Kiersten Schmall, Kylie Norwood, and Elijah Ives

CSCI 234 – Software Engineering

Spring 2019

Project Initial User Requirements - **Updated**

Client is the Schaper Sandwich Shop. There is currently one location of this business. The client is considering starting a delivery service in order to keep up with competitors. The client wants to franchise their business. The client also wants to dominate the sandwich business. To do so, it is believed that rather than create a traditional delivery service, it would be better to put the entire sandwich shop in a truck and have the truck drive to customer houses to deliver sandwiches. A sandwich truck will have the entire ability of a brick-and-mortar store.

The client wants to also update their menu. The client has specialty sandwiches that their customers love. They will also determine the best (most desirable) menu based on customer surveys. The menu must contain sandwiches that can be prepared in the limited space of the truck.

The basic operation of a truck will be as follows:

- Customer places an order by phone or online website or in person.
- Menu on truck will include customizable sandwiches, specialty sandwiches (which allow modifications), sides (e.g. chips), and drinks. Truck will also function as a deli, allowing customers to order cold-cut meat in bulk.
- Order preparation must be as close as possible to delivery time so as to keep the sandwiches as fresh as possible. If two people order at the same time, truck fills the order closest to its current location.
- The orders are prepared and packaged for delivery.
- When a destination is reached, an employee takes the order to the customer's door and collects payment.
- If the truck has no orders, it will patrol neighborhoods and make road side sales.
- If the truck runs low of a product, it will schedule a stop at a supply center to restock (gas included) which will be located at the center of the delivery area and gas station. The Delivery area will have a 10 block radius.
- The truck has a driver, an order taker and deliverer, and an order preparer (3 people total) If and order is being Delivered the driver acts as a secondary Ordertaker.

The client wants a software system that will run the business of the truck. To test the concept of a mobile sandwich shop, the client wants a simulation of the operation of the truck. The software must:

- Allow for order entry, order editing, and order cancellations.
- Scheduling of order preparation and packaging. Will use a number system to keep track of orders, unless customer as an account. Then it is possible to remember orders, recurring customers and even payment information.
- Keep track of the cash register that only accepts bills up to $20 as of now. The starting amount in the register would be $200 (may change).

Other Technical Aspects/Low Priority Ideas:

- Accept cards (hoping as a primary), largest bill will be a 20
- Consider traffic - we will try to address this idea later in the process
- If there are no orders, have the software direct the truck to the distribution center (can restock when not busy)
- There should be control as to how the preparer should make the orders in order to be as fresh as possible

Tasks Completed:

- Orders will be put in a file, will be read, and processed in that order
- The neighborhood will be a square (20 blocks by 20 blocks), print
- Driver will have to follow the route given by the software
- Customers must be able to keep track of the location of the truck.
- The address of the customer is used to determine the route the truck will follow. This route is updated each time an order is placed. Routes must not cause indefinite postponement of deliveries.
- Each order is placed in a queue of orders such that each order will be ready before the time the truck reaches the delivery destination. Priority queue now prioritizes by order times.
- Generate 100 random order times to test system.
- Generate a file that will contain addresses and the order times.
- Computes and outputs the length of the truck's route, but does not show route yet.
- Displays a simple simulation of the truck's movement.
- Generates cost effectiveness of the route used by the truck.
- Neighborhood size can be adjusted easily if desired
- Simulation will remove a house when the delivery is completed

- Schedule delivery times and the truck route **(currently random).** There must be an easy way to change the heuristics used to create the routes so that the company can experiment with different strategies **(currently strategy)**.
- Observer pattern implemented by having the truck observe a graphical user interface display and a monitor display.
- Changed cost effectiveness to be based on time and distance now calculates by miles instead of units.
- Two display methods completed: monitor and GUI **(currently observer pattern)**
- Order events class added that will create the priority queue **(currently singleton pattern)**
- Time units updated to hours/mins/secs and distance updated to miles.
- Documentation in the code added.
- Code is flexible and able to be updated for multiple trucks, neighborhood size, and any changes in time or travel units.
- Second route MOSTLY functioning properly. A few bugs that have not be corrected yet.
- Decorator pattern implemented to construct orders for customers.

The client knows that the requirements for this operation are not complete and is relying on the software development company it hires to help flesh out additional details (or even ideas) regarding the mobile sandwich truck.

Story #1:
Blocks range from 0 to 19. Houses range from 10 to 90, 110 to 190, and so on East is our X axis and corresponds to the horizontal streets, and South is our Y axis which corresponds to the vertical streets. The neighborhood is a square and the distribution center is at 910 South 9th St. Each block is a quarter mile square and the houses are equidistant, but we can consider the distance to be measured in “units”. The truck is currently at a fixed location: the distribution center. Distance should be the road distance rather than the shortest path through the target since we can't drive through houses. The addresses are stored in a priority queue where the highest priority is the distance the address is from the truck.

Story #2:
Add a random time between 10:00am and 7:00pm associated with the address created to represent a time that the order was placed. The orders then should be added to a data structure (priority queue) which is ordered by the times of the orders. The truck will then use that ordered list to compute a route that it will follow for deliveries. We should

then be able to compute the length of the route the truck travels. This should be represented on a GUI display with the truck starting at the distribution center.

Story #3:

Using the strategy design pattern, we are to create two separate routing methods that the truck could follow. It would be our original route and an additional route that either takes only right turns. A random order is to be added to each delivery location, where each customer can order Sandwich 1, 2, or 3, Chips 1 or 2, and drink 1, 2, 3. Client also wants to be able to adjust the size of the neighborhood easily (static variable). For now, change size of the neighborhood to a 10x10 grid, with the distribution center at 510 East 5th Street. When the truck arrives at a designated delivery location, it must show that location was visited on GUI display (change color of delivery location). Must compare the cost effectiveness of each routing method, following the logic of:

- move from one address to another in 1 unit of time
- a stop at a delivery address takes 5 units of time
- a right hand turn takes 2 units of time
- a left hand turn takes 4 units of time
- time to prepare a food order is 5 units of time
- compute the total length of each route in distance and time

Story #4:

Time units should change to hours, minutes and seconds, with the assumptions that truck travels 30 miles an hour, locations are 0.03 miles apart, and values for these assumptions should be easy to change (static variables). We should refactor our code so that a new design pattern is implemented (Observer Pattern). The observer pattern will be implemented using two display methods: one through the GUI and one through the monitor. They display methods will observe and update based on the truck's location. We must also implement the Singleton Pattern, which will be an "Order Events" class that contains the priority queue of orders/addresses. Efficiently document our code with who worked on what, and who helped with what. Complete all tasks from previous sprints. Client would potentially like to expand their business, so our code should be able to handle the creation of more than one truck.

Story #5:

Start by finishing the items on the backlog. When the route is finished and all orders are completed, the truck must return back to the distribution center. We must implement EITHER the Decorator Pattern or the Factory Pattern to represent customer orders.

When the sandwich truck reaches a delivery destination, we should print the order, the cost, and the time it took to prepare. The following data will be used to compute the calculations for the orders:

Meat:

- Ham $1.50 (30 secs), Turkey $1.25 (30 sec)

Bread

- Roll $0.75 (60 sec), Wrap $0.50 (75 sec)

Condiments

- Mayo $0.25 (30 sec), Mustard $0.25 (30 sec), Cheese $0.75 (40 sec)

Vegetables:

- Lettuce $0.50 (20 sec), Tomato $0.75 (35 sec)

Sandwich paper cover $0.50 (75 sec)

Order Bagging per sandwich: $0.75 (20 sec)

Tax: 10%

We will then construct a presentation for the client and the product owner that demonstrates the project, design, what works, what does not, and any unfinished backlog items. The demonstration will show and generate 3 random delivery addresses and show both routing techniques. Each routing technique will display the total distance the truck traveled and the amount of time is took the truck to travel that distance.