COMP1531 Assignment Report

# gourmet-v-a-p-o-r-w-a-v-e

Christopher Shi – z5165244

Melanie So – z5164488

William Chen – z5165219

Table of Contents

[gourmet-v-a-p-o-r-w-a-v-e 1](#_Toc7362418)

[FINAL USER STORIES 3](#_Toc7362419)

[CLASS DIAGRAM 8](#_Toc7362420)

[ENTITY-RELATIONSHIP (ER) DIAGRAM 9](#_Toc7362421)

[LOGBOOK 10](#_Toc7362422)

# FINAL USER STORIES

**EPIC STORY 1:** As a customer, I want to be able to **customise or select (a main), select (sides and drinks) and checkout** a meal order through the online self-service ordering application.

|  |  |
| --- | --- |
| **ID** | US1 |
| **Name** | Select or customise main order |
| **User Story Description:**   * As a customer I should be able to customise my main order of either a wrap or a burger so that I can create food that suits my taste.   **Acceptance Criteria:**   * From the ‘Create New Order’ page, a customer is able to click an ‘Add main to order’ button, which directs them to a page to begin the selection of their main. * A customer is able to select a base main for their meal, which is either a burger or a wrap, limited to:   + Single Burger   + Double Burger   + Triple Burger   + Wrap * Once they have selected their base main, the customer can choose to customise their main by either selecting ‘Yes’ or ‘No’ in a dropdown and clicking a ‘Select’ button. * If the customer does not wish to customise their main, they can select ‘No’, and are able to check the price of their standard base main by clicking a ‘Check Price’ button. * If the customer wishes to customise their main, they can select ‘Yes’ and are able to add up to 10 extra ingredients to their chosen base. * If the customer selects ‘Yes’, they can choose the type of bun they prefer, which includes sesame and brioche. * If the customer selects ‘Yes’, they can choose the type of patty they prefer, which includes beef, chicken, and vegetarian. * Only one type of bun and patty is allowed per main. If the customer selects multiple types of buns and/or patties, the bottom most selection will be used - all others will be ignored. * A customer should not be able to add less than zero of any ingredient. * Once the customer has completed their creation, they can view the net price of the main, which is calculated based on the chosen ingredients by clicking a ‘Check Price’ button. * A customer can add the main to their order by clicking a ‘Confirm Main’ button. | |
| **Priority** | High |
| **Size** | 8 story points (1 point = ½ day) |

|  |  |
| --- | --- |
| **ID** | US2 |
| **Name** | Select drinks and sides |
| **User Story Description:**   * As a customer I should be able to select from a collection of drinks and sides so that I have the option to enjoy extra food which complements my main.   **Acceptance Criteria:**   * From the ‘Create New Order’ page, a customer is able to click an ‘Add side to order’ button, which directs them to a page to begin the selection of their sides. * A customer is able to select the sides that they prefer, which include fries, nuggets, and sundaes, which are listed in a dropdown with options:   + Small Fries, Medium Fries, 3 pack nuggets, 6 pack nuggets, Small Sundae, Medium Sundae and Large Sundae - where the sundaes are available in both Strawberry and Chocolate. * A customer should be able to select the drinks they prefer, which include coca-cola, and orange juice, which are listed in a dropdown with options:   + Coke 375ml, Coke 600ml, OJ 250ml, OJ 450ml * A customer can individually check the price of any available side by pressing a ‘Check Price’ button. * Once a customer has pressed the ‘Check Price’ button for an individual side, they should be able to add it to their order by pressing a ‘Confirm Side’ button. * A customer should not be able to add less than zero of any side or drink. * If a customer does not wish to add any sides or drinks to their order, they can return to the ‘Create New Order’ page to checkout or add other items to their order. | |
| **Priority** | High |
| **Size** | 6 story points (1 point = ½ day) |

|  |  |
| --- | --- |
| **ID** | US3 |
| **Name** | Checkout through application |
| **User Story Description:**   * As a customer I should be able to checkout through the online self-service application so that it is convenient for me.   **Acceptance Criteria:**   * A customer can view the items in their order before checking out. * A customer can view the total computed price of the meal which includes the main and any chosen sides or drinks. * From the ‘Create New Order’ page, a customer should be able to checkout through the application once they have decided on their meal by clicking a ‘Go to checkout’ button. | |
| **Priority** | High |
| **Size** | 4 story points (1 point = ½ day) |

**EPIC STORY 2**: As a customer, I want to **receive an order ID,** and **be able to check the status of my order at any point** through the online self-service ordering application.

|  |  |
| --- | --- |
| **ID** | US4 |
| **Name** | Receive an order ID |
| **User Story Description:**   * As a customer I should be able to receive an order ID post checkout so that I can personally reference my order if required.   **Acceptance Criteria:**   * Upon clicking the ‘Go to checkout’ button in the ‘Create New Order’ page, a customer can view their unique Order ID. * The Order ID should not be a negative number. | |
| **Priority** | Medium |
| **Size** | 2 story points (1 point = ½ day) |

|  |  |
| --- | --- |
| **ID** | US5 |
| **Name** | Check status of order at any point |
| **User Story Description:**   * As a customer I should be able to check the status of my order at any point so that I am readily aware of when my meal is ready to be collected.   **Acceptance Criteria:**   * From the home page, a customer can click a ‘Track Status of Order’ button which redirects to a page with heading ‘Check Order Status’. * A customer is able to view the status of their order, which proceeds from ‘Customer is making’ to ‘Kitchen Received’ to ‘Ready for Pickup’. * A customer is able to input their order ID into a text field with label ‘Order ID’, and click a ‘Fetch Status’ button to retrieve their order status. * If an non-existent Order ID is inputted, an error ‘Order ID must be within bounds’ will be shown. * If an empty Order ID is inputted, an error ‘Order ID cannot be empty’ will be shown. | |
| **Priority** | Medium |
| **Size** | 2 story points (1 point = ½ day) |

**EPIC STORY 3:** As a member of staff, I should be able to **view only the current orders**, and **update their status** when necessary.

|  |  |
| --- | --- |
| **ID** | US6 |
| **Name** | View current orders |
| **User Story Description:**   * As a staff member, I should be able to view all orders so that I have know which meals to prepare.   **Acceptance Criteria:**   * From the ‘Admin System’ page, a staff member can click a ‘View Current Orders’ button * A staff member should be able to see a table of current orders with information including the order ID, the order status, total price, and items in order | |
| **Priority** | High |
| **Size** | 4 story points (1 point = ½ day) |

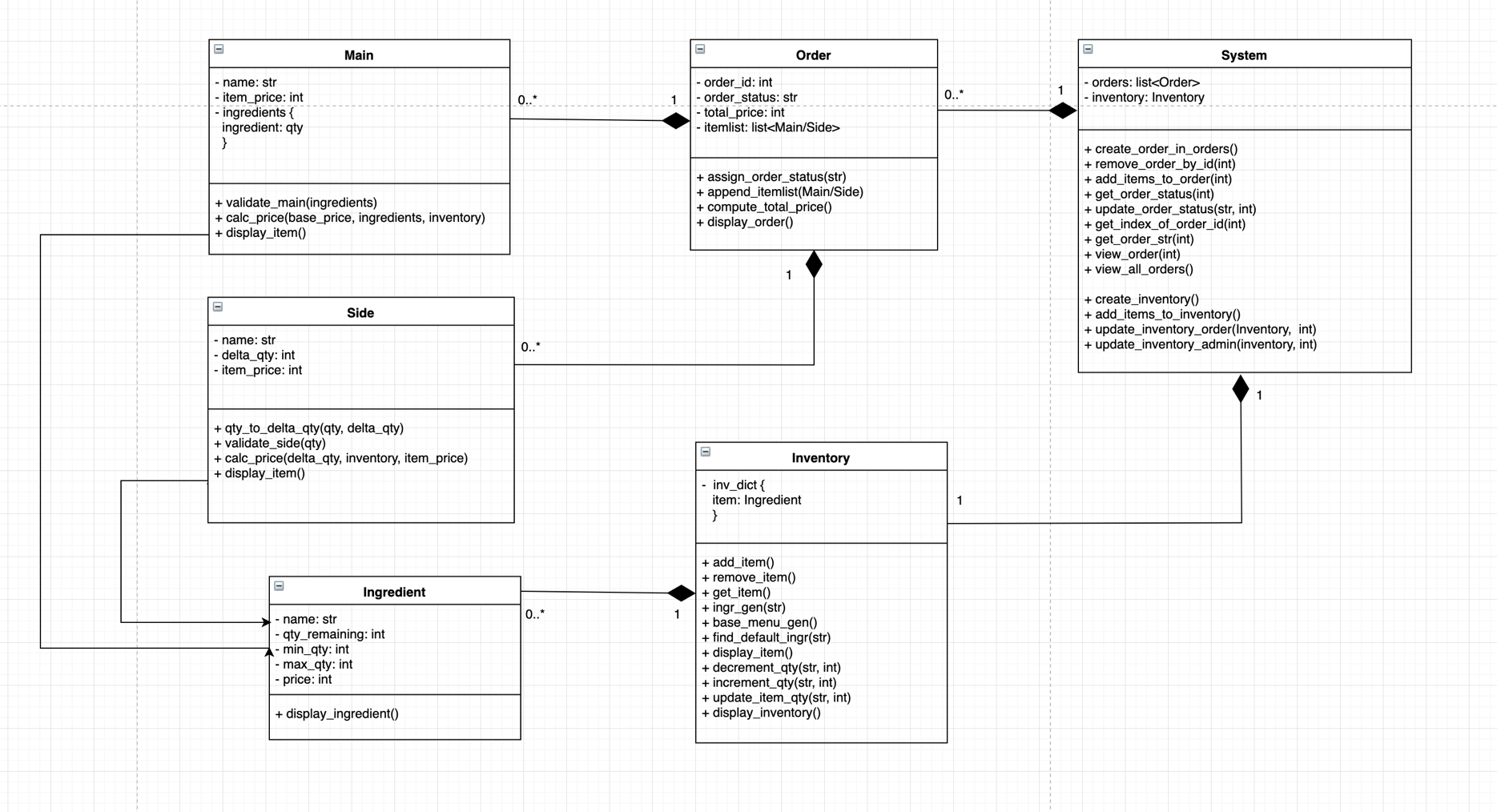
|  |  |
| --- | --- |
| **ID** | US7 |
| **Name** | Update status of orders when necessary |
| **User Story Description:**   * As a staff member, I should be able to update the status of orders so that I can keep track of orders and keep my kitchen uncluttered.   **Acceptance Criteria:**   * From the ‘Admin System’ page, a staff member can click a ‘View Current Orders’ button * A staff member should be able to update the status of the order to “Kitchen Received” once the customer has placed an order. * A staff member should be able to update the status of the order to “Ready for Pickup” once the order has been made. * A staff member is able to delete an order by clicking a ‘Delete from orders’ button. | |
| **Priority** | Medium |
| **Size** | 4 story points (1 point = ½ day) |

**EPIC STORY 4:** As a member of staff, I should be able to **view** and **update** the current inventory levels of the restaurant.

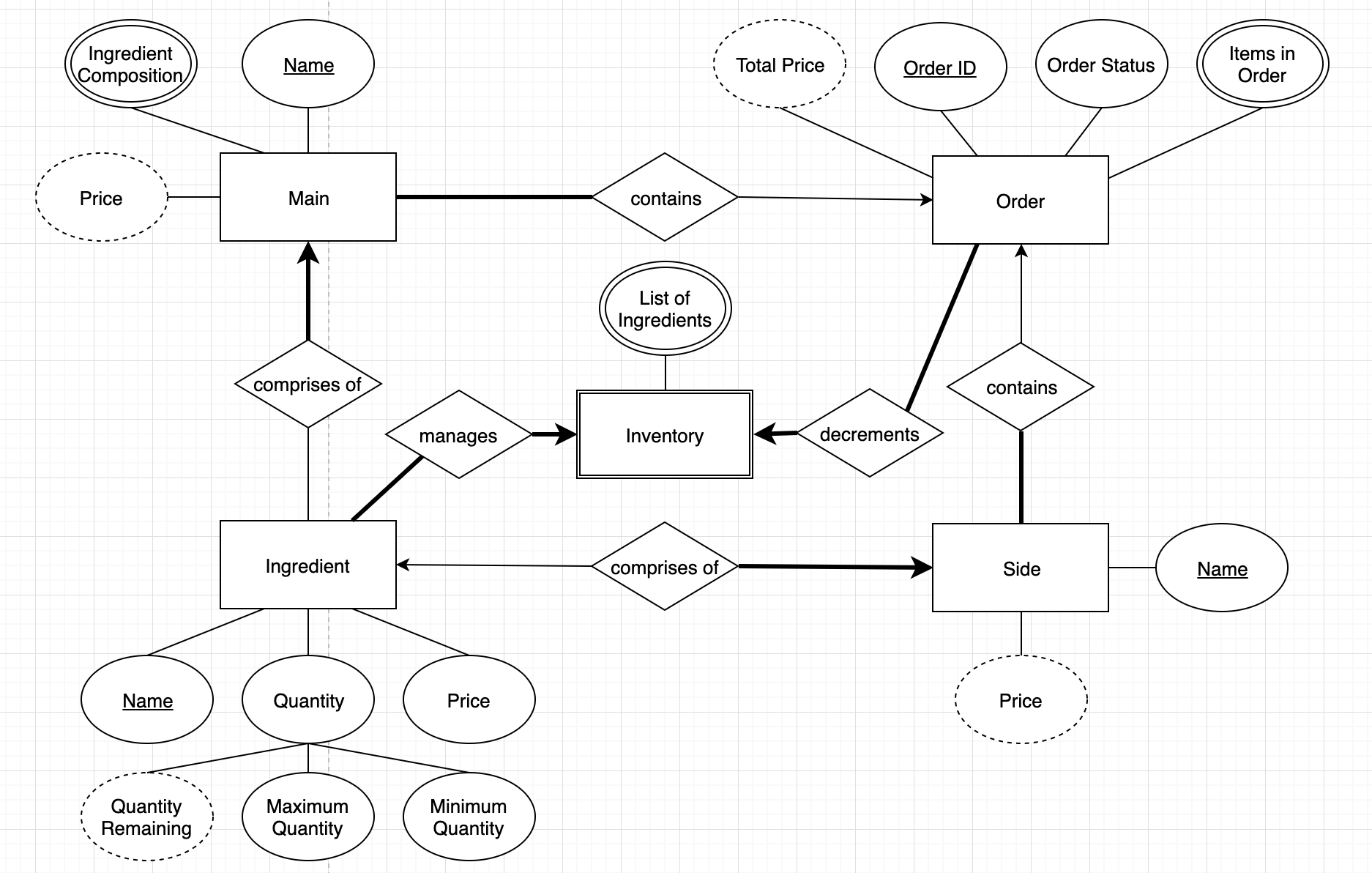
|  |  |
| --- | --- |
| **ID** | US8 |
| **Name** | View inventory levels |
| **User Story Description:**   * As a staff member, I should be able to view the current inventory levels of ingredients so that I am actively aware of when to replenish ingredients.   **Acceptance Criteria:**   * From the ‘Admin System’ page, a staff member can click a ‘View Inventory’ button * A staff member should be able to view the current inventory levels of in-store ingredients, specifically the remaining quantity of   + Nuggets and ingredients for burgers, wraps as a whole number.   + Different drinks in different volumes - 250ml, 375ml, 600ml   + Fries in weight (grams).   + Sundaes in sizes of - small, medium and large, and in two flavours: chocolate and strawberry. | |
| **Priority** | High |
| **Size** | 8 story points (1 point = ½ day) |

|  |  |
| --- | --- |
| **ID** | US9 |
| **Name** | Update inventory levels |
| **User Story Description:**   * As a staff member, I should be able to update the inventory levels of ingredients, so that I am aware that the system reflects accurate inventory levels.   **Acceptance Criteria:**   * A staff member should be able to update the current inventory levels of in-store ingredients, for both new stock or consumption, specifically the remaining quantity of:   + Nuggets and ingredients for burgers, wraps as a whole number.   + Different drinks in different volumes - 250ml, 375ml, 600ml   + Fries in weight.   + Sundaes in sizes of - small, medium and large, and in two flavours: chocolate and strawberry. * A staff member can input the number of each individual ingredient and click an ‘Update’ button, which will reload the page indicating updated inventory levels. | |
| **Priority** | High |
| **Size** | 4 story points (1 point = ½ day) |

# CLASS DIAGRAM



# ENTITY-RELATIONSHIP (ER) DIAGRAM



# LOGBOOK

**TUES, 5th of MARCH**

This was our team’s first in-person meeting and mostly consisted of deciding on communication and team work plans, reviewing the specification, and starting the first iteration of our user stories.

These are the general logistic-related aspects that were established in this meeting:

COMMUNICATION:

* Facebook messenger for general communication and written stand ups
* Skype/Facebook messenger calls for detailed meetings and key decision making
* In person meetings every Tuesday during and after our weekly labs

GENERAL TEAM WORK SETUP:

* Created the team repository
* Created a folder in Google Drive to store notes, user stories, diagrams

EXPECTATIONS:

* Being responsive to messages
* Being accountable for the work each member has been assigned
* Being transparent if a work is unable to be completed, or if any issues arise
* Availabilities were established
  + Chris has 3 days of work and full-time university
  + Mel has 4 days of work and full-time university
  + Will has 2 days of work and full-time university

During this meeting, we also began writing our user stories. This involved general brainstorming, and the main outcomes and decisions made were:

* The first epic story would surround the customer’s ability to customise their main, select their sides and drinks, and checkout a meal
  + Customisation of both burgers and wraps was chosen as the first user story. We had concerns about this at first since we thought the user story would be too large. In the end, we decided to go with this strategy as it was all one process for the customer.
* The second epic story would surround the customer’s ability to check the status of the order after checkout.
* The third epic story would surround staff members viewing the list of orders and updating their status
* The fourth epic story would surround staff members viewing and updating inventory levels

We faced some initial obstacles concerning interpretation of the specification. We spent a decent amount of time discussing the ‘checkout’ component of the application. We understood that the spec was up for reasonable interpretation, but ultimately were unsure if it included payment.

* After clarification, realised that payment was unnecessary for this project

At the end of this meeting, the scaffold for the Milestone was completed, epic stories were decided upon, user story descriptions were finished.

Responsibilities going forwards:

* Chris will own the user stories related to admin/staff
  + First draft to be completed by (8/03/18)
* Mel and Will will own the user stories related to the customer together
  + First draft to be completed by (8/03/18)

**FRI, 8th of MARCH**

We had our first call meeting over Facebook messenger. The purpose of this call was to finalise our user stories for the first milestone.

STAND-UP:

* Chris filled in the user stories, priorities, and sizes related to staff abilities. During the call, he will include the specific inventory items as part of the user stories.
* Mel and Will had filled in the user stories, priorities, and sizes related to customer abilities. They will work together during the call to finalise these and clarify error-conditions.

DECISIONS MADE:

* Each customer will only be able to create one main per order.
* Maximum and minimum quantities of buns and patties

During this call, we spent a lot of time discussing the particular language needed for correct user story descriptions. To help us remain on the right track, we colour coded our roles, goals, and benefits in red, green, and blue respectively.

MILESTONE – user stories were completed. Chris submitted this iteration.

**TUES, 12th of MARCH**

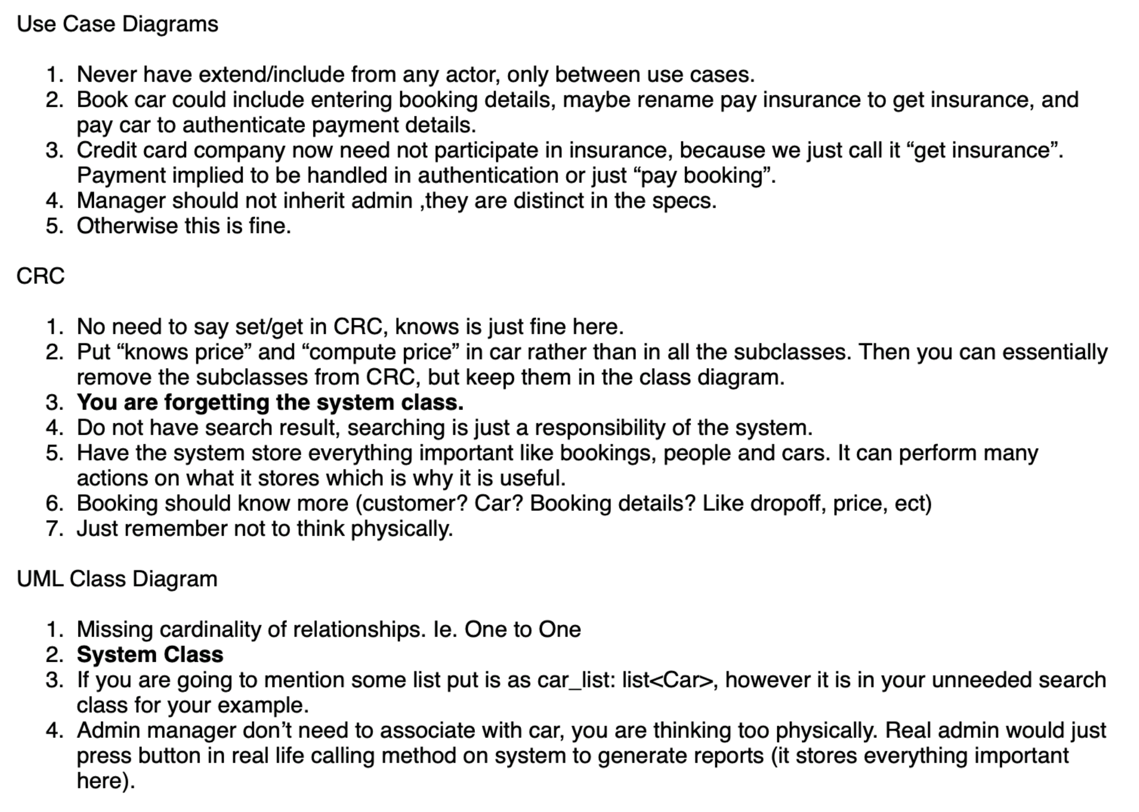
This was our first lab after submitting the user stories. The main feedback we received are as follows:

* Receiving an order ID might be too small to be a user story on its own
* Allowing one staff member to use the admin functionalities at a time will be too hard to implement

Responsibilities going forwards:

* Entire team to get familiar with CRC cards, use-case diagrams, class diagrams
* Will to update the user stories for the next iteration by 19/03/19

**TUES, 19th of MARCH**

We got feedback for our lab 3 and 4, which consisted of our attempts at writing CRC cards and use-case and class diagrams. The main feedback we received are as follows:

After receiving extensive feedback, the entire team reviewed existing knowledge of CRC cards, use case and UML class diagrams.

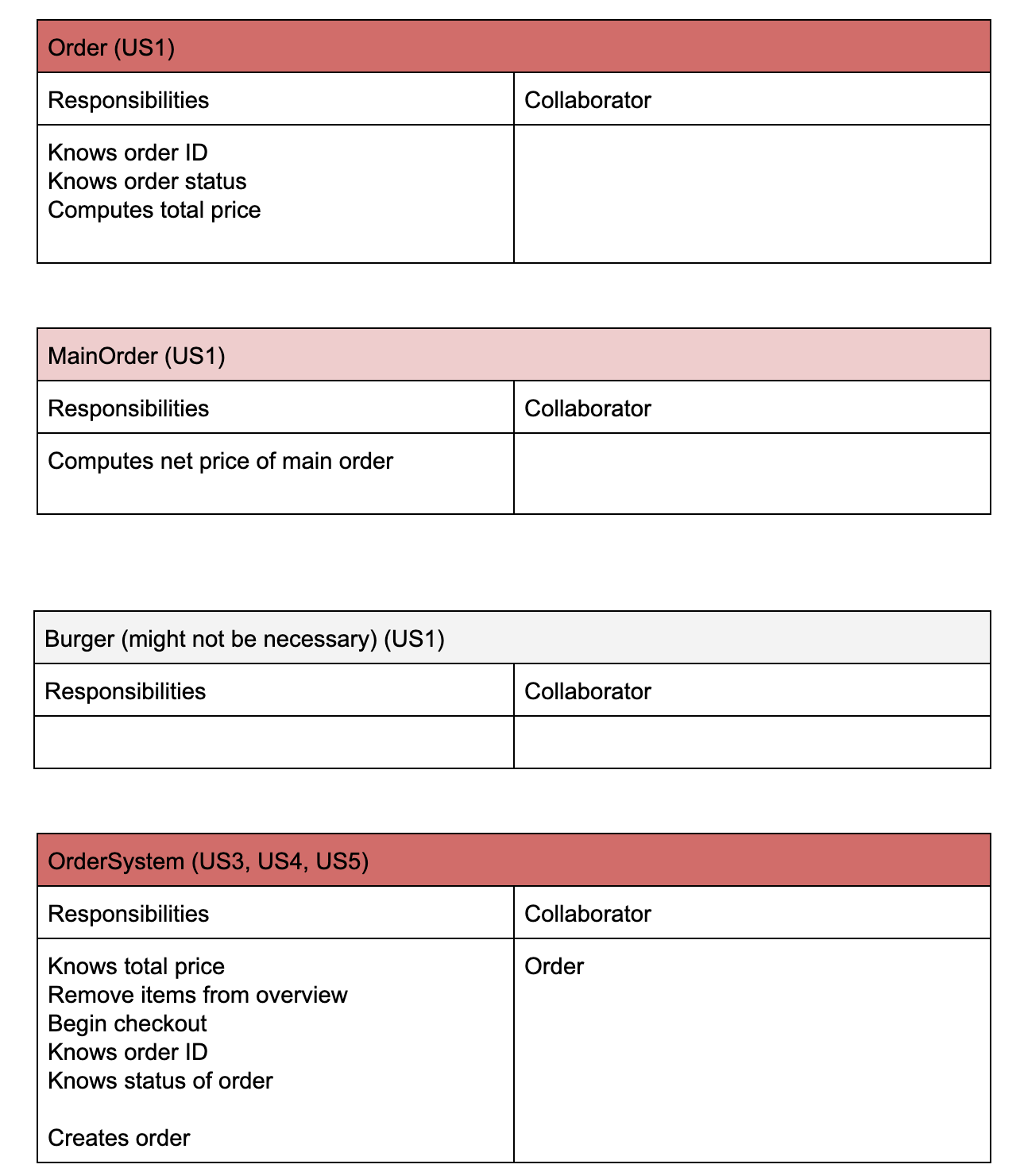
Set the next meeting for the 21/03/19.

**THURS, 21st of MARCH**

This was our second call meeting.

PURPOSE:

* Review the requirements for milestone 2
* Come to a consensus about the classes we’ll include in the diagram

Our first decision in this call was to start by making CRC cards in order to be able to create the UML class diagram with a strong foundation. These are some examples of the draft cards we made:

DECISIONS MADE:

* The classes that we would proceed with
  + Including their responsibilities and collaborators
* No need to include burger and wrap classes in the UML diagram

We also discussed team availabilities leading up to the next submission date:

* Will has a society induction on Sunday, and a 40% exam on Tuesday
* Melanie has a society induction on Saturday

Responsibilities going forwards:

* Decided to try and get most done before our lab on Tuesday for feedback from tutors and lab assistant
* Set next meeting for 25/03/19

**MON, 25th of MARCH**

During this meeting, we got started on the UML class diagram.

PURPOSE:

* To have a first draft of the UML class diagram before the next lab

STAND-UP:

* Will revised the use case diagram that we submitted for the week 3 & 4 lab to gain a better understanding of diagrams
* Similarly, Chris revised the UML diagram
* For this meeting, everyone is working together to draft up a UML diagram for milestone 2

For this first iteration, some decisions made were:

* Chris decided in the long term, it would be good to split up price and quantity, and also to introduce a total quantity class. This would be helpful for:
  + Inventory updating
  + Reducing the amount of main order instances
* Mel suggested breaking up Mains and Sides into their own classes, which would calculate the prices of the main and sides separately. This would be helpful for:
  + Separating the customisation business logic to selecting sides
  + Displaying the net price of the main

By the end of this meeting, we realised that a lot of the CRC cards we drew up were probably incorrect. We also discussed the impact of the changing specification in iteration 3 on our model.

Other things discussed were:

* The need to declare separate attributes for each ingredient
  + Ultimately unnecessary
* Maybe combining the price and quantity classes into one
  + Left it separate so we could get feedback for the original idea

We ultimately finished the first draft.

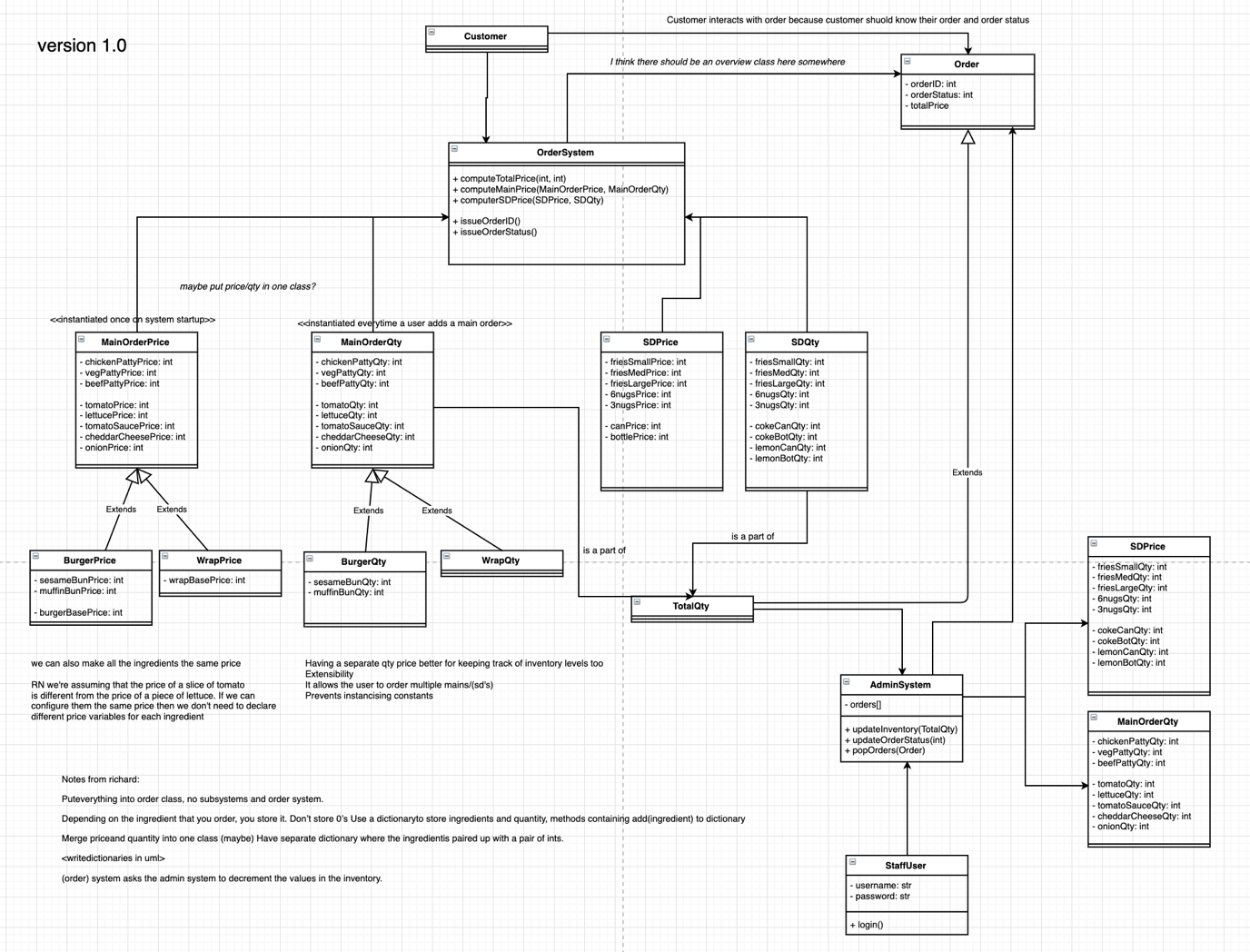
**TUES, 25th of MARCH**

Chris asked Richard for feedback on the first draft of our UML diagram. The feedback was basically that the diagram was wildly overengineered. The key points of feedback were to:

* Put everything into an order class and system -> subclasses were unnecessary
* Store ingredients depending on what is ordered. Don’t store zeros and use a dictionary to store ingredients and their quantities. Methods should be written to update these dictionaries
* Merge price and quantity into one class
* Write dictionaries in UML
* The order system asks the admin system to decrement the values in the inventory

After receiving this feedback from Richard, we set the next meeting for the day after (26/03/19) to discuss and make edits accordingly

MILESTONE - This was the first version of our UML diagram:



**WEDNESDAY, 27th of MARCH**

This was our in-call meeting after receiving feedback for the first draft of our UML diagram

PURPOSE:

* To reflect on the comments made on the first draft of the diagram, and to make changes accordingly

STAND-UP:

* The last thing everyone worked on was the first draft of the UML
* Chris to go through the comments with Mel and Will

REFLECTIONS:

* The first draft was unnecessarily complex
* Splitting classes into burger price, burger quantity, wrap price, wrap quantity was not needed, and probably incorrect since the methods defined in them can all be done within an order class
* The admin system would not need to manage side prices and main order quantities directly.
* A total quantity class is unnecessary, since we’ll be introducing a dictionary

This was our second meeting working together to build a UML diagram. We worked on about four iterations of the diagram.

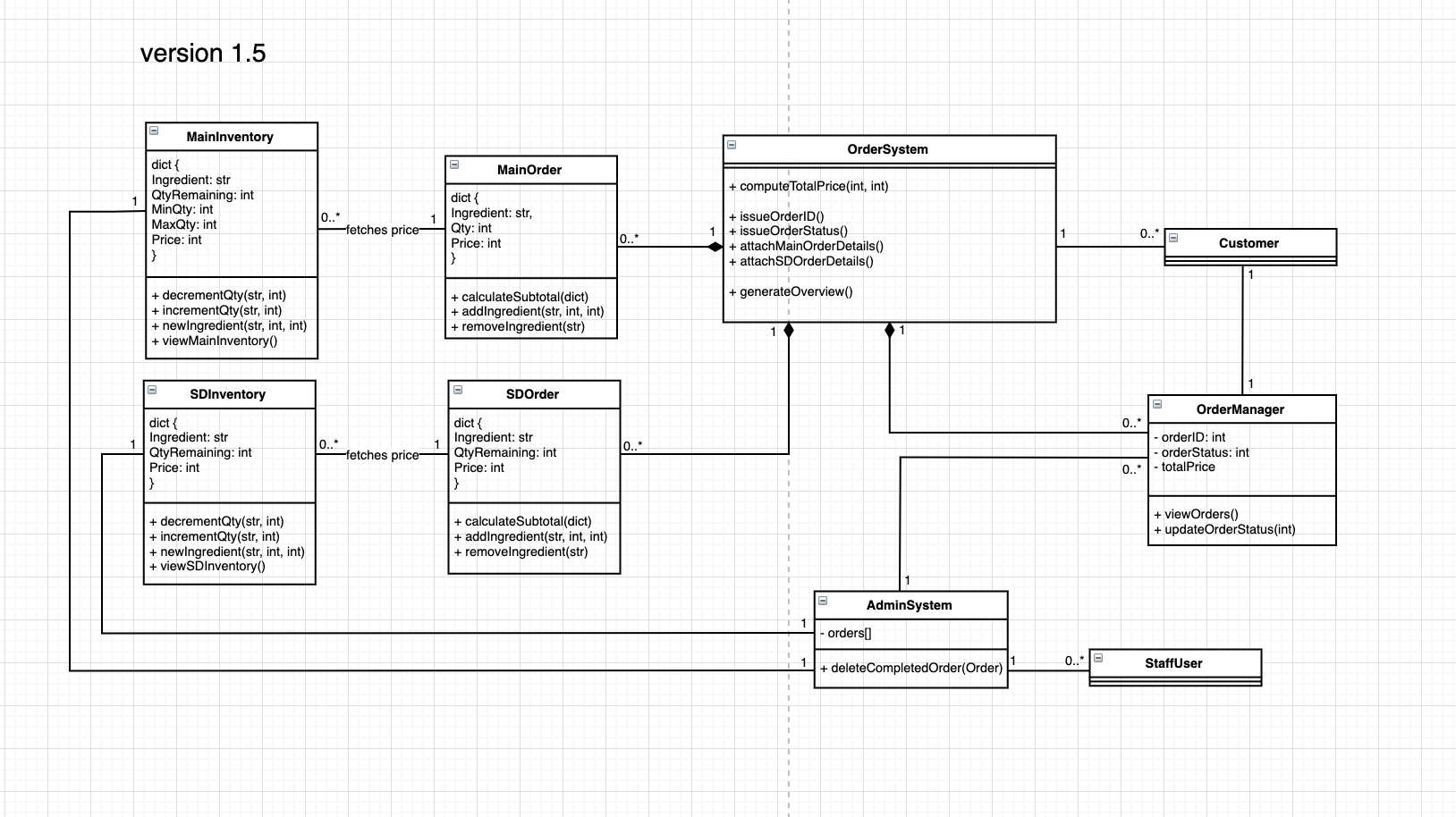
OBSTACLES:

* Using draw.io simultaneously proved to be slightly inefficient, as saving was manual. Sometimes, a lack of communication would result in multiple team members editing the same part of the diagram unknowingly.

Responsibilities going forwards:

* Set a meeting for 30/03/19 for final thoughts and the chance to make edits before submission

MILESTONE – version 1.5 of our UML diagram was the first diagram we all agreed upon after Richard’s feedback:

****

**SATURDAY, 30 of MARCH**

This was our last meeting before the submission date for milestone 2.

PURPOSE:

* To make final edits to the UML diagram

STAND-UP

* Everyone worked together on the last version of the UML diagram
* Mel wants to bring up some points that we didn’t consider earlier

We took this time for a final reflection on the UML diagram. These are the key decisions made:

* Combined the price and quantity classes to track a dictionary
* The Order class is instantiated by the system every time a user starts an order
* Removed the total quantity class
* Introduced idea of having a min and max quantity in the dictionary
* Removed the staff user and customer classes, since they don’t need to be managed

Some other things we discussed during this call were:

* Will brought up the issue of how to interpret the base prices of the mains
  + In the end, we had a different base price for each possible main

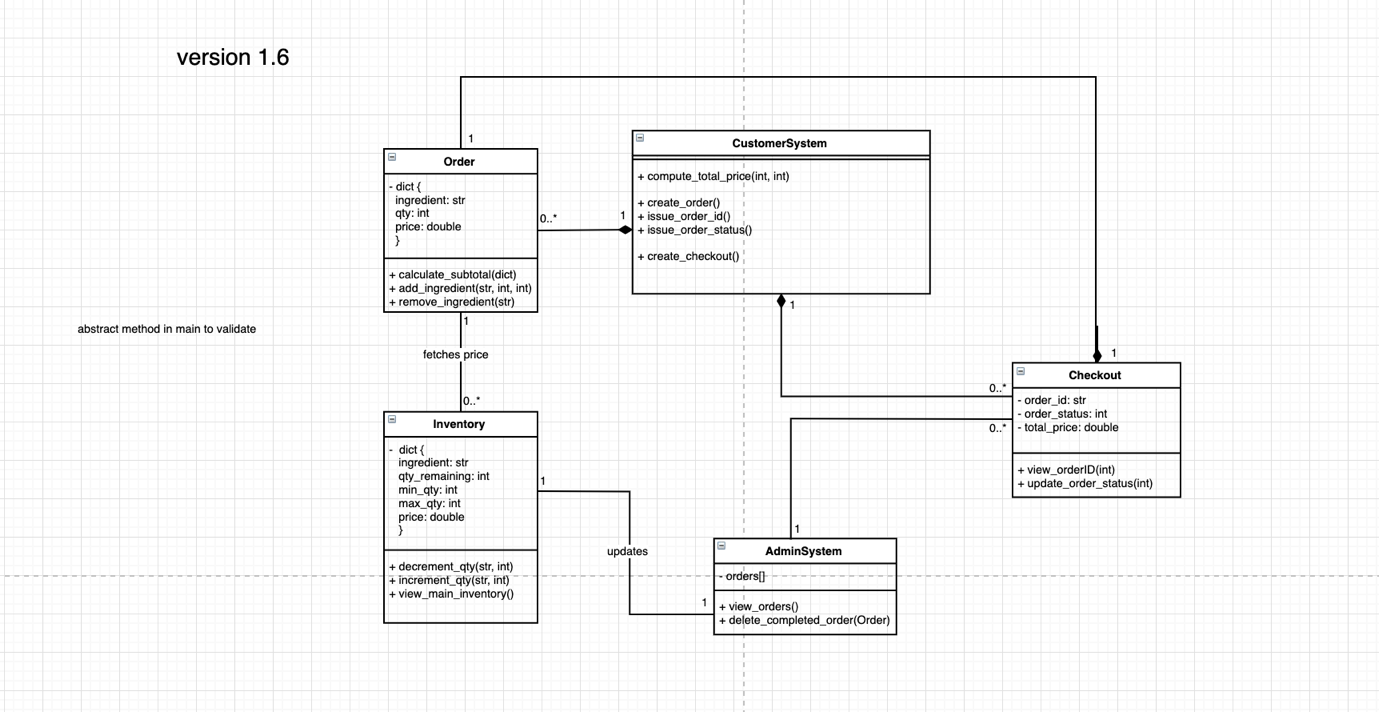
During this call, we also filled in the multiplicities and cardinalities.

OBSTACLES

* We found that having so many iterations of a single UML diagram became an obstacle. Soon, our whole team was getting confused about what needed to be included.
  + In the future, we discussed that everyone working individually on a diagram and then bringing them together to compare would have worked better, since more perspectives could have been discussed.

Responsibilities going forwards:

* Meet up after the next lab on 02/04/19 for UML feedback so we can begin coding

****MILESTONE – this is the final version of our UML diagram that we submitted to GitHub

**TUESDAY, 2nd of APRIL**

We received feedback on our submitted UML class diagram during this lab. The feedback points raised were:

* You can merge customer and admin system into one system that should contain a bunch of methods that calls a bunch of other methods
* Order object should be to compute its own price
* Each class should have methods to control their own attributes.
* Combine checkout and order into order
* Definitely have a separate class for main.
  + Think of how cars are validated differently in the lab exercises. The different types of mains should be validated similarly as well.

While our first draft of the UML was too complex, the one we submitted was ultimately missing a lot of considerations.

Responsibilities going forwards:

* Chris to revise the UML diagram
* Set a meeting for 03/04/19 to start coding

**WEDNESDAY, 3rd of APRIL**

This was our first meeting to discuss the coding portion of the assignment.

PURPOSE:

* To assign responsibilities and expectations for the next milestone with regards to the coding portion

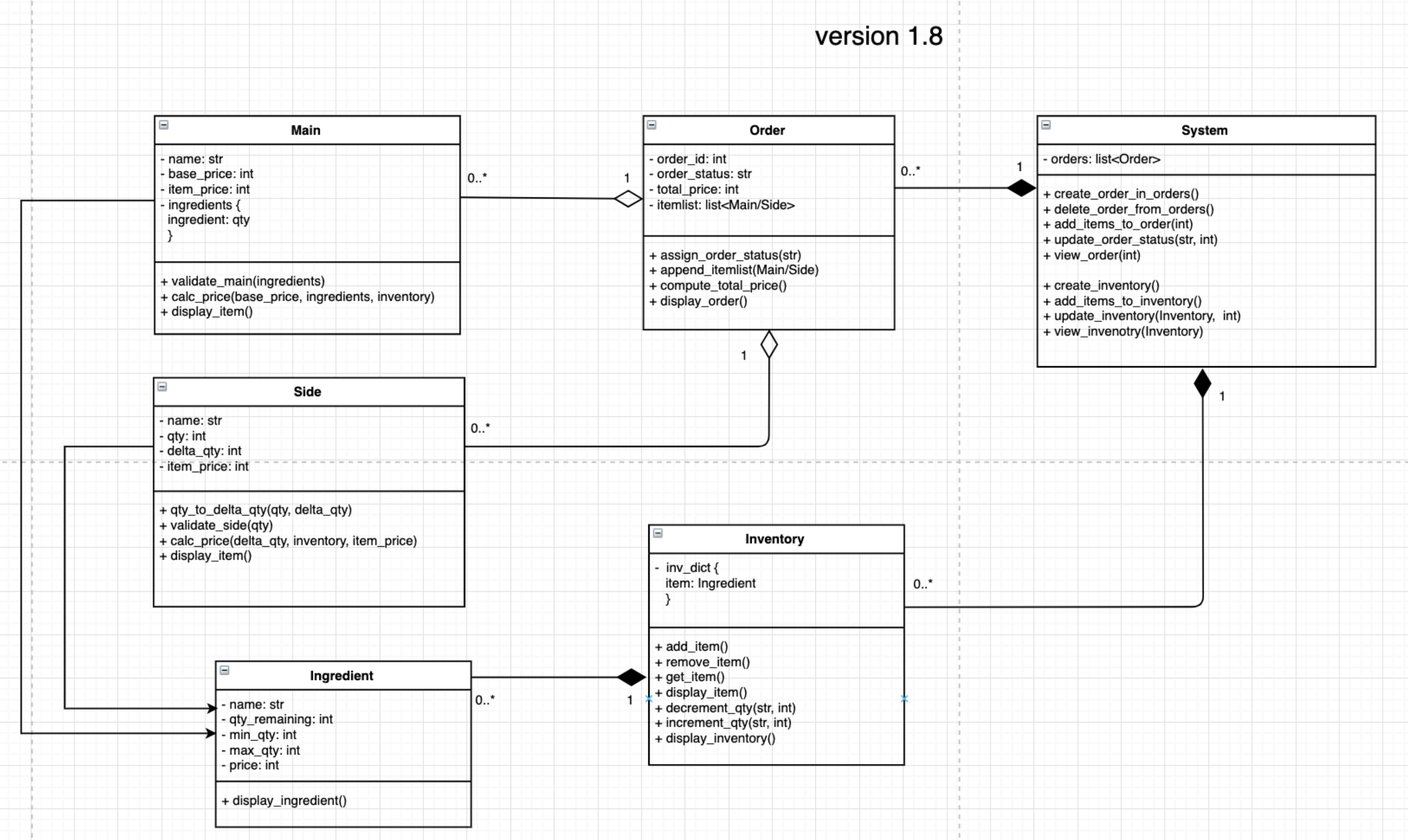
During this meeting, everyone revised their understanding of Git and we set up our branches. We decided that each member would have a branch named after them, which would be merged into master when iterations of code were completed.

Chris set up empty boilerplate files based on revised UML classes on his branch and merged them into master.

Responsibilities going forwards:

* Chris in charge of main and sides
* Mel in charge of order
* Will in charge of inventory
* Set a meeting for 04/04/19 to work together on the code

MILESTONE – this was the final UML diagram after feedback. This is the one we ultimately used to begin coding:



**THURSDAY, 4th of APRIL**

This was a meeting for everyone to work together while implementing their respective classes.

PURPOSE:

* To have a collaborative meeting where members can discuss and work on their technical implementation together

STAND-UP:

* Chris had started setting up the main.py file
* Mel had started on the order.py file
* Will had nearly finished implementing inventory.py

Will had made some key decisions related to the implementation of the inventory component of our project.

* He decided to introduce an Ingredient class in the same file
  + The purpose of this class was to instantiate an ingredient before placing it into inventory
* He created a setter for quantity remaining
* He also added in validations for inventory, including min and max quantities.

Mel began to write some tests to attempt to follow test driven development.

OBSTACLES

* We found that getting used to python as a language was hindering our performance slightly. We needed to take more time to research its behaviours and inbuilt functions
  + Although more time was taken, this was ultimately a good learning experience.

Responsibilities going forwards

* Set up an in-person meeting for 06/04/19 to finalise the assignment together

**SATURDAY 5th of APRIL**

This was a meeting to finalise all files that our system would have to import, and to begin writing tests.

STAND-UP:

* Mel had finished the first version of the order components
* Will had finished the first version of the inventory components
* Chris was nearly finished with the main and side components

FINAL CODE DESIGN RATIONALE:

Chris took some inspiration from the in-tutorial car booking project in terms of using arrays to keep track of objects and combined it with our current strategies in using dictionaries to keep track of ingredients.

* There are three main levels of classes for this iteration – Main/Sides, Orders, and the System.
* The system keeps track of all the orders that have been created using a list. It is a higher-level interface to the Orders, Mains and Sides. Orders will be managed via a First-In First-Out basis – with restaurant only being able to process one order, and able to move on only once that order has been completed. This implementation is not efficient but eliminates the need for parallelism-esque algorithms and concurrency management.
* Orders keeps track of all the mains/sides that have been assigned under it through a list.
* Main contains (initialised with an) ingredient dictionary containing all the ingredients in itself. The main class calculates its own price and validates that the dictionary of ingredients is valid.
* Side converts user specified details IE – 1x Six Pack of Nuggets, into a delta that can be adjusted onto inventory levels. The reason for all the subclasses is because of the different rules for conversion from user spec.

OBSTACLES

* After reflecting on the progress of our project, we realised that having vastly different timetables was a bit of an obstacle. It was definitely difficult to find a time where everyone could meet in person, and phone calls lasting into early hours of the morning were becoming more frequent.

MILESTONE – All files are working individually without the system class

**SUNDAY 6th of APRIL**

This was our last in-person meeting before submitting the coding portion of the assignment.

STAND-UP:

* Mel to finish her tests, finish the log book, and upload the updated UML and user stories
* Will to finish his tests
* Chris to implement system and finish tests

Everyone worked together to finalise pytest files and error files for exception handling.

Final edge cases considered and catered for:

* Orders with no buns or patties
* Empty orders
* Two different types of buns or patties
* Inputs that may include negatives or symbols

Responsibilities:

* Everyone to make sure their tests are working for the files they wrote
* Chris to merge a stable master into release
* Mel to update the docs

MILESTONE – Backend system working with tests

**FRIDAY 19th of APRIL**

This was our first sprint towards iteration 3.

PURPOSE:

* Assign and begin responsibilities for iteration 3
* Discuss spec and requirement changes

RESPONSIBILITIES:

* Chris to make changes to the backend code as needed to facilitate a standard burger
* Mel to work on the front-end for the admin system
* Will to work on the front-end for the customer/order system

OBSTACLES:

* Misread the submission date so only had the weekend to complete the coding component of this iteration

MILESTONE – Backend working as intended

**SUNDAY 21th of APRIL**

This was an in-person meeting to finalise the majority of the front-end of the assignment.

STAND-UP:

* Chris completed the back-end component of the project to function with changed requirements
* Will and Mel began the FE

RESPONSIBILITIES:

* Chris and Will to work together on the customer/order system front-end
* Mel to continue with the admin system front-end

KEY-DECISIONS:

* All extra-ingredient options to be dropdowns for simplicity
  + A decision tree structure for the option of customisation
* Divide main order and side order into separate pages
* Admin system to have 2 pages: View Inventory and View Current Orders
  + All functions will be handled within those 2 pages

**MONDAY 22th of APRIL**

This was our sprint before submitting the coding + testing component of the assignment.

STAND-UP:

* User stories related to the customer journey were complete
* Mel completed user story 7 (view inventory)

RESPONSIBILITIES:

* Mel to work on styling the website
* Chris and Will to start implementing testing

**THURSDAY 25th of APRIL**

PURPOSE:

* A quick meeting to divide responsibilities for the report component of this iteration

RESPONSIBILITIES:

* Chris to own the class diagram
* Will to own the ER diagram
* Mel to own the user stories and logbook

**SUNDAY 28th of APRIL**

STAND-UP:

* Chris had finalised the class diagram
* Mel had finalised the user stories, and will finalise the logbook and compile report
* Will had finalised the ER diagram

MILESTONE – submitted report