UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
SESSION 2018/2019 SEMESTER 2

Code & Subject	: SCSV 2113 – Human Computer Interaction
Name of Lecturer	: Dr Aida Ali
Delivery Date	: 19 March 2019
Submission Date	: 4 April 2019

Project part #2: Establishing Requirements

Throughout class lectures, discussions and activities done, students are expected to have a clear idea on how to design, implement and evaluate a user interface. In this semester-long project, each group will have to a particular interactive system/application (either for computer desktop or mobile) that interests them. The project may aim to replace or update an established system, or it may aim to develop a totally innovative product with no obvious precedent. The project phases will reflect the stages of interaction design process, and submitted as project deliverables throughout the semester.

Process 1: Identifying needs and establishing requirements

Continuing from Submission 1 (project proposal), each group will now start to materialize their proposed solution by designing the interaction of the product/application. Designing interaction firstly require an understanding of the users and their capabilities, their current tasks and goals, the conditions under which the product/application will be used, and constraints on the product/application's performance. Each group should attempt to achieve this understanding through the process of User & Task Analysis described below:

(a) User Analysis.

- Identify the characteristics of your three (3) target user groups. Create one or two personas from each user group (totaling to minimum 3 to maximum 6 personas).
- Determine 3 tasks (with at least 2 of them as main functional tasks for your proposed system. Think of the context behind why your persona would need to do the 3 tasks you identified, and their expectations when doing it.

(b) **Task analysis and User Observation (using Think Aloud technique).**

- Purpose to find and analyze what type of tasks your product/application should offer.
- Find an existing website/system/ application that most similar to what you are proposing for your system/application. E.g. www.phdelivery.com.my for an online food delivery system.

- Identify 3 tasks (with at least 2 of them as main functional tasks related to your domain) that a user can do with the system. E.g. order food, register as member, and search for store.
- Find three (3) REAL users who represents your target user (e.g. a research student, a doctor, a housewife)
- Ask a real user representing each of your target group to use the system to complete each of the tasks, while thinking aloud (**one** of the data gathering technique that you carry out. Explain beforehand to the user what think aloud is). Give the clear instruction to user to do the task. E.g.

    - use Google calendar for 'create event' task,
    - use www.barnesandnoble.com for 'buying the cheapest C++ programming book' task.

- Record and observe each time a user doing the task in order to understand the task analysis. You should have NINE (9) videos (3 user x 3 tasks). You can record using a SCREEN recording software (e.g. Camtasia or any other screen recording software). Make sure that it can also record the voice of the user thinking aloud.
- The output of this observation of user task analysis is a Hierarchical Task Analysis (HTA) which will define the design needs/design requirements for your proposed system.

Report - Report should include the following:

For user analysis: [10m]

- **User's Personas**. Description of each persona created. **Hint**: Create their personalities in line with how your system/application can benefit them. What would be their goal when using your proposed system/application? See example of "Bob" from lecture slides.
- **Scenarios**. Description of **task scenario** for each of the tasks you have identified. **Hint**: Think of what your persona might expect/ his or her goal (this is based on their own characteristics/personality) when doing the task. See example of "Brad".

For task analysis and user observation: [20m]

- **Hierarchical Task Analysis.** Based on the user observation videos, analyze the 3 (or more) tasks you have identified. Create the HTA for each task. At least one task should have more subtasks. Identify plans as needed. See example of "Borrow a book from library". You can sketch the diagram by hand (but you will need to take a picture/scan and upload) or use appropriate drawing software. Makes sure that the HTA is constructed following the correct format (use of numbering, verb-action tasks, plans etc...). You can use textual HTA to describe Task A, and graphical for Task B, etc.
- **Discussion**. What information/issues did you obtained from the HTAs you derived **AND** from the user's thought process (thinking aloud)? Is there any difference on the way the three different users accomplishing the same task? By comparing the HTA's for each user on the same task, what can you conclude? Is there any process that can be made simpler or more effective?

- For your **proposed system**, what lessons **(design requirements)** from the HTA should you take into account i.e. which bits of the task would you keep, and which would you do away with or redesign? What insight have you learned from the user's thought process (thinking aloud)?
- See example on deriving design requirements from HTA from teaching slides (i.e. example on the filing system and cataloguing system). Please DISCUSS YOUR FINDINGS ELABORATELY.
- **Refinement of HTA.** Derive to the refined version of HTA for each task for your proposed new system/tasks.

Reference Slides – Refer on **HTA, User & Scenarios** in *Note – 05a Establishing Requirement and Note 05b-Task Analysis, Scenario, Persona*

FORMAT REPORT SUBMISSION:

Submission 1: For User Analysis

1. **{Title of Submission 1: Gathering Requirements - User Analysis}**

2. **Proposed Tasks**
   {State the 3 tasks for yours propose system}

3. **Persona**
   {A two-paragraph description of Persona 1 from User Group 1. Include his/her picture.}
   {A two-paragraph description of Persona 2 from User Group 2. Include his/her picture.}
   {A two-paragraph description of Persona 3 from User Group 3. Include his/her picture.}

   Note: For a wide-range user group (e.g. visitor), create 1 different personas

4. ~~**Scenario**~~
   ~~{Name of Task 1 e.g. Buy Coffee}~~
   ~~{A two-paragraph narration of Task Scenario 1 for Persona 1.}~~

   ~~{Name of Task 2}~~
   ~~{A two-paragraph narration of Task Scenario 2 for Persona 2.}~~

   ~~{Name of Task 3}~~
   ~~{A two-paragraph narration of Task Scenario 3 for Persona 3.}~~

Submission 2: Task Analysis

1. **{Title of Submission 2: Gathering Requirements - Task Analysis}**

2. **Introduction**
   {Describe in briefly the system/application you're testing. e.g. www.phdelivery.com.my}
   {State the 3 tasks from the existing system that you choose for observation}

3. **Refinement of HTA.**
   {Derive the refined version of HTA for each task for each user.}

   a) **HTA for Task 1 -** {Name of Task 1 e.g. Order Food}

   {Type of User 1}
   {Link/Embedded Video on User 1 doing Task 1 (e.g. research student testing tasks 1)}
   {HTA for User 1 doing Task 1 in textual presentation}

{HTA for User 1 doing Task 1 in diagram presentation}

{Type of User 2}
{Link/Embedded Video on User 2 doing Task 1(e.g. doctor testing task 1)}
{HTA for User 2 doing Task 1 in textual presentation}
{HTA for User 2 doing Task 1 in diagram presentation}

{Type of User 3}
{Link/Embedded Video on User 3 doing Task 1(e.g. Housewife testing tasks 1)}
{HTA for User 3 doing Task 1 in textual presentation}
{HTA for User 3 doing Task 1 in diagram presentation}

b) {Discussion on Findings from Task 1}

Highlight if there are any similarities or differences in HTA for all users. If there are any, what would you make of this? Which tasks will you keep or removed? Are there any *optional* or *sequence* or *cycle* tasks? [Hint: Slide 75 from Lecture Note - Persona_Scenarios_HTA. What are your suggestion for these types of task plan? Will you improve any of the tasks? Why do you want to improve or choose not to improve. Discuss and give reason.

c) {Requirement Specification for Task 1}

Derive needs/requirements for your proposed system for Task 1. Based on the HTA and plans and finding above, Formulate **1 or more requirements specifications** for your new proposed system for Task 1. For each requirement you should give a thorough justification linking it to the HTA. E.g.

Functional /~~non-functional~~ requirement: List type of food
Description: The system should provide the user with a listing of available food.
Justification: The users need to know which food are available before they can choose
One HTA 2.1 (referring to step no. in HTA)
AND

non-functional requirement: Photo of food.

d) {A proposed HTA for Task 1}

{Propose HTA for Task 1 in textual/diagram presentation after adding the new requirement specification}

e) Then, Repeat for Task 2 and Task 3

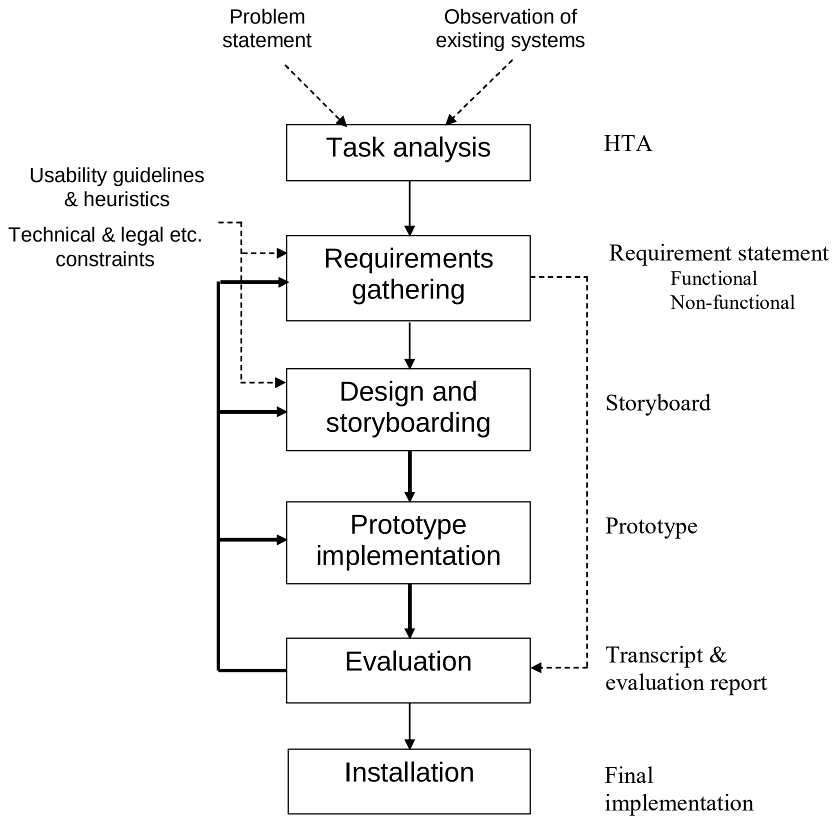
Submission by blog post:

You are required to:

- ~~Under **"Group Project"** menu, create a sub menu **"Project 02-Establishing Requirements"** and also the corresponding page for project User and Task Analysis.~~

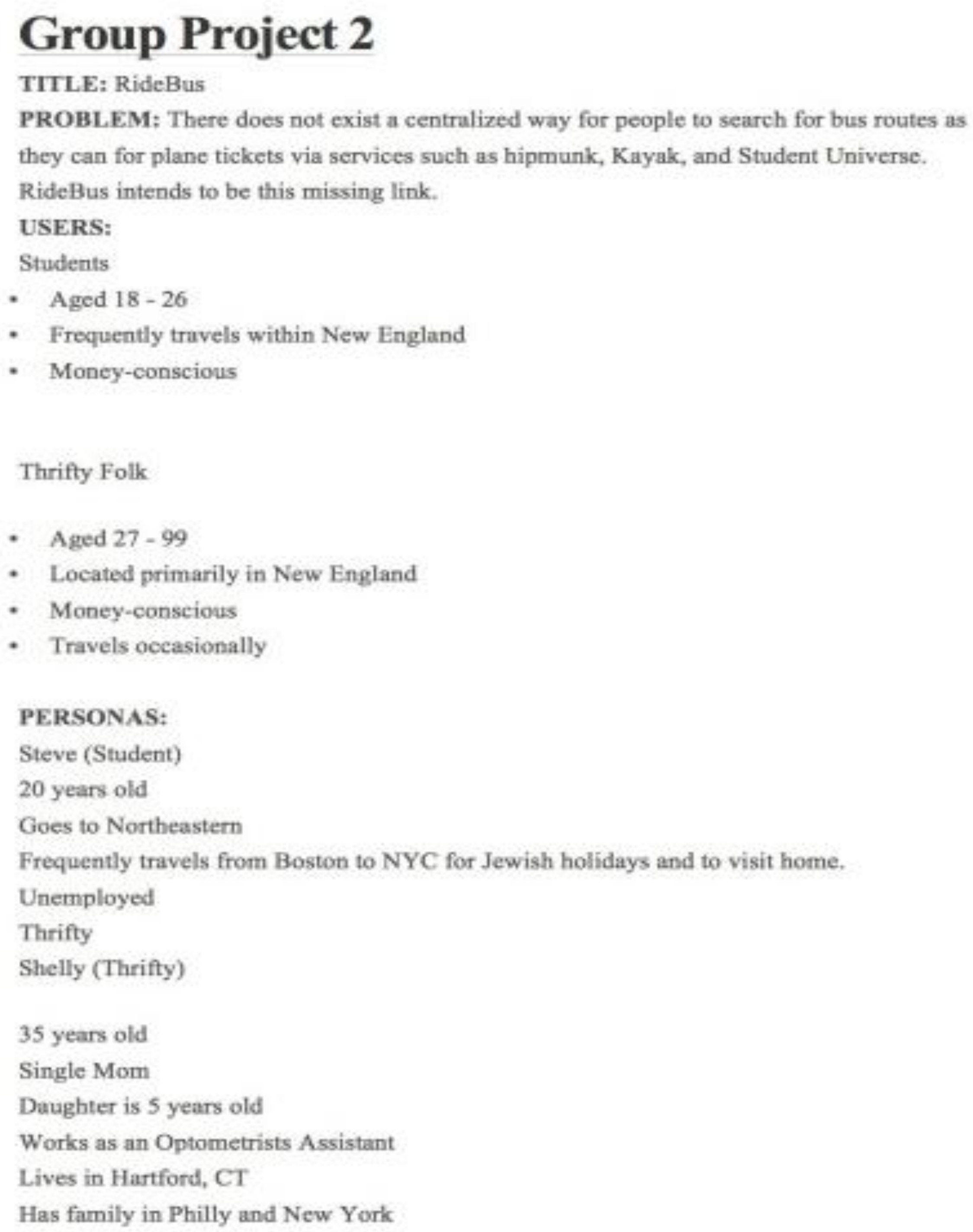
- ~~Based on the group discussion regarding the above matters (what to be done in this section), please put all the required information into your **"Establishing Requirements"** page.~~
- ~~Please upload the new posts before or on **30/11/2018 (Friday)**.~~
- Upload the full report in the e-learning, via **"Group Project 2 Submission"**

Figure 1 - User-Centered Design (UCD) Process



Example 1: Group Project 2

Task Analysis



Group Project 2

TITLE: RideBus

PROBLEM: There does not exist a centralized way for people to search for bus routes as they can for plane tickets via services such as hipmunk, Kayak, and Student Universe. RideBus intends to be this missing link.

USERS:

Students

- Aged 18 - 26
- Frequently travels within New England
- Money-conscious

Thrifty Folk

- Aged 27 - 99
- Located primarily in New England
- Money-conscious
- Travels occasionally

PERSONAS:

Steve (Student)
20 years old
Goes to Northeastern
Frequently travels from Boston to NYC for Jewish holidays and to visit home.
Unemployed
Thrifty
Shelly (Thrifty)

35 years old
Single Mom
Daughter is 5 years old
Works as an Optometrists Assistant
Lives in Hartford, CT
Has family in Philly and New York

TASK SCENARIOS:

1. Enter origin and destination

Shelly is tired of having to visit several bus company's websites one at a time in order to find the best deal when she needs to travel. RideBus eliminates this inconvenience. Instead of checking prices one at a time, all Shelly has to do is enter her travel origin and destination on the homepage of RideBus and all the work is done for her. She can quickly and easily view all bus prices at once.

2. Clear the page / resetting the search criteria

Steve is a busy guy, constantly working on schoolwork and projects. He is always trying to complete activities as quick and efficient as possible. This carries over to all areas of Steve's life. Recently he has practiced becoming a very fast typer seeing as it would be one more way to improve his efficiency when using the computer.

Yom Kippur is coming up and Steve needs to find the cheapest bus ticket to NYC so he can be with his family. He visits the recently-built bus route aggregator RideBus and enters his origin and destination. While doing so, Steve makes a typo in his origin. Instead of having to precisely use the mouse to highlight the entire string in the origin box, all Steve has to do is click the "Reset" button to the side.

3. Switch origin and destination

Shelly is hardly the computer pro — she's a Netflix and Pinterest pro, but that's about it. She types like a snail, and she's meticulous about every key she hits. She's using RideBus to plan a trip from Hartford to Philly, and she's ready to get some results… but she just realized she typed Philly to Hartford, the opposite of what she actually wants. Luckily, there's a button right there next to the origin and destination fields to swap them. This simple button is great for people like Shelly because it quickly mitigates the frustration of input error.

4. Share a bus route

Steve's getting ready to back to New York City for winter break, and he and his friends are looking to take a bus back. Steve's a little thriftier than his friends so he tells everyone that he'll take care of finding the bus fares. Rather then spending his night hopping from site to site, he uses RideBus to quickly compare prices from Boston to New York. Once he's found the right trip, he can quickly share the trip with a generated URL, and he's done. Not only is Steve happy he got the cheapest fare, but his friends are glad they hardly had to do anything.

5. Filter results

Precondition(s): User wants to reverse origin and destination, so these should already have some input in the fields.
Exception(s): There is no input in the fields

- This should make it quick and easy for the user to plan a return trip, a common task. It also allows the user to quickly fix an error in their input if they gave their cities in the incorrect order.

4. Share a bus route

Goal: To let the user share the route with a travel companion or parent who may be paying for the booking.

Precondition(s): The user has found a valid, satisfactory bus route.

Subtask(s): Create a valid, shareable URL for the route

Exception(s): As long as the precondition is met, there shouldn't be any exceptions.

- This feature is useful for anyone booking a bus trip with a friend or a user who wants to let a friend or family member know their itinerary.

5. Filter results

Goal: To filter a query by conditions like price, time of day, station, etc.

Precondition(s): The user has entered their origin, destination and date of departure, but wants to fine tune his results with other conditions.

Subtask(s): Use the knobs, sliders, and other UI elements to filter the results.

Exception(s): As long as the precondition is met, there shouldn't be any exceptions. However, applying certain filters may return 0 results.

- This feature isn't just useful, but essential for anybody who wants to find a bus ticket that matches specific conditions involving time, location, etc.

6. Create an account

Goal: Store user information for better usability

Precondition(s): User knows about RideBus and plans to use it.

Subtask(s): User must input valid information into the required fields. Input should be validated and sanitized.

Exception(s): Account already exists, user fails to enter all information, user enters an invalid account name

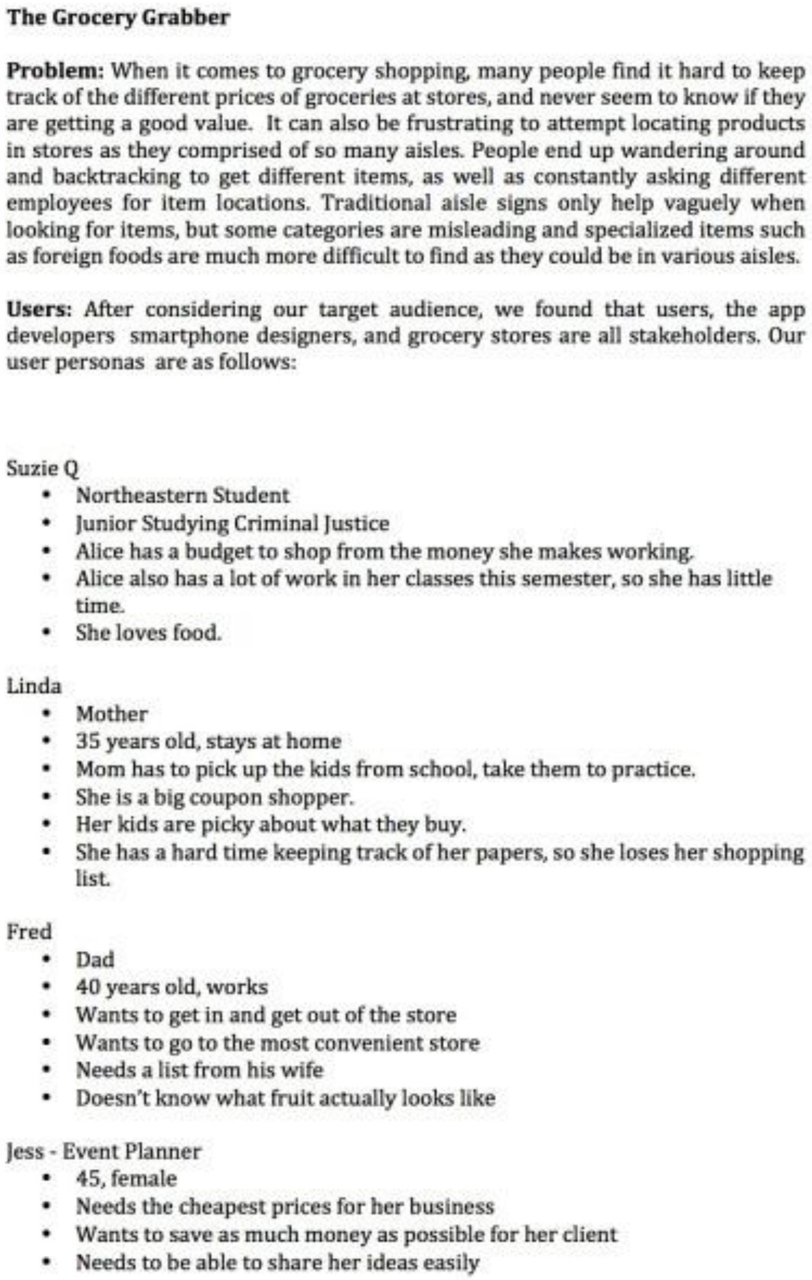
- One relevant feature of this task is that it will help improve the experience of frequent users of RideBus. Creating an account will allow them to store information such as frequently traveled routes, favorite bus lines, trip templates, etc. By storing this information, frequent users will be able to use the site more quickly and more tailored to their individual profiles. This will lead to increased satisfaction with using the service.

ROLES:

- *Ben Lyaunzon*: HCI Researcher. With team, came up with tasks and personas. Wrote a task description and task scenario.
- *Isaac Boehman*: HCI Researcher, Project Manager. With team, came up with tasks and personas. Wrote a task description and task scenario. Organized all information from in-class work and assigned roles for the assignment, created skeleton layout for assignment posting.
- *Jake Ascher*: HCI Researcher. With team, came up with tasks and personas. Wrote task descriptions and scenarios
- *David Ciatto*: HCI Researcher. With team, came up with tasks and personas. Wrote task descriptions and scenarios.

Example 2: Group Project 2

Task Analysis



The Grocery Grabber

Problem: When it comes to grocery shopping, many people find it hard to keep track of the different prices of groceries at stores, and never seem to know if they are getting a good value. It can also be frustrating to attempt locating products in stores as they comprised of so many aisles. People end up wandering around and backtracking to get different items, as well as constantly asking different employees for item locations. Traditional aisle signs only help vaguely when looking for items, but some categories are misleading and specialized items such as foreign foods are much more difficult to find as they could be in various aisles.

Users: After considering our target audience, we found that users, the app developers smartphone designers, and grocery stores are all stakeholders. Our user personas are as follows:

Suzie Q

- Northeastern Student
- Junior Studying Criminal Justice
- Alice has a budget to shop from the money she makes working.
- Alice also has a lot of work in her classes this semester, so she has little time.
- She loves food.

Linda

- Mother
- 35 years old, stays at home
- Mom has to pick up the kids from school, take them to practice.
- She is a big coupon shopper.
- Her kids are picky about what they buy.
- She has a hard time keeping track of her papers, so she loses her shopping list.

Fred

- Dad
- 40 years old, works
- Wants to get in and get out of the store
- Wants to go to the most convenient store
- Needs a list from his wife
- Doesn't know what fruit actually looks like

Jess - Event Planner

- 45, female
- Needs the cheapest prices for her business
- Wants to save as much money as possible for her client
- Needs to be able to share her ideas easily

Steve

- 50 year old
- Is a big business CEO
- Single, so he only shops for himself
- Likes to not have to go far to shop
- Buys whatever he actually wants

Task Scenarios:

Rating System

Suzie Q is a Northeastern Junior who has a tight budget when it comes to her groceries, but she also has a weakness for delicious food...

She has been very responsible by staying under budget, but she often finds that the cheapest items and generic brands sometimes taste horrible. As she has a little more money than she currently spent on all her lowest priced groceries, she wondered if items that were slightly more expensive would taste significantly better and began to purchase them.

After trying various foods at different price levels that still let her remain within her budget, Suzie noticed that there was no direct correlation between price and food when it came to brands that were less popular. Suzie then considers that it would be great if she could utilize a mobile application that would allow her to see these items rated by fellow shoppers to achieve the best quality for the price.

Add/Delete

Linda is a 35 year old homemaker. Every Thursday she has to take her kids to soccer practice, and she likes to do her errands before coming back to pick them up. There is a grocery store right near the soccer field but it doesn't always have good prices. She doesn't have much time, and is on a budget. So she uses her Grocery Grabber app to tell her what things to buy there, and what things to buy at another store she frequents.

Grocery Grabber also helps her keep track of her grocery list. When she runs out of something at home, she just scans the barcode and it gets added to her list. Then at the store, Grocery Grabber tells her which items on her list have good prices. Sometimes she knows she wants a fruit or vegetable, but doesn't have an example with a PLU in front of her. In that case she can search for it in the database Grocery Grabber keeps, and add it to her shopping list.

While she is shopping she she check items off her list, and they get transferred to a past items list so they can be easily transferred back to her shopping list.

Finding Items and Routes

Linda is a stay at home mom who is constantly on the move and does not have a lot of time to spare. Not only is she concerned with getting the best deals on her groceries, but she also wants to make her actual shopping trip as quick as possible.

She often goes to different supermarkets depending on where she is at the time because she is often running all over town. She often mixes up the organization of the stores and spends more time looking for specific items than she needs to.

Linda wishes there were a better way for her to just know where items should be in the store, so she does not have to waste time going up and down the aisles of the store looking for one thing.

Sharing Shopping Lists / Pictures

Fred is Linda's husband. He works long hours and is usually pretty tired by the end of the day. He just wants to get home to put his feet up and relax but sometimes, his wife needs him to pick up extra food at the store for dinner.

Fred does not pay a lot of attention to different fruits and vegetables because they are not really his favorite food groups. So when Linda tells him to get kale and spinach and raspberries at the store, he needs a visual aid in order to know exactly what he is looking for.

Linda often texts him pictures of the fruits and vegetables that she is looking for, but finding the pictures is tedious and they both wish there were an easier way.

Finding the Cheapest Prices

Jess the Event Planner's job rests on her capability to quickly and cheaply coordinate basic catering tasks; whether this means buying food herself or providing lists to workers or clients, she has to be able to easily locate the lowest prices on food and supplies, among both competing stores and brands.

She wants to be able to make a list of the things she needs and have it be automatically sorted into sublists for a set of local stores, where each item on a sublist is the cheapest at the store it's been sorted to. Most items can be of any brand, but sometimes she'll have to specify a particular one. She also wants to be able to look at the total amount she'd have to spend if the list were split that way, and compare it to amounts from splitting it between more or less stores; if she can save $50 by including one more store, she'll add it in, and the opposite holds if she'd only lose $2 by dropping another store.

This task should be easy and fast since it will be done often. It should be doable with one hand while the other hand is holding items. There probably should be multiple lists for different kinds of groceries, or for different contexts (personal vs for events).

Finding Items and Routes

The goal of this task of finding specific items within a store and mapping out the best route to take in that store in order to minimize the time spent shopping. A sub-task / precondition for this task is making the shopping list itself. The user must add his/her items to the list first because without that there is no way to determine which route to take through the store. Then, the app would use an algorithm to determine the quickest route to take through the store in order to spend the least a mount of time walking around, and it will indicate where on the route each item can be found. It could also rely on GPS to provide a turn-by-turn direction experience. This feature would require individual supermarkets to supply their floor plans and how everything is laid out in their specific locations. This additionally means that any stores they change their layout, the data in the app would need to be updated. Otherwise, it may be incorrect.

Sharing Shopping Lists / Pictures

The goal of this task would be to have a sharing option so that multiple people can access the same shopping list, as well as include pictures of the items that they are shopping for. A precondition (or sub-task), again, is making the shopping list to begin with. Then, the list could be shared to one person or to many other people. The sharing feature would also automatically include a photo of each item, which the user could then choose to turn off if they do not need it. This would allow easy communication between families. There would also be a feature that would allow a user to check off when they have purchased something and it would alert the other people who have access to that list. There could potentially be inconsistent data if one person checks off an item and it does not sync properly.

Finding the Cheapest Prices

Provided with a list of grocery items and local stores, divide up the items among stores so as to minimize the overall cost (find the cheapest price for each item).

Preconditions: The user must first build a shopping list and provide a location (automatically via GPS/Network positioning or via manual entry).

This task should be easy and fast since it will be done often. It should be doable with one hand while the other hand is holding items. There probably should be multiple lists for different kinds of groceries, or for different contexts (personal vs for events).

Finding Items and Routes

The goal of this task of finding specific items within a store and mapping out the best route to take in that store in order to minimize the time spent shopping. A sub-task / precondition for this task is making the shopping list itself. The user must add his/her items to the list first because without that there is no way to determine which route to take through the store. Then, the app would use an algorithm to determine the quickest route to take through the store in order to spend the least a mount of time walking around, and it will indicate where on the route each item can be found. It could also rely on GPS to provide a turn-by-turn direction experience. This feature would require individual supermarkets to supply their floor plans and how everything is laid out in their specific locations. This additionally means that any stores they change their layout, the data in the app would need to be updated. Otherwise, it may be incorrect.

Sharing Shopping Lists / Pictures

The goal of this task would be to have a sharing option so that multiple people can access the same shopping list, as well as include pictures of the items that they are shopping for. A precondition (or sub-task), again, is making the shopping list to begin with. Then, the list could be shared to one person or to many other people. The sharing feature would also automatically include a photo of each item, which the user could then choose to turn off if they do not need it. This would allow easy communication between families. There would also be a feature that would allow a user to check off when they have purchased something and it would alert the other people who have access to that list. There could potentially be inconsistent data if one person checks off an item and it does not sync properly.

Finding the Cheapest Prices

Provided with a list of grocery items and local stores, divide up the items among stores so as to minimize the overall cost (find the cheapest price for each item).

Preconditions: The user must first build a shopping list and provide a location (automatically via GPS/Network positioning or via manual entry).

Sub-tasks: Select the shopping list to divide. Select the stores among which you want to split the list, or a radius within which to select all stores. Include/remove stores based on how their inclusion/removal affects total cost.
Finalize/save the divided lists.

Exceptions: If an item on the list is not available at the selected stores, the user should be notified before being shown the list breakdown, and that item should be sorted to a 'Other Store' list.

This task will be performed often, especially by college students and families with tight budgets. As such, it should be kept as streamlined as possible; using saved lists and store selections should make the price sorting quick and painless. An option could be included to perform a 'quick edit' on a saved list, removing anything you don't need more of or adding a one-off item or two.

Finding Local Stores and their Information

Given a location, provide map and list views of nearby grocery stores, as well as their hours and price levels (in comparison to other stores).

Preconditions: The phone should have a source of location data (GPS, network, WiFi, etc.). If it does not, prompt the user to enable one of the sources or enter their address manually.

Sub-tasks: Set the radius within which you're searching for stores (default to 5 miles). Switch between the map and list views. Sort the list view by distance or by store name. Select a store to see detailed information about it.

Exceptions: If there are no stores within the selected (or default) radius, inform the user and prompt them to change it rather than displaying an empty list/map.

This task is mainly useful for users new to an area, or who wish to know more about their shopping options. It will not be used as often as the list/pricing/mapping sections of the app. When looking at a store's detail page, users can get directions to that store if they wish.

Team Roles and Tasks:

Carly will generally be interested in holding the role of designer.
Jonathan will be the software engineer.
Liam will also be a software engineer and/or a test engineer.
Arash will be business analyst/project manager as well contributing to Testing.
Matt will contribute heavily to identifying and implementing the design process.

We decided upon these roles through reviewing our strengths and experience at past Co-Ops, and believe that this division of roles and strengths will help us be successful.

For this project:
Carly, Jonathan, Liam, and Arash contributed to defining the Task Scenarios and Descriptions, while Matt created the Roles for all the personas. The rest we did collectively.