 1.0
Total Time	0:53	1:01	1:10	1:02	N/A	1:23
User Errors	1	1	1	0	0	0.5
Screen Changes	15	17	18	14	5	4
Application Size	0 KB (n/a)	0 KB (n/a)	0 KB (n/a)	0 KB (n/a)	12 KB	98 KB

The **total time** measurement represents the total time taken to accomplish our designated task. The task was to use our application to find a recipe for a salad. The obstacle in this task is that the user must use only ingredients in the user's panty. To do this, the user must enter all items currently in their pantry into the application, and then browse through the results returned by the application. The application will then display the recipe. The timer was started when the application is opened, and stopped when the recipe is fully displayed.



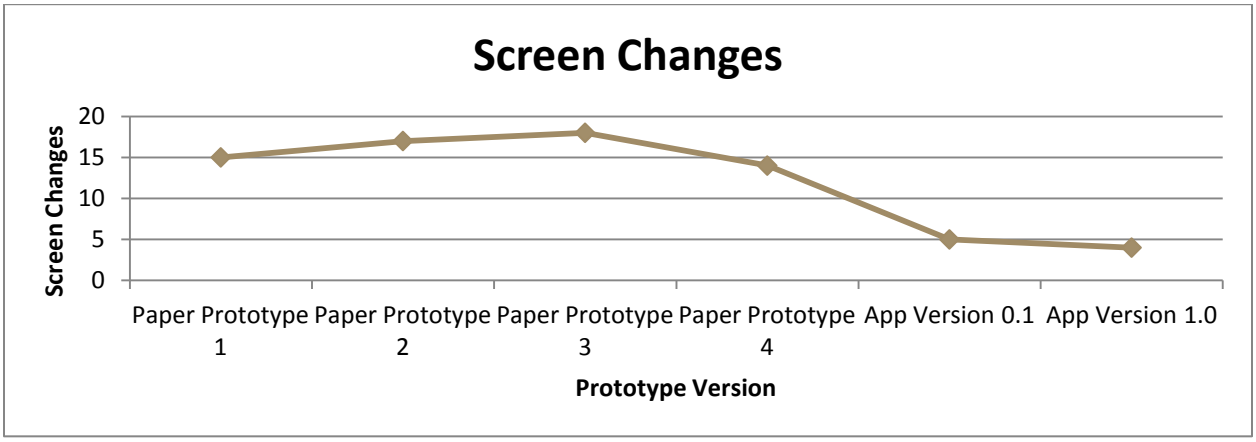
The time required was highest in our final version, though this was due to network latency while loading the recipe.

The **user errors** measurement represents how many times the user backtracks while accomplishing the task. In other words, if the user accidentally exits the application or touches a wrong button that will add time to the process, that is considered a user error.



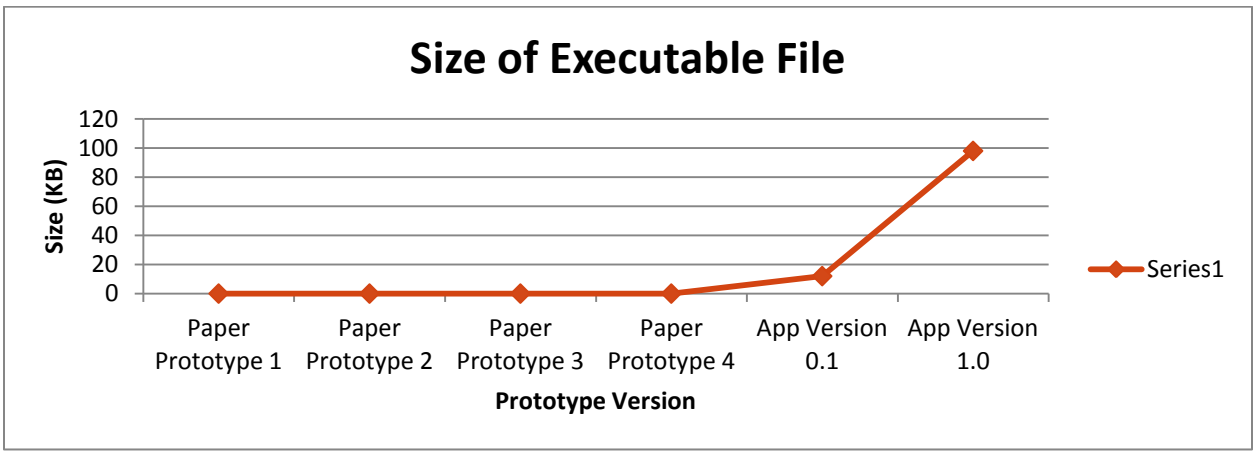
The 0.5 in the final version indicates that the user deviated from the ideal path, but did not have to backtrack to accomplish the task.

The **screen changes** measurement represents how many times the screen changed pages while the user accomplished the task.



The final version of the application kept a lot of activities located on the same page, which minimized the number of screen changes.

The **application size** measurement measured the size of the executable application. For android, this is a file ending in the .apk file extension.



The paper prototypes had no code associated, so this is why they have size 0.

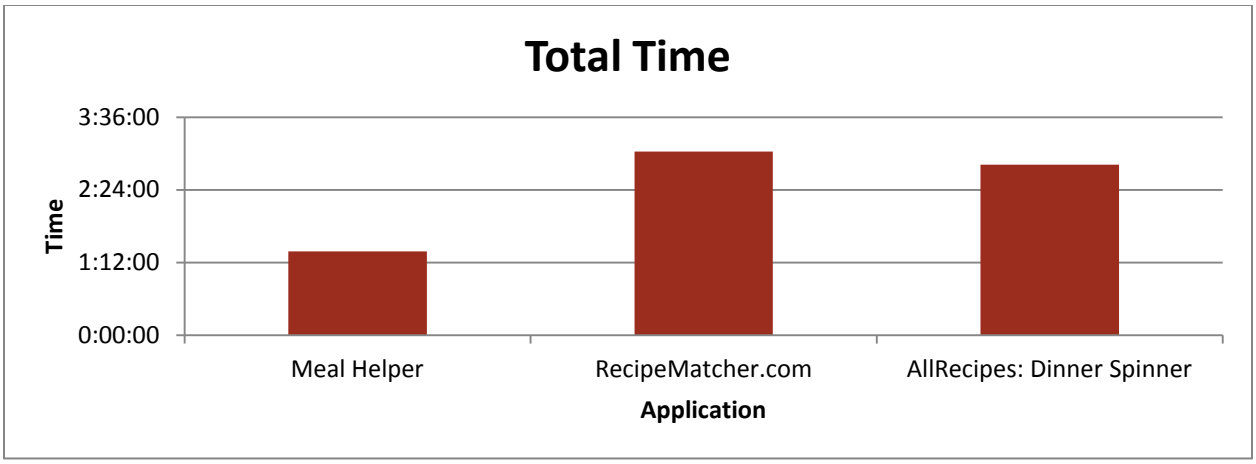
Appendix C: Measurements of Similar Products

There are a few similar products that provide the functionality of Meal Helper, but the only comparable applications do not optimize suggestions based on ingredients available. Due to this, any search for meals would take anywhere from seconds to days if the user cannot find one without extra required items. If the user decides to use one of these alternate products, they also have to deal with driving to the store, finding the item in question, and then returning home before even beginning to prepare the meal. For comparison, we are using a website that provides similar functionality as Meal Helper as well as the top rated recipe application on Google Play.

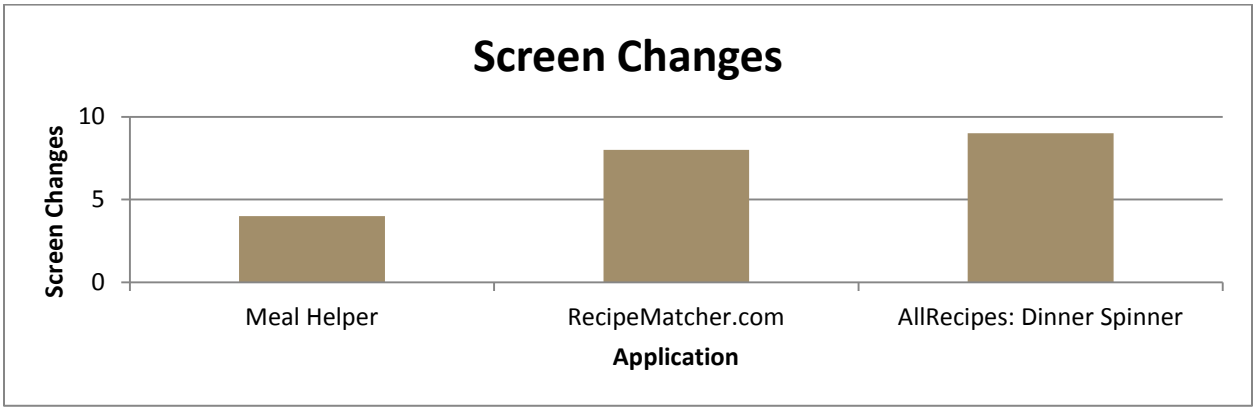
The following table and graphs summarize the measurements of the Meal Helper Application versus 2 competitors: RecipeMatcher.com and another Android application called AllRecipes:Dinner Spinner.

	Meal Helper	RecipeMatcher.com	AllRecipes: Dinner Spinner
Total Time	1:23:00	3:02:00	2:49:00
Screen Changes	4	8	9
User Errors	0.5	0	0
No. of Recipes	22	209	25
Unavailable Ingredients	0	2	6

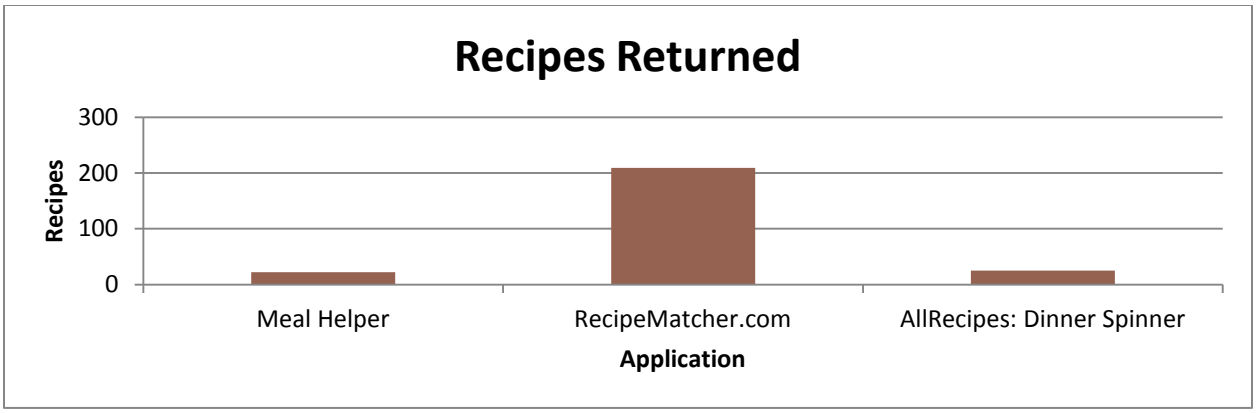
The measurements are the same as in the previous section. However, two new measurements have been introduced for the competition comparison. The **number of recipes** measures how many recipes the application could return based on the ingredients input. The **unavailable ingredients measurement** indicates how many recipes (on the first page) could not be made solely with ingredients in the pantry, thus would require extra store-bought ingredients.



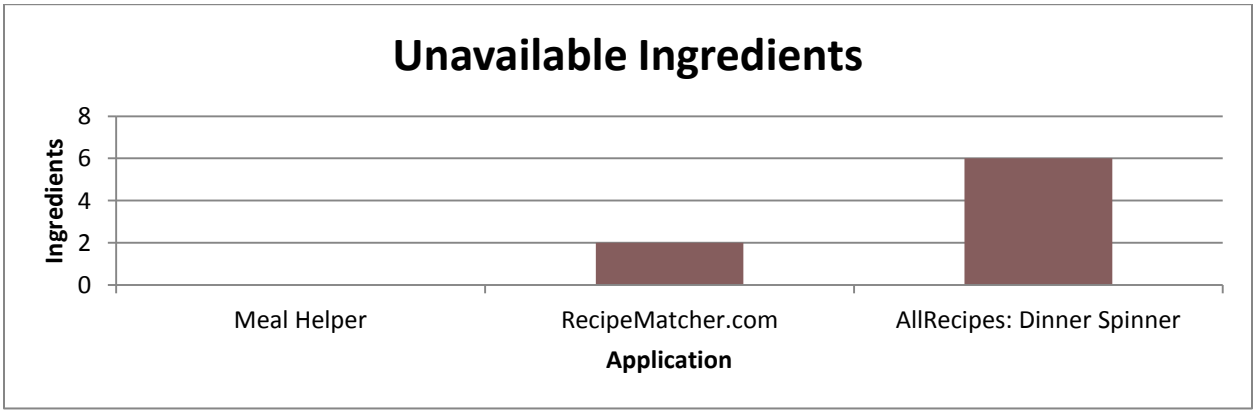
Meal Helper returned recipes in the fastest time.



Meal Helper required the fewest screen changes to accomplish the task.



RecipeMatcher.com returned the most recipes. It would be easy for Meal Helper to return this many or more recipes, but it would require a bit more implementation for this. Also, many of the recipes returned by RecipeMatcher.com include ingredients which are not in the user's pantry, and would require a trip to a grocery store.



Meal Helper was the only application that would return recipes that require no extra ingredients.

Appendix D: Design Decisions

Meal Helper was created by a group of university students for their design project. It was conceptualized when they were hungry, without a decent supply of food ingredients, and wishing they had an app for that. The choices the designers made, from the persistent pantry, to the surprise function, to the method of input for ingredients, each brought about the final product. The first critical design decision was what version of Android to target for the project. The choice came down to version 2.2 or 2.3.3; while more Android owners use version 2.3.3, our primary testing device ran Android 2.2. The second critical design decision was the method users would use to put ingredients into their pantry. While scanning the barcode is by far the easiest for the user, it is unreliable and difficult to build. It would likely have required using an outside database that is owned and operated by a company. Rather than deal with getting access permissions and toil over integrating it, we decided to use text-based entry with an autocomplete/suggestion dictionary to help alleviate spelling errors. The final critical design decision was the source of recipes and method of searching for makeable recipes. The decision on recipepuppy.com was made after searching for applicable application programming interfaces(APIs) found its unique search functionality. While there were more decisions made, these decisions made Meal Helper what it is and how it in itself is unique.