**Milestone 2 - Team 4**

1. **System Overview - (Bryan)**

**Product Description:** Application which monitors a store’s capacity real-time and has an online ordering/pickup feature.

**B. Technical Requirements (Nick)**

* 1. While this portion has been heavily covered in section D, the main functional requirements are that 1. The software allows the manager to see the current capacity of the store as determined by local gov’t/business regulations, change the stores allowed capacity in the event that gov’t or business instructions change, and if the store is at capacity, see the estimated wait time for those who would like to enter the store. 2. For the in-store shopper, the software displays the current percent capacity of the store, and if full, the store displays the amount of time they would have to wait to shop in person, making recommendations based on the length of the line to deter Covid/viral spread. C. For the ordering/curbside pick up
  2. The app should be easy to use for the shopper, either by using an API to allow access via a web browser or downloading the app for their phone.
  3. Regarding the user interface, this has been discussed previously in prior sections, but it should have a shopping interface implemented either in the browser and communicating with the store’s server via an API or the app itself. In the case of the manager, their portion of the application will be solely implemented via the application and the user interface will be designed for the phone.
  4. Once again, depending on whether or not the user is a shopper or manager, the task flow for the manager is as follows Validate Credentials → see current status of store capacity and line → offer the ability to change capacity.

For the shopper, the task flow is as follows sign in→ select whether or not they are ordering online or shopping in store→ in the first case, if they are shopping online, creating a cart, adding items to the cart based on availability → pay for goods → schedule a pick up time → receive a digital receipt. For the second case if the shopper is going in store user credentials → see current capacity and line → receive a recommendation whether they should come to the store now or suggest better times to come.

* 1. The sole inputs on the manager side are user credentials and allowing the manipulation of a global var that determines story capacity. For output, the store manager receives the current capacity and line.

For the shopper, depending on whether in person shopping or curbside pickup, they will require different inputs based on either case. In the case of in person, the only input is user credentials, and the output is the current capacity and a recommendation whether to come to the store or come back later. For the ordering shopper via either a webpage or the app, they will provide user credentials, strings to search for products, ints to select how much of an item they want to buy, credit card information to pay for the goods and services, and a string/selection from a drop-down menu for pick up time. They will receive a receipt, booleans on whether or not the products are available, and a pick up time.

* 1. There are three cases where API’s will need to be employed in order to achieve full functionality. In the first case, there will need to be a way to access the app in a browser for individuals who may not have a smart-phone. In case two, there will need to be an API to interface with the stores inventory in order to determine whether or not a. the item is in stock and b. if in stock, how much of the item(s) are available, and c. prices for the items. Finally, in order to complete the transaction, a payment API will have to exist to process credit card transactions to pay for the goods.

**C. Acceptance Criteria/Interaction Scenarios (Kishan)**

This section should define the functionality the software must implement. It can include interaction scenarios. The scenarios consist of the user inputs and system responses. The low-fidelity prototypes are included. The type of questions that should be asked include:

* 1. i. How-to: Questions ask how some action is performed.
  2. ii. Who: Questions ask who is responsible for a task.
  3. iii. What-kind-of: Questions request further refinements of some concepts.
  4. iv. When: Questions ask about timing constraints.
  5. v. Relationship: Questions ask how one requirement is related to another.
  6. vi. What-if: Questions ask about cases in which an action could go wrong
     1. or its preconditions.
  7. vii. Follow-on: Questions stem from other pending questions.

**D. Validation/Verification—-Santhoshini**

Validation

**Use Cases**

Goal : should be able to select delivery or store pickup

Actor : Buyer

Trigger : Buyer should be able to login

Entry Condition: Buyer should be able to validate the credentials

Post Conditions : Can perform required operations

Basic Flow: Buyer should be able to login and validate credentials and should be able to select Delivery or Store Pick up

Alternate Flow: Credentials not working

Goal : display the waiting time if customer wants to visit the store

Actor : Buyer

Trigger : Buyer should be able to login

Entry Condition: Buyer should be able to validate the credentials

Post Conditions : Can perform required operations

Basic Flow: Buyer should be able to login and validate credentials and should be able to be able to see the waiting time if they wants to visit the store

Alternate Flow: Credentials not working

Goal : Buyer should be able to add multiple products to the cart and checkout

Actor : Buyer

Trigger : Buyer should be able to login

Entry Condition: Buyer should be able to validate the credentials

Post Conditions : Can perform required operations

Basic Flow: Buyer should be able to login and validate credentials and should be able to able to add multi items to the cart

Alternate Flow: Credentials not working

Goal : Various payment methods should be available for Buyer

Actor : Buyer

Trigger : Buyer should be able to login

Entry Condition: Buyer should be able to validate the credentials

Post Conditions : Can perform required operations

Basic Flow: Buyer should be able to login and validate credentials and should be able to able to add multi items to the cart .Buyer should be able to select one among the various payment options like Credit card ,Cash on Delivery Or Wallet etc

Alternate Flow: Credentials not working

Goal : App should be able to show the delivery date and time

Actor : Buyer

Trigger : Buyer should be able to login

Entry Condition: Buyer should be able to validate the credentials

Post Conditions : Can perform required operations

Basic Flow: Buyer should be able to login and validate credentials and should be able to add multi items to the cart . Before Placing the order Buyer should be able to see the Delivery date and time and should be able to change according to their convenience

Alternate Flow: Credentials not working

Goal : App should also allow the customers to reschedule ,cancel the order if needed

Actor : Buyer

Trigger : Buyer should be able to login

Entry Condition: Buyer should be able to validate the credentials

Post Conditions : Can perform required operations

Basic Flow: Buyer should be able to login and validate credentials and should be able to add multi items to the cart . Before Placing the order Buyer should be able to see the Delivery date and time and should be able to change according to their convenience . After ordering they should be able to cancel any time if needed and refund should be processed.

Alternate Flow: Credentials not working

Verification of the system or application

1. application should not suspend while placing an order
2. application should not have performance issues

**E. Requirements Considerations - (Bryan)**

**Assumption made about the software:**

* The mobile device has cellular or Wifi connectivity.
* So that we do not have to monitor locations of users, we would ask stores to provide a mechanism to know incoming / outcoming. (simple turn-style system would work)
* It will be designed and published in the English Language (future implementations can be localized)
* The stores have a database of products already setup that we can tap into. For the purposes of this app we will create some example product listings ourselves.
* The application will not make future capacity level predictions in version 1 (hopefully to be released in a later version).
* Version 1 UI will be basic with the goals mainly to get the core functionality in place.

**End users: Describe each type of user:**

* **Store Managers:** 
  + **Who are they:** Owners of the store in charge of the total business operations and meeting local regulations.
* **Private Citizens:**
  + **Who are they:** Visitors and purchasers of the store.

**Existing systems:**

* Many modern day grocery stores have applications that allow ordering from home and going to pickup. There was a spike in these types of features during the pandemic.
* We will do some market research of these systems to capture the high and low points.

**Environment:**

* The application will run on Android and iOS mobile devices.
* The application should also be web-based to be accessed via browser.

**Limitations: The system will NOT do the following: (out of scope)**

* The app will not track promotions or discounts unless the store manager updates their database.
* The app will not provide payment services native within the app. We will use an outside third party for payment processing.
* The system will not be legally liable for the information published in the app and warning messages will be displayed that the information is for reference only.
* The app will not predict future levels of capacity so that a user can plan a trip in advance. (However this is functionality we would like to add in version 2 by monitoring the statistical data and making predictions.)

**Rationale: Describes how the requirements meet or exceed the needs of the customer.**

* Currently the store has no mechanism for a customer to purchase online and pickup. This will provide HUGE value to the customer avoiding evening having to go into the store at all.
* If a user must enter the store, at the current time there is no way to know the capacity level inside the store until they are already inside. This application will provide reference information so that the user can make an informed choice of when to go to the store.

**F. Other Information**

Any pertinent information can be added to this document.

**Review of progress: (Bryan)**

* **Review of progress since the previous milestone.**
  + Since Milestone 1 we have continued to mature our understanding of how we should be setting requirements. This has helped us as a team refine our goals/objectives for this program and they are reflected in the information above.
  + Our program plan was updated as we learned how we are going to be sharing responsibilities.
  + Github was implemented near the end of Milestone 1 (we were using google drive) and now we are continuing to use it. To aid in documentation update real time we are still using Google Drive but are uploading to Github and the end of a days work. Github is not exactly the best when documentation is being treated as a live document and multiple people are making edits (however it does work well for code).
* **Revised schedule and plan for the remainder of the project.**
  + Our plan has basically stayed the same but we brought focus to the topics that each of us are focusing on. As we learn in lecture we are applying the concepts to our program plan.