UCCS Bachelor of Innovation™
UNIVERSITY OF COLORADO COLORADO SPRINGS

Innovation Team

AJ Tuttle	Jesse Roberts	Mauricio Aquino	Seamus Downey	Grace Neumayer	William Kronfeld
INOV 3010	INOV 3010	INOV 4010	INOV 4010	INOV 3010	INOV 2010

Client

E³

Extravagant Wellness

• evaluate   • educate   • empower

Table of Contents

I. Cover Letter

Dear Mr. Amess,

Seamus and I would like to thank you and your team for the opportunity to work with you on this project. It was a valuable experience leading this team and working with you this semester. In this document, we provide some background on the Innovation program, discuss the work we have completed, go over a formal list of accomplishments, and provide recommendations for future development. We dedicate this document to you and your team, your continued success, and as a point of reference for the next developers to take over and continue this project. On the behalf of our team I would also like to thank you, Cheyenne, and Johnny for your dedication, support, and commitment to this project and our team. It was a pleasure working with each of you.

Thank you,

Mauricio Aquino

Seamus Downey

II. Introduction and Bachelor of Innovation Overview

The Bachelor of Innovation Degree at UCCS

The Bachelor of Innovation is an interdisciplinary bachelor's degree program unique to University of Colorado, Colorado Springs. The program consists of a set of innovation core classes that aim to teach students essential skills to prepare us for the workforce. In the bachelor of Innovation, we define innovation as the transformation of ideas into impact. The innovation core classes contain learning the process of entrepreneurship, gaining real-world teamwork experience, and mastering the practice of innovation. These classes work together to give each student current world experience and skills that will prepare them for the industry before graduation.

The Innovation Teams Courses

The BI consists of three iterations of innovation teams courses, each building on one another. There are three levels of teams courses: 2010, 3010, and 4010. This means students take this course at different stages of their education which gives them the opportunity to take this course as a sophomore, junior and senior. These courses connect a team of students, all from different years and majors all into one place where they can share ideas and collaborate. Senior level students are then assigned to clients that applied through the BI. These senior level students then interview the class and pick team members. Each group of students will work together to solve problems and innovate for a client. The first two levels act as members of the team, while the 4010 students take on the role of team leads. These courses offer students the opportunity to do real work with actual clients in the community. The advantages of this experience are countless. Students gain not only real work experience in a series of different industries, but they

also learn the art of teamwork and communication. These classes give students practice in skills that otherwise they may not have had the opportunity to learn. These skills can range anywhere from research and marketing to web design and app development. All of these unique skills and experiences make the Teams classes the heart of the BI program.

The Extravagant Wellness BI Team

Our team, Extravagant Wellness, was given the task of researching and developing a prototype phone application for Extravagant Wellness. This app will allow patients and clients to import medical test results, monitor nutrition requirements with meal tracking, browse recipes, and create shopping lists. This helps the client's customers make healthier choices and incorporate a longevity of wellness. The app will provide a centralized location for logging daily meals and drinks to keep track of the user's nutrient intake. It also will have a wide variety of health focused recipes which cater to specific patient needs. Our team consists of six UCCS BI students who were hand selected for their technical skills and knowledge in the software development field. Seamus Downey and Mauricio Aquino are 4010 computer security students who have taken on the roles of co-leaders. AJ Tuttle, Jesse Roberts, and Grace Neumayer are 3010 students studying game design, computer science, and computer security respectively. Additionally, our team took on a new member of the BI degree program, William Kronfeld, a 2010 student studying computer security. With the cumulative knowledge base and skill set of this handcraft team, we set ourselves up to not only meet, but exceed the Extravagant Wellness client's expectations.

The Value of the BI

The Bachelor of Innovation gives students the opportunity to develop skills that are useful and necessary in industry. The program not only teaches teamwork with peers, but the skills to work with clients as well. This skill is cultivated through experience, which the BI program provides in a secure environment which students can apply in future jobs in their career field. The program is built with the premise that innovation can be taught, and therefore used to create societal impact. Every student will leave school with the skills necessary to build or contribute to the society around them. The BI also gives students the opportunity to work on real life problems and projects. Learning about how to solve problems that have already been solved is one way of learning, but actually solving current real problems gives true experience and lessons that can only be achieved with hands on experience.

III. Executive Summary

The Bachelor of Innovation team was tasked to work alongside David Amess from Extravagant Wellness to research and develop a nutrition tracking mobile application for their customers. "The purpose of the project is a mobile application that will monitor nutrition values based on nutrition requirements, recipes, and shopping lists, to incorporate a longevity of wellness. The application will also allow the user to communicate directly with their doctor for proper monitoring of the patient's goals." The original proposal submitted by David Amess requested the research and development of a fully functional and deployment ready mobile application by the end of the allotted time frame. Knowing the resources and time requirements associated with developing a mobile application, the BI team pivoted and refined the scope to focus on the building of a prototype app that contains the core functionalities from the original design. Mr. Amess also provided two associates from the industry, Cheyenne McCoy, who was in charge of design and implementation, and Johnny Seay, who was an advisor for any technical questions or problems that members of the team may have had with the applications' overall development.

The BI team conducted their work through multiple platforms and environments. Coding of the front end of the application was done entirely through Android Studio in the Dart programming language with the Flutter Software Development Kit. This allowed the team to work and monitor the usability of the app in real time through an emulated mobile device. The server portion of the application responsible for the storage and retrieval of user data was programmed through a combination of Node.js and Python. The codebase was easily shared with all involved parties through GitHub, a cloud based storage and versioning service that allows multiple users to make alterations to code without conflicting with one another.

The team decided to distribute the technical workload by delegating sub teams and assigning specialized work based on preexisting skills and experiences. The teams created a series of different pages that would seamlessly work together to track a user's nutritional goals

and daily intake based on through a variety of different meal selection methods. The nutritional values for all foods tracked within the application were provided by the Food and Drug Administration and utilized throughout the application to show progress towards daily goals. The application takes the daily values and displays them for the user using percentages that allow the user to work at improving over health by tracking specific target nutrition goals.

Despite pivoting, the BI team was able to accomplish all the requirements that were laid out in the newly formed scope of work. Along with the mobile application, a market facing presentation was created to display the overall application design as well as the developed functionality of the application at the current time. The team was able to provide our client, Mr. Amess, with a working prototype application that matched the recommended design and specifications that were set in place. The results were monitored through weekly meetings with Mr. Amess, where members of the team would display and identify the struggles and accomplishments for each week. This allowed the BI team to remain on track and in sync with one another to successfully produce the desired deliverables.

IV. Statement of Accomplishments

The main goal of this project was to create a working prototype of the app laid out in the design document drafted by the client and his team. This process can be broken down into three phases that give a good idea of what our team accomplished during the semester. These stages are the planning phase, learning phase, and the development phase.

Phase I. Planning Stage

Our initial goal for the start of the semester focused on determining the details of the project and what our end deliverable would look like for the client. We had broken this process down into four main objectives within our scope of work document at the beginning of the semester.

Our initial objective was to get to know the client and his business to better understand the project. This consisted of several meetings between the clients and the leads as well as the client's team and our BI team. We were able to ask the client questions and have a discussion on the background of the project. From these meetings, we were able to gain a basic understanding of the client, his business, and his role within that business. We determined our client worked within a longevity clinic. This longevity clinic focuses on providing a unique series of medical tests for the client's customers and using this data to help the customer to make healthier choices. This is where our app came into play. We determined we would be creating an app that would eventually import this data, determine the customer's nutrient deficiencies, and track the meals they consume to help them meet goals to get to a healthier state of being.

Our next objective was to narrow down the scope of work in a way that would make the project feasible given the constraints placed on the project. This project was a massive undertaking due to the large scope of the app the client wanted our team to develop. The client already had an idea in mind of what he wanted the app to be like. This was laid out in a technical document provided to our team at the beginning of the semester. The result of this objective was the decision to provide a prototype of the app by the end of the semester instead of a fully functional and completed app. Our team decided to focus primarily on basic functionality as well as front facing design. The main constraints we identified within our scope of work were time, manpower and technical ability. These were all valid constraints and played a large role in the difficulties we encountered and what we were able to accomplish this semester. The largest obstacle here was that our team would be learning a brand-new programming language and had no prior app development experience. This required learning about the programming language we would be using and determining how to use and install the software to write with that language. Getting this software installed and working for all members of the team was the first accomplishment our team had made toward starting development. From here, we were able to move to our third objective.

The third initial objective was to plan out the development of the app from a high-level overview to develop an understanding of the design and implementation process. This was accomplished through a design document provided by the client and his team. His project lead, Cheyenne, provided us with an Adobe XD document that displayed the various screens of the app and some brief explanations as well. This design document provided further detail than the initial technical document that was provided by showing what the app would look like and how the screens would work together.

The final initial objective for our team was to begin assigning group members to roles and responsibilities in a way that would best fit the needs of the project. Once we reviewed the design document and discussed it with both teams, Seamus and I began to look at how to divide up the work for the semester. We ended up deciding to divide the group into three teams of two people to work on three different parts of the app at a time. These teams consisted of William and Seamus working on the login page and backend, Mauricio and Grace working on the add meal and drink pages, and AJ and Jesse working on the nutrition page. We stayed in these teams for most of the semester as we continued to develop these screens. This did not change till towards the end when we started to link the pieces of our app together which required team members working on different parts to meet and work on making those parts work together.

Achieving these four main objectives was the first major accomplishment our team made during the semester and boosted our morale. We finally had an idea of what we would be developing and how we would be accomplishing that. We also had tasks assigned to each team, so we knew what direction we were heading.

Phase II. Learning Phase

Throughout the development process we had several accomplishments that were not outlined in the Scope of Work due to the number of unknowns caused by the nature of the app development process. During this phase of the semester, we accomplished many learning goals of the project to begin development.

The first accomplishment that I would like to highlight is that our team managed to self-teach ourselves how to develop a mobile app with no prior experience. Many members of

our team had experience programming applications for computers, but never had developed a mobile app. This is something that is not taught at UCCS and that we had to learn ourselves. Learning app development entailed learning a new language called dart and the software infrastructure that houses that language called flutter. Many of our team members also had to learn how to design user interfaces, which is also not something that is taught within our university. This was a huge accomplishment. By the end of the semester, about half of the team has a strong understanding of the language and mobile development skills.

The next accomplishment I would like to highlight is our team had to learn how to use a version control software called git. Many members of our team had never worked with a version control software and so this was also a new learning experience. This was an important skill to learn as git is the industry standard for version control software and would help our team collaborate without the risk of issues that arise when collaborating on an app in a team environment. These issues include losing work, overwriting another teammates work, breaking a functionality that was previously working, etc. This software provided a way for us to keep our work separate, track changes to our code, and roll back to previous versions if necessary.

Another accomplishment our team members achieved is learning how to program within a team and group-based environment. Many members of our team were newer to programming and only had experience programming alone within a class environment. As many of us figured out, programming by yourself is entirely different from programming within a team of two and within a group of 6. While this has clear advantages in terms of performance and productivity, it also requires strong communication and teamwork skills. This allowed each of our members to grow and expand their networking skills within a team.

Phase III. Development Phase

The next phase our team entered was development of the app. At this point, we had learned the basics of the language and we began to reach some major accomplishments towards the end deliverable for the client.

The first development accomplishment for our team was the implementation of a machine learning software that could take an image of a food and determine what the food is from that picture. The success of this component was due to the hard work one of our team members, William put in to making this work. William became the backend technical specialist for the project and was essential for the success of this project.

The next development accomplishment was that our team created a server backend that would store and retrieve data for the app as well as house the machine learning functionality we had previously developed. We initially created a fake backend to meet the needs of passing data until we were able to arrange getting a server for the client to work off of. Once we had this server, we were able to create a fully functional and working server to work with our app.

The next major accomplishment was getting our individual functionality and components within our assigned screens working. This was a two-step accomplishment. The first was getting the functionality for our screen working as intended and the second was designing the user interface and making it look good. Getting functionality to work is one thing, but as many of us discovered, making it look good is an entirely other challenge. This was a challenge many of us were not familiar with. In some cases, we also added a third step which was adding backend functionality so that our app would work with the server we created.

The final accomplishment of app development was getting our individual screens and components working together to create a working app prototype. This required collaboration from different team members within the group as well as further backend integration with the server. Once we accomplished this goal, we were able to present a working prototype to the client which we refined during the last few weeks of the semester.

Phase IV. Presentation and Paperwork

During the final weeks of class, we have continued to refine the app, so it is presentable and so that it is ready to be passed off to a future team. We have also prepared a presentation for the client and his superiors, this final documentation, and a thank you video for the client as per the course requirements.

V. Recommendations and Innovation Discussion

Recommended Focus

The BI team recommends that Extravagant Wellness continues to develop this application due to its potential to revolutionize the health field. One potential route the client could pursue is requesting the help of a second team during the spring semester of 2021 in order to continue the development of the mobile application. As it currently stands, many of the foundational pages and feature sets required for the prototype app to stand on its own feet have been implemented; however, sub-pages such as "Plan", "Bio", and "Cookbook" as well as minor features such as page transitions still require development. The scope of the app is also ambitious and there are many additional unique features and functionalities that need to be implemented. If the client decides to apply for another BI team, our team also highly recommends that the succeeding team's scope be narrowed down to a few pages and feature sets to ensure their completion and avoid spreading resources too thin. The Extravagant Wellness app is a large and complex piece of software with many moving parts that could easily result in scope creep for teams if not defined concretely from the beginning. Future BI team applications should also include the request that at least one member has experience with backend and database systems due to their complexity and importance to the app's success. It is also recommended the client gets the team started on learning the skills necessary to develop the app as app development and the language and tools to create the app will have to be acquired as the team progresses the semester.

An alternative to applying for the BI team that the client may be interested in pursuing is hiring a professional developer or group. This may be the best option since many of the features

yet to be implemented may require or benefit from an advanced skill set. This may also save the client time and money in the long run as hiring another BI team would require that team to learn the skills and existing implemented work from scratch. Another potential route that the client could take is to hire previous members of the team which would require renegotiating terms outside the context of the Teams course on an individual basis. Whichever route the client decides to pursue we wish you success in future development of this application and contribution towards helping individuals making healthier choices in their lives.

Security Recommendations

Upon the current BI team's completion of work, Extravagant Wellness should take several steps to ensure the security of their software and systems. All BI team member's github accounts should be moved from a developer role to a read only role or be removed from the repository entirely. Extravagant Wellness should change passwords on any and all accounts created by or shared with the BI team. Alongside passwords, a new private key file should be generated for the AWS EC2 instance that hosts the backend server. This ensures that only those with the new private key can access the backend. Additionally, Extravagant Wellness should apply for their own USDA food database api key. The current api key used within the application is in the name of a BI team member and should be replaced.

Innovation Potential

The BI team strongly believes that the Extravagant Wellness app has the potential to bring positive and innovative change to the medical field. Due to the ongoing Covid-19 pandemic, new opportunities within the medical field continue to present themselves. The

demand for remote care is at an alltime high as hospitals are filled to capacity and people are

urged to remain in their homes. The Extravagant Wellness app's ability to dynamically track and

correct individuals' dietary deficiencies via proprietary testing pushes Extravagant Wellness into

a low competition sector of medical care with plenty of room to continue growth. Should

Extravagant Wellness continue to take advantage of these newfound opportunities and concrete

their position in the field, they will become a dominant force with a cardinal head start over

potential competitors.

VI. Conclusion and Discussion

Moving into the future, the BI team suggests that Extravagant Wellness continues momentum in development by focusing their resources around the medical test data upload, storage, and interpretation feature sets. The team believes this is an important focus point of development as it plays such a key role in the evaluation of user's nutritional deficiencies and recommended daily nutritional goals. The data extracted from the user's medical tests is prevalent to many of the pages within the application and could likely lead to a potential development bottleneck as other feature sets to be developed rely on it. Additionally, the team believes that the implementation of a "practitioner portal" is of particular importance. A practitioner portal is necessary to create the direct link between clients and practitioners.

We believe that the Extravagant Wellness mobile app will play a key role in helping your customers remain healthy. Our team believes that your application has the potential to revolutionize the health field with its unique and innovative qualities. These qualities include unique health tests, medical specialist coaching, image recognition for meal tracking, and a wide variety of healthy recipes. We wish you success in future development of this app, wherever it may take you. On behalf of the Innovation Program we would like to commend your team for setting such immense goals for this project and your unwavering commitment to the health field.

The BI team would like to personally thank Shawnee Starr and David Amess for providing us the opportunity to work with Extravagant Wellness. Our team has gained so much from working with you as our client this semester and we are truly grateful for this opportunity.

This experience has allowed each member of our team to learn to efficiently work within a professional environment, as well as improve our knowledge base in the technical field of app development. We are honored to have worked with you and we wish you continued success in your future endeavours.

VII. Supporting Materials

Scope of Work

<div align="center">

Extravagant Wellness App
Seamus Downey
Mauricio Aquino

</div>

Name of the project:
Extravagant Wellness Mobile App Research and Development Team

Name and contact information of the clients:
Shawnee Star [Extravagant Wellness CEO] <sstarr@techwise.com>
David Amess [Primary Client] - (719) 210 8169 <damess@techwise.com>
Cheyenne McCoy [Technical Director] <ember.rane@gmail.com>

Purpose of the project:
"The purpose of this project is to research and develop a prototype phone application that patients and clients can use to monitor their testing results, nutrition requirements, recipes, and shopping lists to incorporate a longevity of wellness."

Objectives and Success Criteria:
- **Objective 1:** Get to know the client and the project. Determine purpose and motivation behind client BI team application and request for mobile app. Understand the client's business, their customers, their purpose for what they do and as well as their goals for their company. Research meeting the needs of the customer's clients.
  - **Success Criteria:** All team members should have a strong understanding of the client and the project which will help throughout the app research and development process.
- **Objective 2:** Narrow down the scope to fit both the needs/wants of the client and the time and manpower constraints of the semester. Develop a realistic expectation for ourselves and for the client of what we should have done by the end of the semester. Determine what is the priority for the client and what they really hope to gain from our team this semester. This will require a deeper understanding of the client and project, which we will have acquired from objective 1.
  - **Success Criteria:** Have a mutual understanding between us and our client on what the finished product will look like.
- **Objective 3:** Plan out app development from a high level technical overview. This will involve no code, but it will help us determine design and implementation considerations based on the prior two objectives and how we can meet those. For example: What programming languages will we use? What platforms will we develop on? What potential

obstacles or issues could we run into? What technical resources will we need to complete the project? Will we need a database, a server, specialized software?

- ○ **Success Criteria:** As a team, we should understand the technical implications of developing our mobile app. We should know from a high-level technical perspective of how we will do what we need to do. We will know all resources and high-level implementation details.
- **Objective 4:** Begin assigning group members to roles and responsibilities. Start initial app development process. Have the project broken down into different modules from the previous objective. Keep in contact with clients throughout the completion of each module to ensure we are staying in line with what the client wants.
  - ○ **Success Criteria:** Have a successfully working prototype by the end of the semester.

High-level requirements - Deliverables:
- **Requirement 1:** A working prototype of the app's user interface and front end functionality. This will allow the user to interact with the device and display their nutritional/medical data to them.
- **Requirement 2:** A working prototype of the app's backend database system. This will have the functionality to store and retrieve user data when called by the front end.
- **Requirement 3:** A detailed technical manual that adequately describes the purpose, functionality, modifiability of the developed prototype's code.

Assumptions:
- **Assumption 1:** We will need to acquire access to a web server and or a database system to create a working prototype.
- **Assumption 2:** The client will need to work with the team to schedule meetings during times that work both for the client and the team.
- **Assumption 3:** The client must prioritize and modify their expectations according to the given constraints laid out below. The client must be reasonable in this regard.
- **Assumption 4:** The client must be present and interested in the project and the team and give meaningful feedback.

Constraints:
- **Constraint 1:** Time - The time given this semester to complete the project.
  - ○ Will there be enough time to complete the project? What will we be able to complete with the time given? How can we narrow our scope to fit within the time given?
- **Constraint 2:** Manpower - The amount of people we have on our team or that can assist with application development.

- Will we have enough people to complete the project? What can we accomplish with the people we have? Do we have enough people with the necessary skill sets to achieve our goal?
- **Constraint 3:** Technical Ability - Skills, knowledge, and technical expertise in required fields.
    - Does our team have the skills we need to complete our project and provide what the client is asking for? What does our team not know that they will need to know? Is there enough time to learn this throughout the semester?
    - Example: Working with new software and coding in a particular language outside of our familiarity. Working with databases and online unix servers.

High-level Project Description and Boundaries:

High-level Risks:
- **Risk 1:**
    - **Description:** There is potential that the developed mobile app and revamped website do not mesh as seamlessly as initially planned.
    - **Probability of Occurrence:** Low
    - **Impact on Project:** Medium
    - **Mitigation Actions:** Team leaders of both Extravagant Wellness projects shall actively speak and work together in order to maintain on the same page.
- **Risk 2:**
    - **Description:** Due to the nature of software development, scope creep is a large concern.
    - **Probability of Occurrence:** Medium
    - **Impact on Project:** Medium
    - **Mitigation Actions:** The BI team will set unambiguous and defined deliverables and success criteria that must be accomplished before any further additions can be made.
- **Risk 3:**
    - **Description:** There is potential that UCCS may bring the semester to an unexpected halt should a Covid-19 outbreak happen on campus.
    - **Probability of Occurrence:** Low
    - **Impact on Project:** High
    - **Mitigation Actions:** The most action that can be taken to mitigate this risk is to keep ourselves healthy and avoid entering situations that may be high risk for Covid-19 spread.

Summary Milestone Schedule:
- **Milestone 1:** The BI team is partitioned into sub-groups to focus on specific aspects of development.
  - **Description:** Once the entire team has met the client and discussed finer details, members of the team will be assigned to specific regions of the project that play into their individual strengths.
  - **Date:** 9/15/2020
- **Milestone 2:** The BI team conducts meaningful research into their specified regions of the project.
  - **Description:** Each BI team member will be responsible for furthering not only their own knowledge, but the knowledge of the client so that all involved parties have a higher chance of success.
  - **Date:** 9/22/2020
- **Milestone 3:** The BI team begins the iterative design process for the Extravagant Wellness mobile application
  - **Description:** The BI team members begin prototyping and designing the mobile application. The client will be kept in close touch for approval and alterations of the prototype.
  - **Date:** 9/29/2020
- **Milestone 4:** The BI team finishes and delivers the working mobile application prototype to the client.
  - **Description:** Upon finishing the working prototype, the prototype will be delivered to the client so that any last minute alterations can be requested.
  - **Date:** 11/24/2020
- **Milestone 5:** The BI team makes any last minute alterations to the mobile application by request of the client.
  - **Description:** Should any last minute alterations be necessary, the BI team shall work to make the necessary amendments.
  - **Date:** 11/26/2020
- **Milestone 6:** The BI team delivers the final version of the mobile application prototype to the client.
  - **Description:** The BI team will deliver the final version of the mobile application prototype once the remaining alterations and amendments have been made.
  - **Date:** 12/1/2020

Stakeholder List:
- **Stakeholder 1:** David Amess & Shawnee Star
  - **Relationship:** Co-founder and co-owner of Extravagant Wellness.
- **Stakeholder 2:** Our BI Team

- ○ **Relationship:** Actively assisting with Extravagant Wellnesses Innovational needs.
- ● **Stakeholder 3:** The BI Program
    - ○ **Relationship:** Referred to program for client application.
- ● **Stakeholder 4:** UCHealth
    - ○ **Relationship:** A potential business partnership between UCHealth and Extravagant Wellness is in the works.

Signatures and Emails:

Seamus Downey - sdowney2@uccs.edu

Mauricio Aquino - maquino@uccs.edu

William Kronfeld - wkronfel@uccs.edu

Jesse Roberts - jroberts@uccs.edu

AJ Tuttle - atuttle@uccs.edu

Grace Neumayer - gneumaye@uccs.edu

Extravagant Wellness App Development B.I. Team

About Us

Our team consists of six members studying under the Bachelors of Innovation degree program at the University of Colorado Colorado Springs. The BI program provides our team with important real world skill sets such as the use and application of intersectional innovation, entrepreneurship, communication, and business strategies. All of our team members are currently studying varying skews of computer science and have a passion for innovation. We believe that through this degree and the experience gained working with real world clients, that we will be set for success when entering industry.

Meet our Team

Seamus Downey
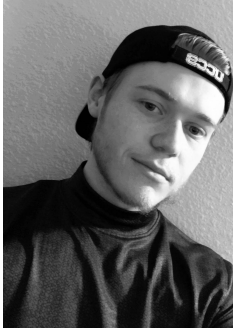
Email: sdowney2@uccs.edu - seamusdowney22@gmail.com
Phone Number: (408) 881-4309
Degree and Major: Bachelors of Innovation in Computer Security
Teams Position: 4010 Lead (Senior Level Status)
Experience in C, C++, Java, Javascript, and Python programming languages. Basic knowledge in website and UI design. Strong conflict resolution skills.

Mauricio Aquino

Email: maquino@uccs.edu - aquinomo@gmail.com
Phone Number: (719) 660-7035
Degree and Major: Bachelors of Innovation in Computer Security
Teams Position: 4010 Lead (Senior Level Status)
Programming Experience: Fluent in Python and C++. Familiar with C, cLisp, ML (Meta Language), and Java.
Other Experience: Unix & Bash Scripting, 3+ years in the grocery industry.

AJ Tuttle

Email: atuttle@uccs.edu - ajtuttle99@gmail.com
Degree and Major: Bachelors of Innovation in Game Design & Development
Teams Position: 3010 (Junior Level Status)
Programming Experience: Experience in C#.
Other Experience: Familiar with Github & experienced working with virtual teams

William Kronfeld

Email: wkronfel@uccs.edu - wkronfeld123@gmail.com
Degree and Major: Bachelors of Innovation in Computer Security
Teams Position: 2010 (Sophomore Level Status)
Programming Experience: C++, C#, C, Objective C, .Net core 3.1, .Net core 4.8, Python 2.7,
Python 3, Javascript, NodeJS, x86 assembly, 8086 assembly, Bash scripting, Visual Basic, Java,
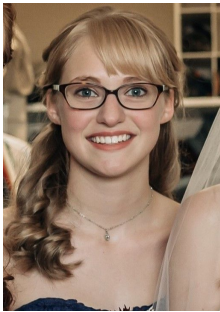PHP, TypeScript, ES6

Jesse Roberts

Email: jroberts@uccs.edu - Jesroberts101@gmail.com
Degree and Major: Bachelors of Innovation in Computer Science
Teams Position: 3010 (Junior Level Status)
Programming Experience: Java

Grace Neumayer

Degree and Major: Bachelors of Innovation in Computer Security
Email: gneumaye@uccs.edu - graceneumayer01@gmail.com
Teams Position: 3010 (Junior Level Status)
Programming Experience: C++, C.

Meeting Records

Client Meetings:
9/15/2020 - 6:30pm - 7pm
9/22/2020 - 6:30pm - 7:30pm
9/29/2020 - 6:30pm - 7:30pm
10/06/2020 - 6:30pm - 7:30pm
10/13/2020 - 6:30pm - 7:20pm
10/20/2020 - 6:30pm - 7:20pm
10/27/2020 - 6:30pm - 7:10pm
11/03/2020 - 6:30pm - 7:10pm
11/10/2020 - 6:30pm - 7pm
11/17/2020 - 6:30pm - 7pm
12/01/2020 - 6:30pm - 7:30pm

Technical Manual

Front End

The latest version of the front end code currently exists on the "showcase" github branch

Login and Registration

login_screen.dart - Responsible for the building of the login screen of the app that is presented upon startup

    Important Classes:
        _LoginScreenState - Contains all widget building functions responsible for the
            login screen's layout and functionality

    Important Functions:
        _buildLoginBtn - Upon the login button being pressed, _buildLoginBtn calls
            handleUserData with the user's username and password credentials in the
            attempt to log the user into their account

    Important Info:
        The loginUrl string "http://54.70.209.230/login" will need to be changed if the
            servers ip or login path is updated.

register_screen.dart - Responsible for the building of the account registration screen which users sign up through

    Important Classes:
        _RegisterScreenState - Contains all widget building functions responsible for the register screen's layout and functionality

    Important Functions:
        _buildRegisterBtn - Upon the register button being pressed, _buildRegisterBtn calls handleUserData with the user's email, username, and password credentials in the attempt to register a new user account

    Important Info:
        The registerUrl string "http://54.70.209.230/register" will need to be changed if the server's ip or register path is updated.

user_auth.dart - Responsible for authenticating a user's data upon login or account registration

    Important Functions:
        handleUserData - Handles post requests to the server for both login and account registration. Upon response from the server, handleUserData returns the response's Json body to the function caller

        validatePassword - Responsible for ensuring password length and complexity requirements are met for any new accounts attempting to be registered

    Important Info:
        handleUserData dictates whether to make a login post or registration post based on whether the passed in url parameter contains the keyword "login" or "register". If the urls are updated, the keywords must be updated as well.

setup_screen.dart - Responsible for building the setup screen that collects user's personal info for account setup

    Important Classes:
        _SetupScreenState - Contains all widget building functions responsible for the setup screen's layout and functionality

    Important Functions:
        _postUserInfo - Makes post requests to the server with the new user's biometric data so that it can be stored in the database

Important Info:

        The post request made in the _postUserInfo function currently only stores the users biometric data that is entered at the bottom of the setup screen. This function can be later expanded to upload the allergens/preferences and medical test documents.

        The setupUrl string "http://54.70.209.230/setup" will need to be changed if the server's ip or setup path is updated.

setup_search.dart - Responsible for creating the search functionality of both the dietary allergens and dietary restrictions section of the setup screen

Important Info:

        The dietaryRestrictions and dietaryAllergens lists can be easily edited and expanded to include more options for users to choose from. However, the backend server would be required to track which recipes abide by the newly added restrictions/allergens.

login_constants.dart - Is a constants file that contains many of the design settings and parameters for all pages related to login/registration

Important Info:

        The constants file increases design modularity by allowing a setting to be changed in a single file and having it take effect over multiple widgets and pages. Be weary when making edits to the contents of the constants file as it could result in unintended changes elsewhere within the app's design.

Meal Tracking

For the add meal functionality of the app, there are several files that make this component work these are:

- addMeallHomeWidgets.dart
  - This file implements the homescreen widgets for adding a meal and a drink.
  - There are two main Widgets in this file _initMeal and _initDrink that allow users to:
    - View their current progress for daily nutrients through the use of a progress bar
    - Add a meal or drink with the click of a button
    - View previously added meals or drinks and their details using a scrollable list of buttons

- addMealScreeen.dart
  - This file implements the main screen for adding a meal and the functionality to take a picture or select one from your gallery.
    - Utilizes a library to implement this functionality
  - This screen presents the user with four options to add a meal. These are upload a photo from the gallery, take a picture, search the USDA food database, or search favorited recipes through the users personal cookbook.
  - One important function in this file is _pickImage function which takes in an image source and uses the library to acquire that image, then passes it to the cropScreen class.
    - Image source is gallery or camera depending on what user selects.

- callNutritionApi.dart
  - This screen implements the USDA food database. It calls the database, searches it, returns results, and parses that data into multiple classes.

- cropScreen.dart
  - This file implements the crop screen page.
  - This screen is presented after a user has selected an image and utilizes a library to crop the image.
  - The screen presents the image and then the option to confirm the image, edit it with the cropping library or reset the image to clear it.
  - There are two important functions within this file which are _cropImage and _clear. These each implement their corresponding functionalities.

- mealConfig.dart
  - This file contains all constants/globals for colors, sizes, widgets, etc.

- recipeView.dart
  - This file implements the recipe view page.

- The recipe view screen contains a placeholder for an image, displays the name of the selected food, shows the user the ingredients that make up the meal and presents the nutrients for the food.
- There are a variety of important functions in this file that provide various functionality for the screen
  - The favorite function adds a meal to the current list of meals in the user's personal cookbook
  - The unfavorite function removes the meal from the user's personal cookbook.
  - The function, isFavorite, determines if the currently viewed meal exists in the user's personal cookbook.
  - The _removeFood function allows the user to remove the currently viewed food from their currently added meals for the day.
  - The _reportFood function is a placeholder for future functionality to report an inaccuracy in a meal.

- searchCookbook.dart
  - This file creates a search delegate object for the my cookbook search.
  - This search gets suggestions and results from the user's personal cookbook.
  - The important functions for this file are:
    - passData which takes the selected food and portion size in the user's current meals for the day.
    - _choosePortionSize which allows the user to select from a list of predefined portion sizes after selecting a meal.
  - The search delegate class allows us to have a functional search within the app.
  - There are four methods that are defined to control a search delegate object.
    - buildActions defines the button on the right of the search and controls what it does
    - buildLeading defines the widget to the left of the search
    - buildResults is called whenever the user selects a suggestion or hits the search key after typing a query. It displays results that match the query or selected suggestion.
    - buildSuggestions is called whenever the user updates the search query and displays a list of suggestions that contain the query
  - buildActions and buildResults are basically null since they are overwritten by the searchDelegateWrapper.dart and not actually used within this app. However, they do work if you choose to utilize them.

- searchDatabase.dart
  - This file creates a search delegate object for the USDA food database search.
  - This file searches by utilizing the callNutritionApi to search the USDA food database.
  - There are a few functions in this file that provide important functionality for the app
    - getApiData uses the callNutritionApi.dart to get results based on the query entered by the user

- ■ passData which takes the selected food and portion size in the user's current meals for the day.
- ■ _choosePortionSize which allows the user to select from a list of predefined portion sizes after selecting a meal.
  - ○ There are four methods that are defined to control a search delegate object.
    - ■ buildActions defines the button on the right of the search and controls what it does
    - ■ buildLeading defines the widget to the left of the search
    - ■ buildResults is called whenever the user selects a suggestion or hits the search key after typing a query. It displays results that match the query or selected suggestion.
    - ■ buildSuggestions is called whenever the user updates the search query and displays a list of suggestions that contain the query

- searchDelegateCallback.dart
  - ○ This file talks to the search delegate classes and passes the data to the wrapper. This is the middle man between the search delegate classes and the wrapper.
  - ○ The main function in this file is setCallbacks which is what links the suggestions and the results from the search classes to the wrapper.

- searchDelegateWrapper.dart
  - ○ This wrapper allows the search delegate object to be used within a container widget.

- uploadScreen.dart
  - ○ This implements the upload picture page.
  - ○ Implements functionality to upload the image to the server and retrieve the result to the server.
  - ○ There are three main functions within this file which are
    - ■ _showDialog which prompts the user to accept or deny the machine learning result for the image they uploaded to the server
    - ■ _showError which captures and errors from the connection to the machine learning and displays them to the user
    - ■ _upload which connects to the server, uploads the image, and receives the response back for what the food is.

User Profile

user_profile.dart- Responsible for the building & updating of the user's profile page

Important Classes:

Bubble- Contains all of the moveable bubble widget's data

_BubbleState- Contains the progress bar's colors, as well function for building the template of all draggable bubbles

_NutrientsState- Contains all widget building functions responsible for the user profile's layout & functionality

Important Functions:

initState- Everytime the main menu is reloaded (which, in turn, every time the user profile is reloaded), the grid of bubbles is made & nutritional data is assigned & updated

updateNutritionalData- Using the nutritional information from the passed in food, updates the corresponding nutritional bubbles to display the new changes in nutritional value

Important info:

When implementing file read & write, make sure to check if the file exists in initState; if it doesn't, run the code that creates myButtons & save the file after. If the file does exist, set myBubbles equal to the saved list. There should also be code that saves the file in the updateNutritionalData function, as well as the DragAndDropGridView's onReorder method.

Daily nutrients are currently hard-coded & there is no logic to fetch the user's actual daily nutrients.

The user's total nutritional value is calculated but not displayed anywhere

If adding more nutrient bubbles, the height of the container that encloses the DragAndDropGridView will need to be increased

Because of line 353, the bottom right bubble will never fill up. Rather than checking k's value, check if myButtons[k]'s text is not + (this entire check can also be removed in replacing the + button)

DragAndDropGridView is not a base part of flutter, so if documentation is needed please see here: https://pub.dev/packages/drag_and_drop_gridview

Backend

The Extravagant Wellness BI team will not be including any server passwords within this document for security purposes. Passwords will instead be delivered in a separate document only to be shared with the client.

Backend files (located in the ~/backend directory on the amazon ec2 instance):

For detailed documentation on API parameters, there is a packet structure documentation page located at http://54.70.209.230/docs

|- backend
  |- config
    |-index.js - Main configuration file, contains database ip, database port, database username, database password, database name, socket url, socket port, logging level, and API port.
  |- index.js - Responsible for loading files and starting the server. To run it simply use node Index.js or start it on a screen session.
  |- logs
    |- debug.log - A log file if the configuration file is set to debug
    |- info.log - A log file if the configuration file is set to info
    |- errors.log - A log file if the configuration file is set to error
  |- node_modules - All the dependency files for the backend
  |- package-lock.json - A lock file for listed dependencies
  |- package.json - A list of dependencies and other things Node.js uses
  |- public - A directory the holds all uploaded images
  |- src
    |- routes
      |- login.js - The login handler
      |- main.js - Main page, simply outputs {"hello":"World"}
      |- recog.js - Image recognition handling system
      |- register-get.js - A simple registration page for testing purposes
      |- register-post.js - Registration system for the API
      |- setup.js - Account setup system
      |- test.js - PDF text extraction testing
  |- server.js - The main API web server that loads the routes from the routes directory
  |- util
    |-crypto.js - Responsible for handling encryption
    |-db.js - Responsible for handling database interactions
    |-logger.js - Responsible for logging messages to the console and files

The only file that needs to be changed is located in the configuration folder config named index.js

VIII. Appendix A - Weekly Standup Documentation

10/6/2020 Standup Schedule

Updates/Progress:

Mauricio Aquino

- Focused on intercommunication between team members
- Working to ensure all team members are comfortable in their environment
- Clarified general questions/concerns with Cheyenne
- Learning Git and Flutter through online resources

Seamus Downey

- Learning Git and Flutter through online resources
- Developed simple layout for home page to use as template
- Began implementing standup structure to client meetings

Grace Neumayer

- Installed Adobe XD, Git, Flutter, and Android Studio
- Learning Git and Flutter through online resources

William Kronfeld

- Making progress on backend framework
- Trained an MLS to recognize different foods based on pictures
- Set up simple sample API
- Waiting on further details on data to be stored in database

AJ Tuttle

- All required tools downloaded and installed
- Following online flutter tutorials
- Working on the "write your first flutter app" listed on flutter's website

Jesse Roberts

- Installed Adobe XD, Git, Flutter, and Android Studio

- Learning Git and Flutter through online resources

- Yet to receive an invite to github repo (jesroberts101)

10/13/2020 Standup Schedule

Updates/Progress:

Mauricio Aquino & Grace Neumayer (Meal page & Drink page):

·   Converted meal page's adobe XD wireframe into code

·   Added buttons to the meal page (Still need functionality)

·   Currently working on search bar functionality and implementation

AJ Tuttle & Jesse Roberts (Nutrition page):

·   In progress of converting the nutrition page's adobe XD wireframe to code

·   Working on dragging functionality to drag down page from top of screen

·   Successfully created movable icons for the nutrition circles

Seamus Downey & William Kronfeld (Login page & Backend):

·   Built the sign in screen for the app (design is work in progress since no XD file to reference)

·   Interfaced login screen with backend system (working on register now)

·   Encrypted login traffic between front and back end

·   Set up backend communication for food image recognition

<div align="center">

10/20/2020 Standup Schedule

</div>

Updates/Progress:

Seamus Downey & William Kronfeld (Login page & Backend):

·   Expanding on login screen by adding functionality to the buttons

·   Created a basic registration screen much like the login screen

·   Working on password requirements/restrictions (length, complexity, allowable characters)

·   Researching and experimenting with new ML models and datasets for multiple object recognition

Mauricio Aquino & Grace Neumayer (Meal page & Drink page):

·   Researched into how to take, store, and upload pictures of food for meal/drink page

·   Started experimenting with implementation of picture taking/uploading functionality

·   Cleaned up existing code for ease of reading and working

·   Converted design aspects to variables for easy change in the future

AJ Tuttle & Jesse Roberts (Nutrition page):

·   Continuing to convert the User Profile wireframe into a working page

·   Successfully got the dragging function of widgets to work

·   Beginning research on the implementation of dart.io to save user preferences to local device

10/27/2020 Standup Schedule

Updates/Progress:

Mauricio Aquino & Grace Neumayer (Meal page & Drink page):

· Finished research and implementation of the image functionality.

· Images can be uploaded/taken/cropped from the user's device.

· Created a standardized layout for the constant page and how it will be utilized.

· Currently researching shadow effects of buttons and widgets to improve design.

AJ Tuttle & Jesse Roberts (Nutrition page):

· Finished converting the Nutrition XD wireframe into a page.

· Fixed drag and drop functionality to swap bubbles rather than shift them.

· Will start research on persistent storage so that user preferences can be stored and loaded at runtime

Seamus Downey & William Kronfeld (Login page & Backend):

· Small design changes on login/register pages to reduce "blockiness".

· Added password requirement validation (length & complexity).

· Fixed some small bugs in frontend preventing proper registration.

· Added session data so that users don't have to login every time the app is started.

· Changed app name and logo to Extravagant Wellness on device.

· Created the login route of backend so that registered accounts can be referenced.

· Added backend validation to prevent users from registering/logging-in with nonexistent or partial data.

· Created an interactive documentation page in the backend.

IX. Appendix B - Showcase Presentation Slides

Listed below are the PowerPoint slides associated with the showcase demo conducted by the BI team on 12/01/2020

Account Setup Screen ✓

Users are directed to account setup upon first login

Users upload their GI, NutrEval, and SpectraCell test files upon account creation*

Users can search for and chose all food related allergens and dietary preferences to refine recommendations

Users enter their general health information and have the ability to setup a smart device for live monitoring
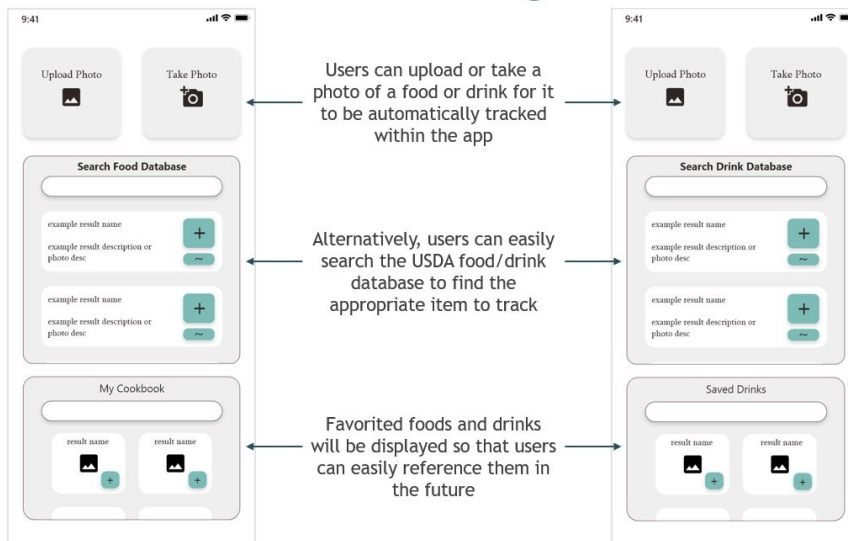
Home Screen ✓

Home Screen acts as central hub for many other screens

Bottom of the user profile screen is visible from the home screen to display key nutrition tracking information

Users can include their daily meal and drink intake to easily track their nutritional goals
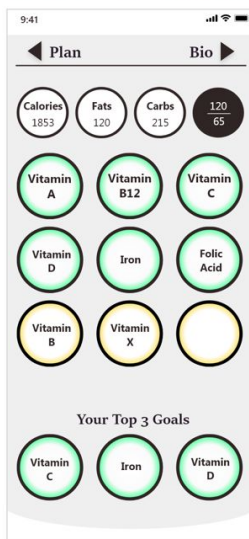
Top of the cook book screen is visible from the home screen to display all dietary preferences being taken into account

Meal✓ and Drink* Tracking Screens

Users can upload or take a photo of a food or drink for it to be automatically tracked within the app

Alternatively, users can easily search the USDA food/drink database to find the appropriate item to track

Favorited foods and drinks will be displayed so that users can easily reference them in the future

User Profile Screen✓

Responsible for all nutrition deficiency, goal, and intake tracking

Macronutrient tracking keeps count of total daily calorie, fat, protein, and carbohydrate intake

Micronutrient tracking keeps count of daily intake of assorted micronutrients

Border color shifts from red to green as users come closer to hitting their daily intake goals

User can move any micronutrients into their "Top 3" goals section to always be visible from the home screen
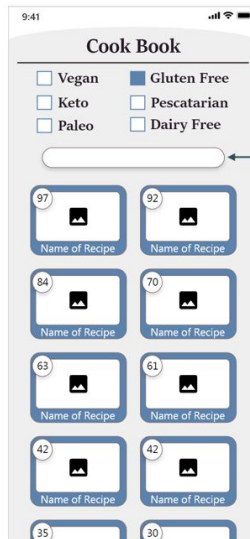
Bio Screen*

Bio screen is responsible for the tracking of all user biometric information

9:41

◄ Plan   Back ►

Biometrics

| age | weight |
| height | BMI |

(user name) ⟳

Device Name
sync date

Data 1:
Data from Garmin App
displayed as it is in garmin app

Data 2:
Data from FitBit App
displayed as it is in Fitbit app

Users can view and edit their basic health information that was set at account creation

Users can view their real time health data collected from their synced smart devices and associated apps

Cook Book Screen*

Cook Book screen houses Longevity Wellness' preapproved, health focused recipe list

9:41

Cook Book

☐ Vegan   ☑ Gluten Free
☐ Keto   ☐ Pescatarian
☐ Paleo   ☐ Dairy Free

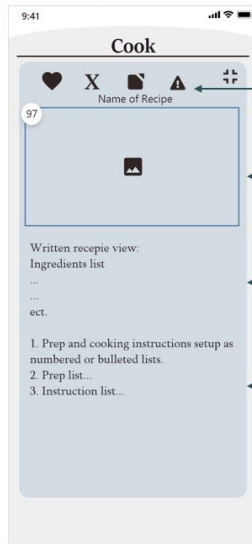| 97 🖼 Name of Recipe | 92 🖼 Name of Recipe |
84 🖼 Name of Recipe	70 🖼 Name of Recipe
63 🖼 Name of Recipe	61 🖼 Name of Recipe
42 🖼 Name of Recipe	42 🖼 Name of Recipe
35	30

User's dietary restrictions are set upon account setup and can be changed at any time

User can narrow the cook book's returned recipes by searching for keywords

List view of all Longevity Wellness preapproved recipes that promote a healthy diet

Upon selecting a preapproved recipe, the respective recipe screen will be presented for users to follow

Recipe Screen ✓

Recipe view allows users to view ingredient/nutrient breakdowns as well as recreate meals

9:41

Cook

♥ X 🗒 ⚠ ⤢

97

Name of Recipe

Users have the option to favorite, remove, copy, or report a recipe

Image of selected food will appear here

Written recepie view:
Ingredients list
...
...
ect.

1. Prep and cooking instructions setup as numbered or bulleted lists.
2. Prep list...
3. Instruction list...

Entire ingredients list and nutritional breakdown of the food is displayed to the user

Prep and cooking instructions for the selected recipe is displayed for users to follow along*

Prototype App Development Progress Demo

Development timeframe: Sept 2020 – Dec 2020

Demo conducted through GIF images for record keeping purposes and ease showcase via virtual meeting
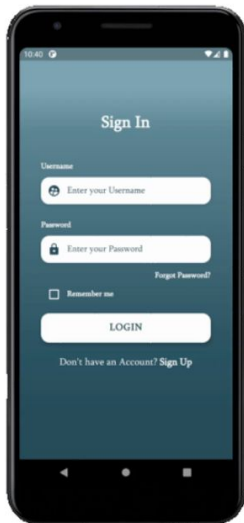
User Data and Server Security

User credentials and medical data is stored on a HIPPA compliant AWS cloud server

Data transmitted between the user's local device and the AWS server is encrypted via SSL (Secure Sockets Layer)

User passwords are ran through a secondary layer of encryption using Bcrypt and are only stored in a hashed format

User's login sessions are given a unique 512 character string to ensure session security

User Registration and Login

Sign In

Username

Enter your Username

Password

Enter your Password

Forgot Password?

☐ Remember me

LOGIN

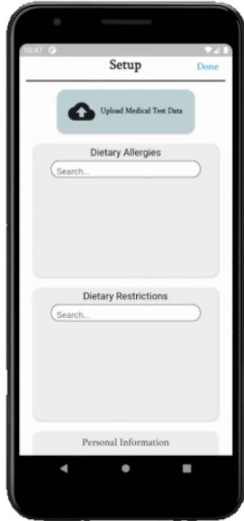Don't have an Account? **Sign Up**

Straight forward account registration process

Unique accounts tied to user's email and username

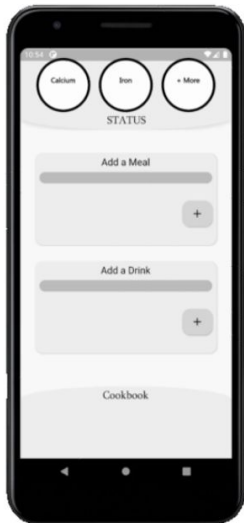Enforced password length and complexity for security

Initial Account Setup

Account setup required at first login

Allergen list used to refine recipe recommendations to user

Dietary preference list used to further refine recipe recommendations

General biometrics used to track user's starting point and progress over time

Tracking a User's Meals

Meals can be added to the meal tracker via image recognition or manual search
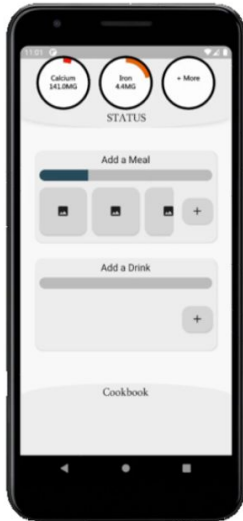
Image recognition is currently limited to small subset of foods

Manual search can identify any food registered with the USDA

Meals added to the meal tracker are immediately calculated into user's daily intake

"My Cookbook" stores any meal/recipe saved by the user for easy future access

Tracking a User's Daily Intake

Nutrition screen is dragged down from the top of the home screen

Keeps track of user's daily caloric, macronutrient, and micronutrient intake

Each micronutrient has a progress indicator to show progress towards reaching the user's recommended daily intake

Progress indicators shift from red to green as users come closer to reaching their recommended intake

Progress indicators shift back towards red as users begin to exceed their recommended intake

Questions & Feedback