

COMP 1531

SOFTWARE ENGINEERING FUNDAMENTALS

2019 TERM 1

PROJECT: GOURMET SUPER FRESH

Team: Super Fresh

Sherry Liu z5208583

Cuthbert Zhong z5206138

Eun-Sang Yu z5208376

Due date / Sunday 28th April

Initial Epic Story:

Key:

Priority: Ranges 1-5, where 1 is highest priority and 5 is lowest priority

Size: 1 Story Point = 2 hours

1. As a customer, I want to be able to place a food order online. I should also be able to view the status of my order at any point.

ID	US1
Name	Select Main
User-Story Description: As a customer, I want to be able to select a main so I can choose the most suitable meal. Acceptance Criteria: <ul style="list-style-type: none">• Should be able to navigate to a burger or wrap page from the navigation bar to add mains to order• Should be able to select to create a custom main or choose from a list of standard mains• A custom burger may be single, double, or triple; selectable by the customer via a radio button at the start of the customisation page• Single burgers must have 2 buns and 1 patty• Double burgers must have 3 buns and 2 patties• Triple burgers must have 4 buns and 3 patties• Should be able to enter a quantity for each ingredient listed on the burger/wrap customisation page• Quantity selected must be ≥ 0• Confirm button at bottom of page• The max quantity allowed for each ingredient is 5, except for buns and patties which will have a max of 4 and 3, respectively• If the customer selects an invalid combination of number of buns and patties, an appropriate error message will be displayed on the page when attempting to confirm the item• If the quantity selected of a particular ingredient (in a burger or wrap) exceeds the maximum allowable, the customer is notified by an appropriate error message when attempting to confirm the item	
Priority	1
Size	4 story points
ID	US2
Name	Select Side
User-Story Description: As a customer, I want to be able to select my sides so that I can choose the most suitable meal. Acceptance Criteria: <ul style="list-style-type: none">• Should be able to navigate to a sides page from the navigation bar to add sides to order• Should be able to select quantity for a side listed on the sides page• Quantity selected must be ≥ 0	

<ul style="list-style-type: none"> Confirm button at bottom of page 	
Priority	1
Size	4 story points
ID	US3
Name	Select Drinks
<p>User-Story Description:</p> <p>As a customer, I want to be able to select my drinks so that I can choose the most suitable meal.</p> <p>Acceptance Criteria:</p> <ul style="list-style-type: none"> Should be able to navigate to a drinks page from the navigation bar to add drinks to order Should be able to select quantity for each drink listed to add order Quantity selected must be ≥ 0 Confirm button at bottom of page 	
Priority	1
Size	4 story points
ID	US4
Name	Place Order
<p>User-Story Description:</p> <p>As a customer, I want to be able to place my order online so that I don't need to wait in line</p> <p>Acceptance Criteria:</p> <ul style="list-style-type: none"> Should be able to navigate to a view/confirm order page from navigation bar Order page will show unique order ID Order page should list items added to order Total price of order should be displayed on the order page Confirm button at bottom of page If an item selected is out of stock, or the quantity selected of an item exceeds stock levels, the customer is notified by an error message when attempting to confirm order Once an order is confirmed, inventory levels are adjusted accordingly 	
Priority	1
Size	4 story points
ID	US5
Name	Check Status
<p>User-Story Description:</p> <p>As a customer, I want to be able to check the status of my order so that I know when to collect my order</p>	

Acceptance Criteria: <ul style="list-style-type: none"> • Should be able to navigate to a 'check status' page through the navigation bar • Should be able to enter their order ID into a text box to check order status • If the ID entered does not exist, the customer is notified by an error message • If the ID is valid, the customer is notified whether or not their order is ready for collection 	
Priority	3
Size	2 story point

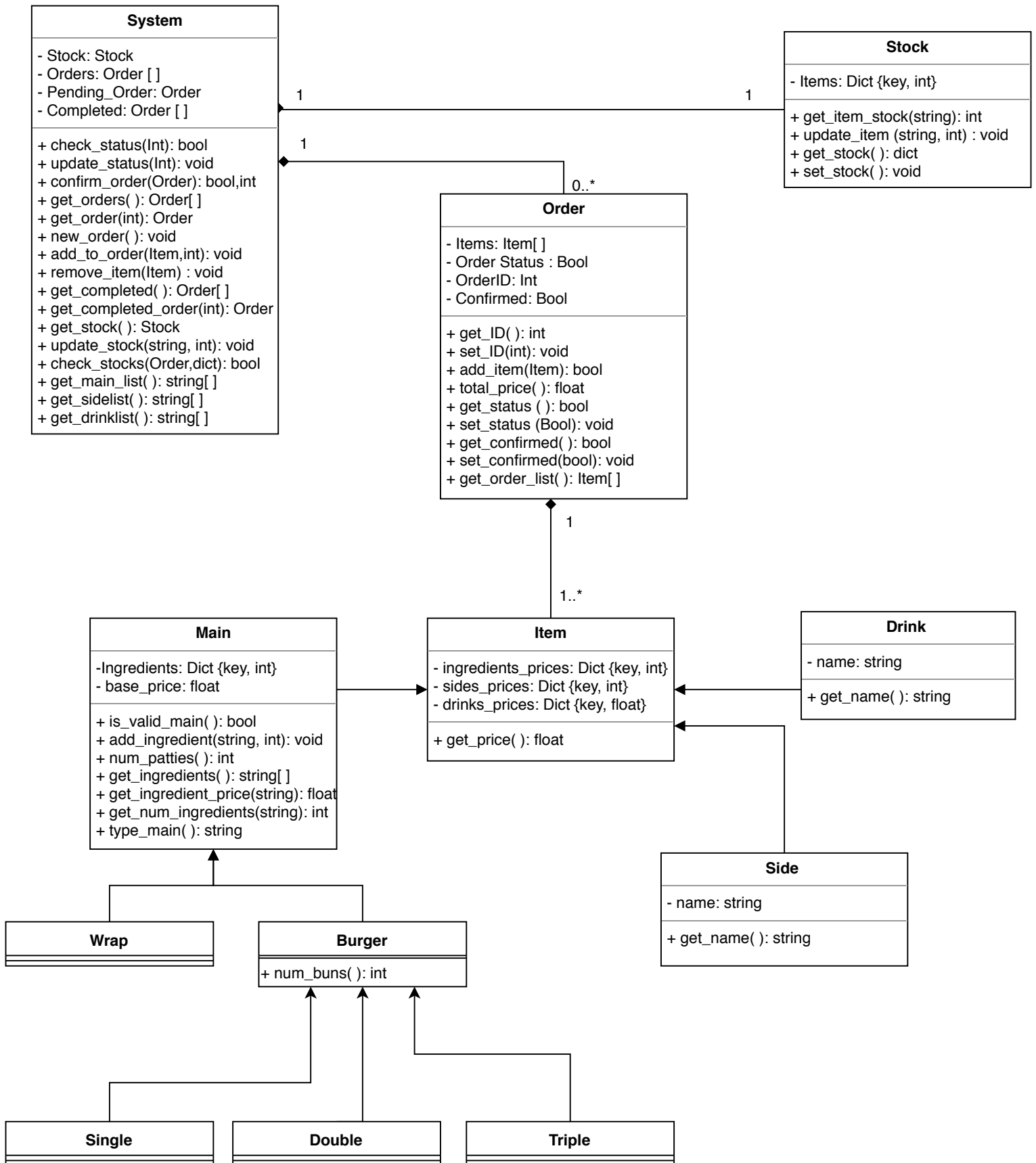
2. As a staff member, I should be able to view and update the status of an order.

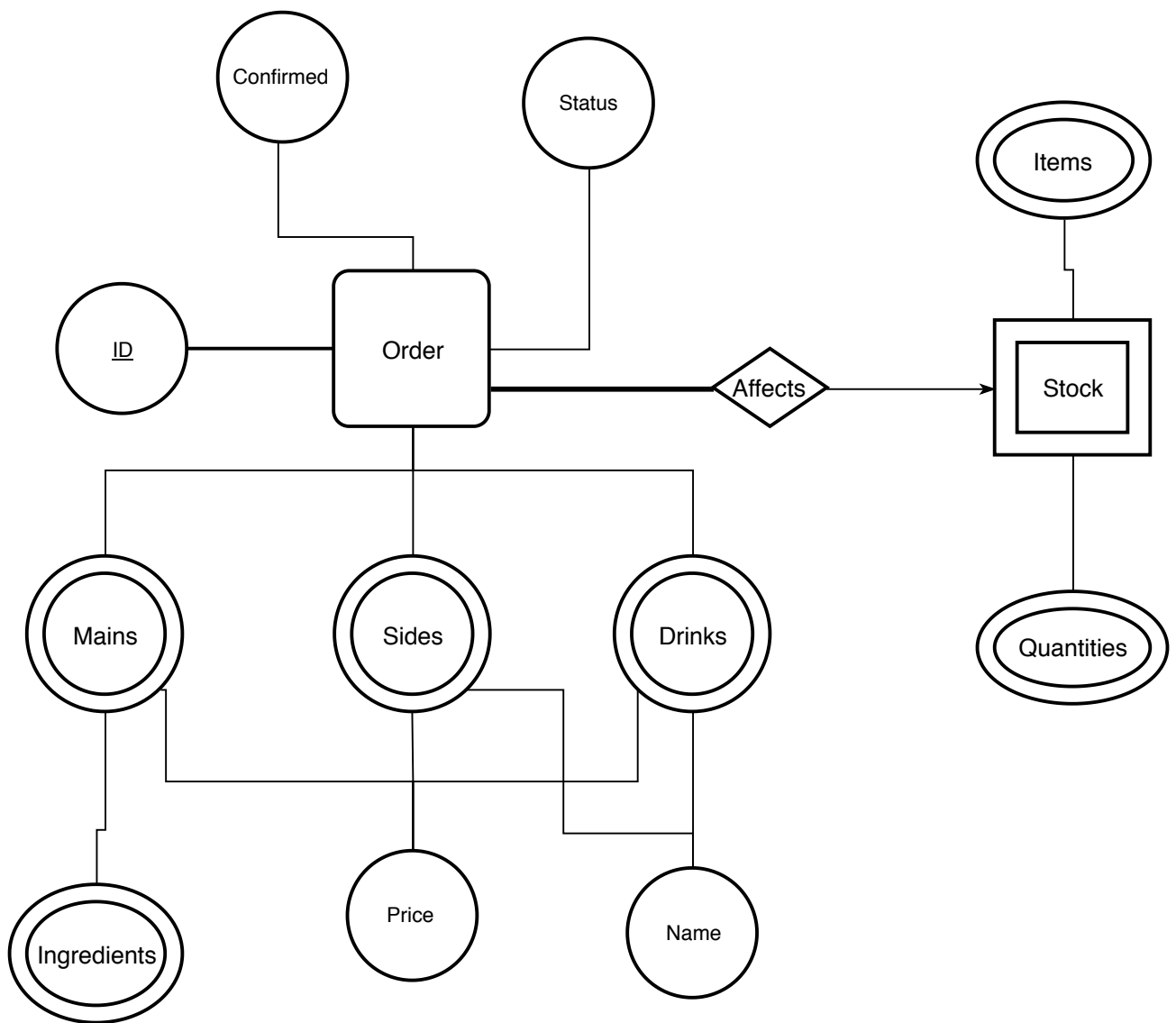
ID	US6
Name	Update Order Status
User-Story Description: As a staff member, I want to be able to update the order status so that I know which orders need to be completed.	
Acceptance Criteria: <ul style="list-style-type: none"> • Should be able to reach a 'staff portal' through the navigation bar • Should be able to reach an 'orders' page through the staff portal • Should be able to view pending order IDs in the form of a list • Each order ID will show the contents of the order • Should be able to mark a selected order as ready for pickup by the customer <ul style="list-style-type: none"> ○ This order will be cleared from the staff orders list 	
Priority	2
Size	4 story points

3. As a staff member, I should be able to update and view stock availability of each consumable item.

ID	US7
Name	View Stock
User-Story Description: As a staff member, I want to be able to view the stock availability of each item so that I am aware of stock availability.	
Acceptance Criteria: <ul style="list-style-type: none"> • Via the staff dashboard navigated to through the navigation bar, should be able to click a "View Stock" link and be redirected to a page listing availability of each item <ul style="list-style-type: none"> ○ Bottled drinks are stocked in either cans (375 mL) or bottles (600 mL) ○ Juice will be sold in small, medium, or large sizes; stocked in mL ○ Fries will need to be stocked by weight (in g) ○ Sundaes will need to be stocked in mL ○ All other items stocked in whole quantities 	

Priority	3
Size	3 story points
ID	US8
Name	Update Stock
<p>User-Story Description:</p> <p>As a staff member, I want to be able to update the stock availability of each item so I can track changes in stock.</p> <p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Via the staff portal, should be able to click an “Stock” link and be redirected to a page listing current availability of each item • Textboxes next to each item allow staff to enter quantity of additional stock to be added to existing stock • Once the “update” button is clicked, changes to stock availability are updated accordingly • If the quantity entered is invalid (ie. Less than or equal to 0, or not numerical), an error message is displayed when attempting to update the stock 	
Priority	2
Size	4 story points





Date	Key Decisions, Progress Made, Reflection on Decisions etc.	Tasks To Be Completed	Milestones Achieved	Obstacles
06/03/19	<ul style="list-style-type: none"> Meeting will involve writing out epic and user stories All work on project will be documented in a Google Doc for ease of collaboration and later converted to a pdf file to push to GitHub Anticipating requirement of 3 epic stories rather than 2 Decided 1 story point = 2 hours User story priority system will range from 1-5 where 1 is highest priority Epic stories completed User stories completed 	<ul style="list-style-type: none"> Complete acceptance criteria for each user story 		<ul style="list-style-type: none"> Settling on an appropriate 'benefit' for each user story was the largest issue
08/03/19	<ul style="list-style-type: none"> Meeting will involve completing acceptance criteria for user stories Decided to add an epic story for staff login User stories relating to staff login epic story completed Acceptance criteria for each user story completed 	<ul style="list-style-type: none"> Elliot to convert user stories to pdf and push to GitHub 	Milestone 1 Achieved	<ul style="list-style-type: none"> Uncertainty with how the 'bread' on a wrap is customised Uncertainty of the design of the website made writing 'Acceptance Criteria' a bit difficult Settling on appropriate 'benefits' still remains an issue
16/03/19	<ul style="list-style-type: none"> As per feedback from Milestone 1, US1 will be split into 4 separate user stories so workload may be more efficiently delegated later <ul style="list-style-type: none"> US1: "As a customer, I want to be able to place a custom order so I can select the most suitable meal" will be split into separate user stories for mains, sides, drinks, and another for placing the order Drew up a rough use-case model 			<ul style="list-style-type: none"> Finding times for meetings when all members are available is a challenge and often leads to us meeting at very late hours when we are tired and productivity levels are low

	<ul style="list-style-type: none"> Tutor has clarified that the bread on a wrap is not customisable 			
23/03/19	<ul style="list-style-type: none"> Completed CRC cards on whiteboard; typed up into Google Doc Completed UML class diagram based on CRC cards on whiteboard; subsequently drawn up on draw.io Upon reviewing the diagram, much of the information classes we were storing seemed redundant <ul style="list-style-type: none"> Ability for staff to login to the system was deemed unnecessary and would only cause extra work in the development phase hence decided to remove this from the product backlog. This means we no longer needed a Staff class 	<ul style="list-style-type: none"> Consult tutor about how wrap bread is customised Delegate backend development 		<ul style="list-style-type: none"> Uncertain about how to best model Burgers and Wraps; as separate classes or under the same class Not knowing how the backend relates to the frontend of the system caused many issues when trying to decide which functions each class required draw.io gave us some technical difficulties whilst using a trackpad
29/03/19	<ul style="list-style-type: none"> Reviewed CRC cards and changed how data will be stored in classes <ul style="list-style-type: none"> Dictionaries within classes will now store quantity and price of ingredient, rather than having each ingredient be an attribute of the class. This means getter/setter methods won't be required for each ingredient, increasing flexibility of system Upon review, Customer class is being removed from the system as it seemed 	<ul style="list-style-type: none"> Sherry to begin backend for Main class Elliot to begin backend for Side class Cuthbert to begin backend for Drink class Get feedback from Milestone 2 and update accordingly 	Milestone 2 (Part 1) Achieved	<ul style="list-style-type: none"> Still not fully certain about whether some functions are necessary due to ambiguity on how the backend and frontend of the system work together but reviewing the labs helped a bit Whether some associations in the UML class diagram should be composition or aggregation were uncertain due to the same reason as above

	<p>more efficient if order ID was simply stored in the Order object rather than in the Customer object since Order objects were already stored in the system</p> <ul style="list-style-type: none"> Updated interactions between classes to better reflect changes in stock level and price of each item <ul style="list-style-type: none"> Update price is now a function under System Update quantity is now a function under Main, Side, Drink Reviewing this week's/last week's lab as an example was very helpful in figuring out under which class functions should go UML class diagram updated to reflect changes in CRC cards 			
01/04/19	<ul style="list-style-type: none"> Taking into consideration feedback from Milestone 2; <ul style="list-style-type: none"> Drink and Side class will no longer store dictionaries of food items, each new drink/side will be represented by a new instance of a Drink/Side object Main will take on subclasses of SingleBurger, DoubleBurger, TripleBurger, Wrap Main, Drink, Side, will all be subclasses of a MenuItem class Most of these changes are to reduce repetitiveness in common functions 	<ul style="list-style-type: none"> Sherry to make changes to Main class according to feedback and write tests Elliott and Cuthbert to write tests for Side and Drink classes, respectively 		<ul style="list-style-type: none"> We had large issues with creating Git branches and pulling these branches on different machines as we had not been taught how to do this

	<ul style="list-style-type: none"> The Stock class will also now simply contain a dictionary to store all stock levels and no longer collaborate with the Main, Side, Drink classes Tutor also mentioned prices of items can be hardcoded into the system rather than taken in as parameters Completed 1.5 User Points for US3 			
02/04/19	<ul style="list-style-type: none"> Continued work on the mains and drinks classes (Mostly complete) Completed Drinks Class Elliot completed Sides class Completed 3 User Story Points for US1 and US2. 			
03/04/19	<ul style="list-style-type: none"> Tests for the Main class and its subclasses have been completed and bug fixes implemented Basic draft for Stock class also written 	<ul style="list-style-type: none"> Sherry to write tests for Stock class Order, System classes must be written; corresponding tests also must be written 		
04/04/19	<ul style="list-style-type: none"> Meeting will involve working together to complete Order and System classes, and respective tests Made changes to Main, Side, Drink by overriding the <code>__eq__</code> method so that the items could be easily searched for in the Order class Completed 3.5 story points for US6, US5, US4 	<ul style="list-style-type: none"> Checking stock and ID functions for System still to be written Tests for System must be written 		<ul style="list-style-type: none"> Throughout the process, following the methods/attributes we had initially planned out in the UML class diagram wasn't really working, because during the development phase, we consistently came up with new ideas as to how features could be better implemented, meaning the UML diagram needed a large overhaul to reflect these changes We found the specifications to be quite vague as they did not take into consideration multiple cases such as if a customer is able to make changes to an

				<p>order prior to confirmation, or if a customer may choose to never confirm their order, and waiting for replies on the forum to answer these questions took up a lot of time</p> <ul style="list-style-type: none"> At what point an order class should be instantiated was also hard to discern as we had not had much practice working with building the front-end of a system or seen an example of a front-end which would be as complex as the one we are implementing
05/04/19	<ul style="list-style-type: none"> Basic tests for System written Velocity chart drawn up 			<ul style="list-style-type: none"> Uncertain whether to raise customer exceptions and returning error messages vs simply returning error strings
07/04/19	<ul style="list-style-type: none"> System class completed Extra tests for System written User stories updated to reflect any changes Velocity chart updated 		Milestone 2 (Part 2) Achieved	
08/04/19	<ul style="list-style-type: none"> As per feedback from Milestone 2, exceptions raised in functions in System.py should not be caught in the respective functions but simply raised and try/excepted in the front end 	<ul style="list-style-type: none"> Organise a meeting time and delegate frontend tasks accordingly 		
12/04/19	<ul style="list-style-type: none"> Began work on Front End Outlined basic design for Front-End and delegated tasks to everyone 	<ul style="list-style-type: none"> Sherry to do Burger page, Wrap page, Homepage, Navigation bar Cuthbert to do Order Page, Check Status, Staff Portal and checkout. Elliot to do drinks, sides, staff orders and stock. 		

14/04/19	<ul style="list-style-type: none"> Navigation bar, homepage html completed, associated functions added to routes.py 			
16/04/19	<ul style="list-style-type: none"> Sides, drinks html pages completed , associated functions added to routes.py 			<ul style="list-style-type: none"> Had some issues trying to figure out how to avoid hardcoding names of sides, drinks in
17/04/19	<ul style="list-style-type: none"> Order page html completed, associated functions added to routes.py Burger, wrap html pages completed and associated 	<ul style="list-style-type: none"> New test cases based on user stories. 		
18/04/19	<ul style="list-style-type: none"> Updated Side, drinks html pages Staff's functions added to routes.py Extra html pages for confirming, displaying orders 	<ul style="list-style-type: none"> Start on ER Diagram 		
21/04/19	<ul style="list-style-type: none"> Completed test cases based on user stories. Completed ER Diagram 			
22/04/19	<ul style="list-style-type: none"> Final testing Debugging Completing workings software 	<ul style="list-style-type: none"> Final report Update Class Diagram Edit User Stories 	Milestone 3 Achieved	
26/04/19	<ul style="list-style-type: none"> Updates on User Stories Edited ER Diagram 			

Velocity Chart

