# Ingredible

## CS701 Final Project Report

Chloe Johnson
Middlebury College
Middlebury, VT
cejohnson@middlebury.edu

Kayla Nahi
Middlebury College
Middlebury, VT
knahi@middlebury.edu

## ABSTRACT

The two main goals of our computer science 701 senior seminar project, Ingredible, were 1) to develop a feature-forward mobile application that provided users with ingredient-to-recipe search functionality and 2) to carefully design the user-interface in an effort to cultivate the user-experience. This project was motivated by three factors: food waste at the consumer level, budget-conscious behaviors, and a lack of existing ingredient-to-recipe mobile applications. We identified U.S. college students as the primary target audience for our final product. Developing our application on the Apple iOS Xcode platform, we used recipes from Allrecipes, CookingLight, Epicurious, Food Network, and Food 52 to stock our Firebase cloud-based database. The success of Ingredible was assessed in a beta-test phase, which yielded both quantitative and qualitative feedback. The final result of this project was a fully-functional, feature-filled, elegantly-designed, and user-approved ingredient-to-recipe mobile application. Our 11 college-aged beta-testers, on average, rated the application's main functionality as "very easy" or "easy" to use via a 5-point Likert-type scale. Our beta-testers also contributed many positive comments about the numerous features in our the application. Many of them reported liking the meal type and dietary restriction filtering on the ingredient form, customizable pantries, and the ability to favorite and unfavorite a recipe. Overall, the final product achieved both primary project goals, delivering on usability and functionality. Areas for future work include supplementing our database to accommodate thousands of available online recipes, the implementation of editable pantries, continued beta-testing, and submission of Ingredible to the AppStore.

## Keywords

CS701; LaTeX; mobile application, ingredients, recipes, food waste, budget, college, vegan, vegetarian, United States

## 1. INTRODUCTION

Up to half of all the food that is produced globally each year is wasted.[6] In the United States alone, almost 60 million tons of food is thrown away annually, 40% of which is associated with solely retail and consumer levels. The environmental and economic impact of this food waste is enormous. This U.S. consumer-level, wasteful behavior does not stem from a lack of conscience or irresponsible buying behaviors, as evidenced by the almost 80% of the U.S. population who experience feelings of guilt when throwing away food.[9] In reality, many people, especially college students, frequently demonstrate concern about their finances and food shopping habits.[2][7] Given this data, the issue of consumer food waste in the U.S. appears to derive from the consumer's inability to maximize use of their bought produce, their ingredients. The mobile application, Ingredible, targets the U.S. college-aged consumer cohort and attempts to address the issues of food waste and expenses by providing users with the ability to find recipes from their existing ingredients. The app aims to be both waste-conscious and budget-friendly as users can create recipes from soon-to-expire ingredients,

thus maximize usage of current ingredients without requiring another trip to the grocery store.

The design and implementation of Ingredible fits within the mobile development subfield of computer science and gives careful consideration to the user experience (UX). Our novel contribution to this field and problem is the creation of an aesthetically pleasing ingredient-to-recipe mobile application. No mobile application has successfully implemented this idea. The few apps that have made strides toward this functionality have either failed to provide a true ingredient-to-recipe search or buried this function in an unusable UX.

The remainder of this paper will provide (1) a more detailed discussion of the problem and motivation for this project, (2) pre-existing work related to Ingredible, (3) a description of the methods and techniques used to develop Ingredible, (4) beta testing feedback and app design results, (5) a discussion of these results, and (6) an exploration of possible avenues for future work.

## 2. PROBLEM STATEMENT

The issue of food waste is a global phenomenon. In 2013, it was estimated that "30%-50% [1.2 to 2 billion tons] of all food produced on the planet is lost before reaching a human stomach".[6] The global greenhouse emissions resulting from the entire food production process for wasted food totals about 3.3 billion metric tons of CO2.[12] Additionally, as shown in Figure 1, the higher the country's standard of living, the more food is wasted at the consumer level.[8][3] It is for this reason that we chose to center our efforts on the United States, despite food waste being a global issue.

The United States alone is responsible for more than 60 million tons and $160 billion in food waste each year and approximately 40% of that food comes just from the retail and consumer levels. [10] [1][3] This is roughly twice the amount that the U.S. Department of Education received from the Federal Budget in 2017. Despite this high degree of waste, an estimated 77% of people in the US feel guilty about throwing away food. [9]

We recognize that attempting to tackle the issue of global food waste is not a reasonable target for the scope of this project. Instead, Ingredible focuses specifically on the U.S. college student demographic, with the potential for scaling to other subsets of the U.S. and global population. Our interest in and selection of this subsection is primarily because the application will also address this population's desire to save money. A national survey conducted by Ohio State University reveals that seven out of ten U.S. college students feel stressed about their personal finances.[7] A study at the Wisconsin HOPE Lab, affiliated with the University of Wisconsin at Madison, found that 71% of college students have changed their eating or food shopping habits due to a lack of funds.[2] This is particular cohort could greatly benefit an application that provides recipes from their pre-existing ingredients, negating waste and the need for further expenditures.

The decision to implement ingredient-to-recipe matching functionality in a mobile application hinged on a series of assumptions about the benefits of a mobile app. First, mobile applications provide a much faster alternative to web browsing. Web browsing requires a user to launch a web browser, enter a URL and wait for the site to load, whereas it takes only a few seconds to launch a mobile application. Mobile apps additionally increase customer engagement by expanding the level of accessibility. The portability of a mobile device is also an applicable advantage for the functionality of our app. Further, as mobile phones are less expensive than computers, we are able to reach a broader audience by providing a cost effective alternative to accessing web applications via a PC. With nearly 77% of all Americans owning a smartphone, the potential user base is enormous.[11]

## 3. RELATED WORK

The web applications SuperCook and FOODWISE implement similar functionalities to our mobile application. Users provide their ingredients and then are presented with a list of recipes from popular recipe websites that match their in-

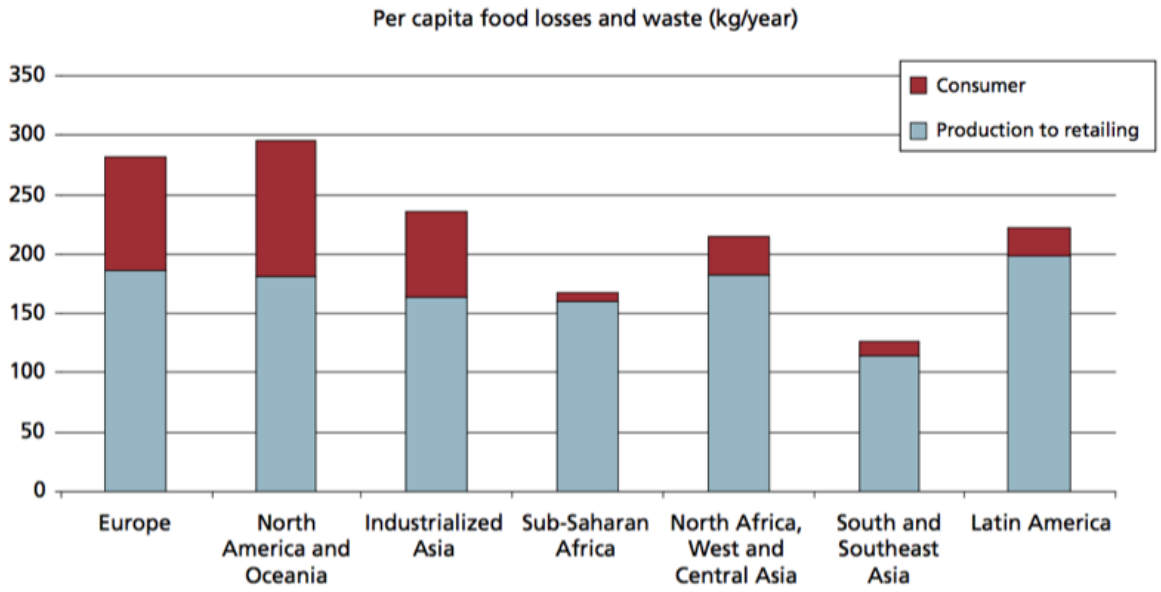


**Figure 1: Per capita food losses and waste, at consumption and pre-consumptions stages, in different regions.**

put ingredients. SuperCook and FOODWISE each provide a different mechanism for users to select their ingredients. SuperCook allows users to type-and-search for ingredients as well as locate ingredients through food categories (e.g. dairy, vegetables, etc.). The items within each food category are non-alphabetized. FOODWISE asks users to select the first letter of the ingredient they are searching for, and then select their ingredient from an alphabetized list. Both SuperCook and FOODWISE allow users to select dietary restrictions, but only SuperCook gives the user the option to select a meal type (e.g. breakfast, dinner, etc.). SuperCook also allows users the ability to "heart" their favorite recipes. Of the two, SuperCook delivers a better UX and more comprehensive functionality.

However, no mobile application provides the user with elegantly designed ingredient-to-recipe functionality. Mobile apps like BigOven, CookPad, and EpiCurious are recipe-related and have achieved good ratings and many positive reviews, but do not offer an ingredient-to-recipe service. SuperCook, on the other hand, has attempted a mobile version of its successful recipe generating webpage, but has received horrendous reviews on the App Store and an overall rating of 1.5/5.0 stars. AppStore reviews for the aforementioned mobile applications highlight many shortcomings: (1) the user can only search by one ingredient, (2) the ingredient search is not well-organized, (3a) the user is forced to register to use the application, (3b), there are issues when logging into the application, (4) the application crashes, (5) the user interface is jumbled, and (6) the app does not provide true ingredient-to-recipe functionality.

Our application has improved upon the negative critiques of SuperCook's mobile application and implemented many of the positive aspects of the successful web applications. Ingredible was built to provide a clean and usable mobile UX. The app design is easy to navigate and crash-free. Ingredible does not require user registration. It allows the user to filter recipes for meal types and dietary restrictions. The user can select as many ingredients as they wish from alphabetized lists categorized by food type – combining the best elements from SuperCook's and FOODWISE's ingredient selection. The user can also favorite and unfavorite recipes. The biggest, novel innovation in our application is the "pantries" functionality. The user can choose from a preset series of ingredient lists called "pantries" (e.g. "Dining

Hall", "Kitchen Classics", etc.), create custom pantries, and delete existing pantries. This negates the repetitive work of a user finding and selecting their standard kitchen pantry items (e.g. salt, pepper, milk, eggs, bread, etc.) each time they conduct a recipe search.

# 4. METHODS

## 4.1 Xcode

Given our initial focus on HCI and creating a feature-forward, user-friendly mobile application, we began our design process by sketching a visual representation of our idea. We chose to paper prototype instead of building a wireframe because of the expedited timeline. While both methods provide the app more clarity and functionality, paper prototyping is a faster and more cost effective method. Once completed, we began implementing our concept in Xcode, an integrated development environment for macOS apps, using the Swift programming language. We chose Swift over Objective C, another possible language built into the Xcode interface, because it is a safe and interactive programming language that provides powerful type interface and pattern matching, allowing complex ideas to be expressed clearly and concisely. With Swift, memory allocation is handled automatically and built-in error handling provides controlled recovery from unexpected failures. Additionally, as the front end of our app was a priority, we required the shortened build times Swift allows to constantly review our interface progress. Finally, as one individual in our developing team was new to app development, Swift was additionally appropriate as it is fairly intuitive and friendly to new programmers.

## 4.2 Data

The backend data was generated by collecting a series of recipes to serve as our database. This set of recipes was carefully curated with the key demographic of college students in mind, sourcing from popular recipe websites like AllRecipes, Food52, Epicurious, Cooking Light, and FoodNetwork. Intentionally choosing recipes with smaller ingredient lists and commonly accessible ingredients helped us accomplish this goal. The majority of recipes included in the database do not have an extensive time requirement or heavily complicated procedure. Additionally, each selected recipe has mostly positive reviews on the website through which we accessed it.

## 4.3 Firebase

Within the first few weeks of Xcode development, it became apparent that local storage would not be sufficient to store our recipes. Switching to Firebase, a cloud based database, allowed for faster run times, smaller space requirement, and enormous scalability. Firebase is a mobile and web application development platform run by Google that provides functionality like databases, analytics, and crash reporting. We transferred all Ingredible recipes to the Firebase NoSQL cloud database, storing them as JSON files. This database is synced to all users in real-time and remains available when the app is offline.

## 4.4 Beta-testing

At the beginning of development, we made the choice to work in a combined iterative and incremental style, maintaining a visible product at every step. It was because of this decision that we were able to offer a beta test soon after implementing Firebase. It was crucial to be able to conduct and successfully execute a beta testing phase in our application development, as one of our main goals throughout this project was to provide a clean, accessible user interface and an enhanced user experience. We offered a beta version to a small set of testers before presenting our preliminary progress in order to better understand how to shape the final deliverables. The beta test process mimicked many of the strategies for beta testing outlined by Stavova, Dedkova, Ukrop, and Matyas (2018) and Amazon's "Best Practices for User Acceptance Testing and Beta Testing".[13] When

deciding who should make up the body of testers, we considered only representatives of our target user group: U.S. college students. All 11 of our beta testers are current Middlebury College students or recent college graduates. This was an important consideration for us because the changes made as a result of beta test feedback needed to generalize to our expected user-base. Additionally, the deliberate selection of beta testers guards against potential failure or user dissatisfaction when the app is launched. This is especially true when the beta test pool is small, like ours: "the fewer testers a company has, the pickier it should be about their selection".[13]

Once selected, we met with each beta tester individually, clearly explained the beta testing process, and defined the tasks, goals, and response expectations.[13] Each beta tester was asked to perform three to five of the following tasks in a random order: find a recipe, favorite a recipe, make a pantry, unfavorite a recipe, and read more about Ingredible. After each task was completed, the beta tester was asked to assess how difficult the task was on a 5-point Likert-type scale, with 1 corresponding to "very easy" and 5 representing "very difficult". The beta tester was also given the opportunity to provide a response to the following two questions: "What did you like about this process?" and "Would you make any changes to this process?". Once the beta tester completed all three to five tasks, they were asked the following questions: "What did you like most about the mobile application?", "Would you recommend this app to a friend, yes or no?", "What is your least favorite part about this application?", and "What barriers, if any, exist that would prevent you from using this application again in the future?".

The benefits from beta testing like this are numerous. On the pre-release end, having beta testers means finding - and resolving - potential problems, getting a sense for the features and interface elements that the users like and dislike, and learning how the users interact with the app. Testing early in development also gave us the opportunity to implement tester ideas. Using beta testers also creates the potential for word-of-mouth marketing following the app's launch, which may increase profits and bolster the size of the user-base.[4] The decision to include beta testing in the development of our application was an obvious one given the countless research and practice indicating how it improves reliability, performance, and profitability, as well as our demonstrated commitment to enhance the user experience.

### 4.5 Repository

*https://github.com/knahi/Ingredible*

## 5. RESULTS

### 5.1 Design

Our application is presently fully functional, feature-filled, and employs a clean design. The application launch screen takes the user to the homepage from which they can navigate to any of the other main tabs ("Pantries", "Favorites", or "About") or begin their recipe search via the "Start Cooking" button. The recipe search allows the user to filter their results by meal type ("All", "Breakfast", "Lunch", "Dinner", "Dessert", or "Snack") or dietary restrictions ("Vegetarian" or "Vegan"). The user can select a pre-existing pantry of ingredients and/or select ingredients from a table sorted by food category. Selected ingredients will appear below the ingredient table. The user can find recipes containing their selected ingredients via the "Get Results" button. If the search yields any recipes, then the user can click on one to see the details of the recipe. If the user likes the recipe, they can "favorite" it by clicking the star in the upper-right corner. The user can navigate back to the recipe results via the "Back" button in the upper-left corner. If the user wants to view existing pantries or create their own custom pantry, they will click on the "Pantries" tab on the bottom tab bar. From there, they can select an entry in the pantries table to see a detailed view of all of the ingredients contained in this pantry. The user can return to the table of pantries via

the "Back" button in the upper-left corner. They can also delete an existing pantry from this list by sliding left on the table entry. The user can add a new pantry by selecting the "+" button in the upper-right corner. The user can title their new pantry and add as many ingredients from the ingredient table as they wish. Clicking on the "Add Pantry" button will save the pantry. If the user navigates back to the pantries table via the "Back" button in the upper-left corner, then they will see their new pantry of ingredients listed in the table of pantries. If the user wishes to see their list of favorited recipes, then they can simply click on the "Favorites" tab on the bottom tab bar. By clicking on one of the recipes here, the user can see the details for the selected recipe and "unfavorite" the recipe by clicking the star in the upper-right corner. Selecting the "Back" button in the upper-left corner will return the user to the previous screen. Finally, the user can select the "About" tab on the bottom tab bar to see more information about the application. The storyboard of this design is displayed in Figure 2.

## 5.2 Beta test feedback

Our beta test consisted of one quantitative and one qualitative portion.

### *5.2.1 Quantitative*

Each of the 11 beta testers were asked to complete three to five of the following tasks in a random order: find a recipe, favorite a recipe, make a pantry, unfavorite a recipe, and read more about Ingredible. After each task was completed, the beta tester was asked to rate the task's difficulty on a scale from 1 to 5. A score of 1 corresponded to "very easy", 2 to "easy", 3 to "moderate", 4 to "difficult", and 5 to "very difficult". The average ratings for these quantitative tasks are presented below.

Find a recipe: 1.67

Favorite a recipe: 1

Make a pantry: 1.25

Unfavorite a recipe: 1

Read more about Ingredible: 1

These results indicate that all of the tasks in our application are, on average, very easy or easy to identify and accomplish. As such, we did not spend considerable time making changes to the design-flow of the application following the beta test phase.

### *5.2.2 Qualitative*

Each beta tester was also given the opportunity to provide us a response to the following two questions for each of the predefined tasks: "What did you like about this process?" and "Would you make any changes to this process?". Following the completion of the three to five tasks, each beta tester was also permitted to answer the following questions: "What did you like most about the app?", "Would you recommend this app to a friend, yes or no?", "What is your least favorite part about this application?", and "What barriers, if any, exist that would prevent you from using this application again in the future?"

The qualitative results were overwhelmingly positive. Beta testers used phrases like "super intuitive", "user-friendly", and "accessible" to describe the ease of accomplishing the assigned tasks. They also demonstrated a strong liking for the features in our application. Five of the beta testers commented on how much they liked being able to customize the ingredient search through meal type selection and filtering. Eight of the beta testers spoke specifically about enjoying the option to "favorite" and "unfavorite" a recipe. Six beta testers talked about the ability to customize and use pantries. This particular subset of testers was very enthusiastic about this feature. "I love the pantries; I'm on the pantry train – full steam ahead", said one Ingredible beta tester. The suggestions we received during the beta testing phase were very helpful in achieving our application usability goals. Given our early timeline for beta testing, we were able to implement many tester ideas following the completion of this process. For example, two of the beta testers demonstrated issues with the scrollability on our ingredient

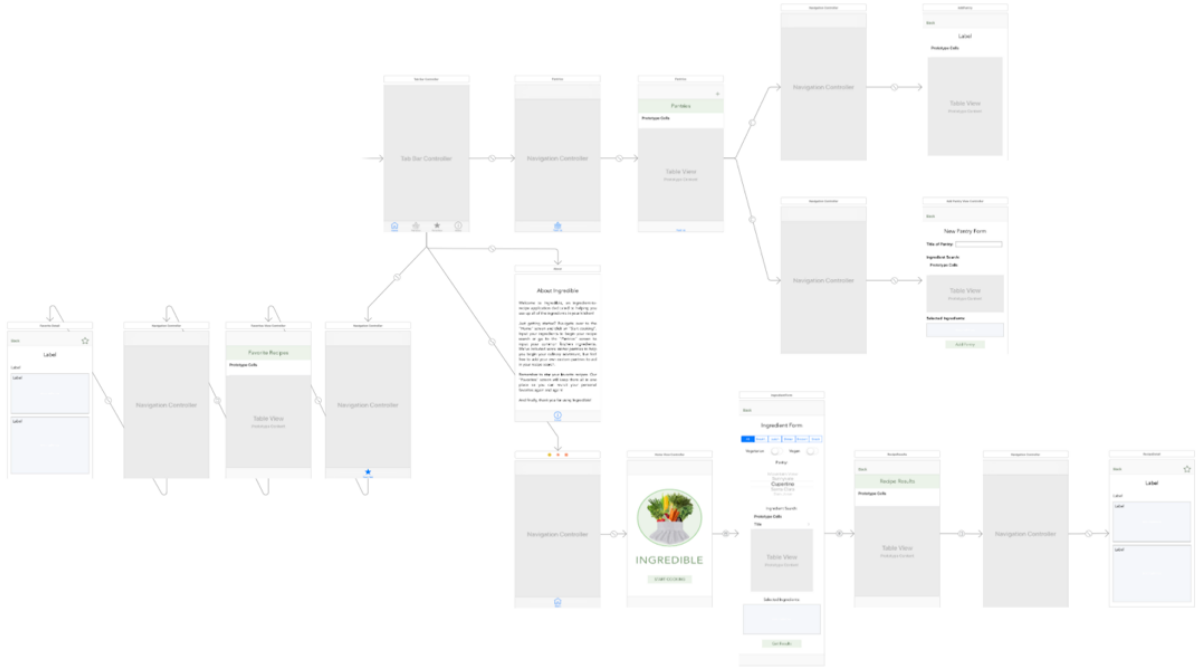


**Figure 2: The Xcode storyboard for Ingredible.**

form. As a result of this feedback, we were able to redesign the form to allow more room for scrolling on the sides of the form for both right-handed and left-handed users. Another beta tester thought that it would be helpful to see an alert pop-up when you favorite a recipe. We were also able to implement this type of alert system. Changes like these were instrumental in accomplishing our goal of providing the user with a clean, easy-to-use application.

We were unable to resolve two of the tester suggestions by the end of the semester: increasing our number of recipes and implementing editable pantries. These tasks required more time to execute than we had available to us, but are discussed below as possible arenas for future work. No beta testers specified any notable barriers in our application that would prevent them from using Ingredible again in the future. All beta testers indicated that they would recommend this application to a friend, some even asking us to provide them with a download of our product as is.

## 6. DISCUSSION

Our beta-testing results confirmed that our project goals of functionality and a cultivated user experience were met. The quantitative results reveal that the main functionality of our application was executable with no prompting on our behalf. The qualitative results provided a broader overview of the usability and user-experience. Users described the application as being highly accessible and they provided predominantly positive feedback regarding the features in the application. Ultimately, throughout the mobile application development process our application improved at a much quicker pace through human responses in the beta-test phase than through our reliance on and implementation of HCI principles and theory. We learned the most about the user's interests when testing it on the users themselves, rather than predicting what they might like using HCI guidelines like affordance, consistency, constraints, feedback, mapping, and visibility.[5] These concepts abstract concepts proved too broad to successfully implement without user feedback. Perhaps the main take-away from this project is that Ingredible's features and design are well-liked by the college-aged demographic, which means that the issues of food waste and budget-consciousness have the potential to be success-

fully addressed in our target demographic via the application. We are extremely happy with the user experience and feature-forward design of our mobile application. Ingredible was well-received and provided all of the functionality that we sought to accomplish. However, by concentrating our time and focus on developing features, like favorites, filtering recipe search, and customizable pantries, are recipe database suffered. Although this decision was entirely intentional, as we did not want to spend the majority of our time manually inputting recipes or cleaning up recipe scraper results, the clear downside is that our recipe database is relatively sparse. The project would be improved by having more recipes in our Firebase database, as indicated by beta-test feedback. This could be accomplished by building an effective scraper to pull recipes from popular recipe websites into a JSON format or crowdsourcing JSON recipes from sites like Amazon Mechanical Turk. As one beta tester suggested, we could also continue to improve the pantries feature by implementing an option for users to edit their existing pantries, rather than deleting and restarting. Furthermore, we could continue to beta test our project and eventually put our application on the AppStore.

## 7. ACKNOWLEDGMENTS

We would like to acknowledge Professor Grant for his support of our initial idea and suggestions and continued help along the road of implementation.

## 8. REFERENCES

[1] S. Goldenberg. Half of all us food produce is thrown away, new research suggests, 2016. [Online; accessed 15-May-2018].

[2] S. Goldrick-Rab and K. Broton. To cut costs, college students are buying less food and even going hungry, 2015. [Online; accessed 15-May-2018].

[3] J. Gustavsson, C. Cederberg, U. Sonesson, R. van Otterdijk, and A. Meybeck. Global food losses and food waste, 2011. [Online; accessed 15-May-2018].

[4] Z. Jiang, K. P. Scheibe, S. Nilakanta, and X. S. Qu. The economics of public beta testing. *Decision Sciences*, 48(1), 2016.

[5] D. A. Norman. *The Design of Everyday Things*. Currancy Doubleday, 1988.

[6] I. of Mechanical Engineers. Global food: Waste not, want not, 2013. [Online; accessed 15-May-2018].

[7] O. of Student Life, C. of Education, and H. Ecology. National student financial wellness study. report, Ohio State University, 2015.

[8] P. Pate. The first comprehensive look at global food waste is as bad as you'd expect, 2016. [Online; accessed 15-May-2018].

[9] D. Qi and B. E. Roe. Household food waste: Multivariate regression and principal components analyses of awareness and attitudes among u.s. consumers. *PLoS ONE*, 11(7), 2016.

[10] S. Sengupta. How much food do we waste? probably more than you think, 2017. [Online; accessed 15-May-2018].

[11] A. Smith. Record shares of americans now own smartphones, have home broadband, 2017. [Online; accessed 15-May-2018].

[12] R. Smith. How reducing food waste could ease climate change, 2015. [Online; accessed 15-May-2018].

[13] V. Stavova, L. Dedkova, M. Ukrop, and V. Matyas. Best practices for user acceptance testing and beta testing, 2018.