Software Engineering Homework 2 Daniel Crawford

* 1. My solution to this problem is going to be a UTD dining app which will allow a user to schedule when they want to eat and pay online, then the food will be prepared for them when they get there. That way, the lines will not be as clustered and results in less traffic. Once the initial development phase is done, I will give it to the people are see how they are using it. First, I will do the interview, which will ask people how quickly they think they can perform these orders, if they believe they are reliable and questions about quality of the program. With these responses, the code will be developed to fit some of the recommendations from user stories to polish the code. Ethnography will be very important to this process because it will show what the most used features are, and what is not providing results. Because this will be an app, most of these observations will be made through what the data tells us. So, for example, if people are starting to use the feature to reserve food through the app but stop in the middle, it probably means the transaction process is too tedious. Besides just the data, observations can also be made in the Student Union to see what other places the crowds cluster in. Features can be made to combat this as well and can be interleaved into the development process.
  2. **Initial Assumption:** Users will use the app to avoid the lines and will prefer a slightly more difficult transaction phase over a long wait in a line. After they place the order, the restaurant will prepare the food and the student can give evidence with their phone and pick it up.

**Normal:** The user chooses to buy from a restaurant and is brought up a menu. From this menu they can choose exactly what they want, and it will be stored within a cart. Once the user is ready, they will transition to the transaction phase. In this phase, they will type their credit/debit card information and give them an option to save it, and have it secured through a fingerprint or password. After this, it will auto fill all the fields and request the user for verification. The user will verify, and the order will be placed at a certain time.

**What can go wrong:** The user could correctly input the wrong credit card information but save it, making it so every time he comes to order he has incorrect information. Users should have the ability to edit these fields quickly and the program should identify the problem as fast as it can.

Another issue for concern is that the restaurants may not supply the orders that a user placed. For this reason, an accessible refund option that cannot be used after a verified pickup for food should be implemented.

**Other Activities:** A user may be ordering from several different restaurants at once using the same app.

**System State on completion:** User sees that food is pending and being prepared, transaction says that it is approved and data is presented that shows the current crowd in the Student Union.

* 1. 1. App should have an accessible refund system that is secure, there should not be an exploit to refund an item that has already been picked up

2. There should be a time graph that represents the crowds around certain times. This way, people will decide different times to eat at the SU based on times of lesser traffic.

3. App should not act as a medium for advertising since the user is already going to be purchasing a product. Users will likely uninstall the app if they feel aggravated by it.

4. Owners of restaurants should be able to modify their own pages so that they tell users that the stock of an item is currently out or if the place is closed for a special holiday.

5. After purchase, the program should tell the user that the food is pending, cancelled, ready or any state that is necessary.

6. App should support ordering from multiple restaurants at once, so should display a list on a page which shows all the food ordered, and their status.

## External requirements:

1. Make sure that app securely holds all information about the credit card, encrypt precious data.
2. Display calorie and nutrition information so that users can maintain a healthy diet.

Organizational Requirements:

1. Should be developed using Java so that there is cross platform with Apple and Android. Preferably Android Work Studio
2. Users save their data using fingerprints, pins and passwords. Once authenticated, can autofill credit card information. Should have priority and be quick authentication.

Product Requirements:

1. App should launch relatively quickly and not noticeably lag. Users use app so they can quickly buy food, so using app should feel quick as well.
2. App should look simple and organize all restaurants participating nicely. Do not allow companies to customize their restaurants with their own personal looks, maintain consistency.
   1. Domain requirements:
3. All credit card information held should follow PCI DSS requirements
4. Want to keep code consistent, follow a guideline how its organized in an object-oriented language like Java.
5. Software must be made to be easily updated with new technology.