COMP1531 Group Project Milestone 3 Report

Team Gourmet Wanna One

|  |  |
| --- | --- |
| Derrick Foo | z3463395 |
| Rui Wang | unsure |
| Henry Ye | z5163982 |

User Stories

**Epic Story 1**: As a Customer, I want to be able to place online orders and check the status of them.

|  |  |
| --- | --- |
| ID | US1 |
| Name | Customising a main |
| User Story Description:  As a Customer, I want to be able to customise an order online, so that I can create a meal that suits my needs.  Acceptance Criteria:   * The Customer must first be able to select a burger or a wrap. Burger will be provided in three options: single, double or triple. * The customer once they have selected their choice of main, if the main was a burger they must be given the option of ordering a premade cheeseburger or a custom burger. * If the customer selects a premade cheeseburger, they will be taken to a summary screen with the order and be given the option to add to order. * If the customer on the other hand selects a custom burger/wrap, they must then be able to select the type of patty. Once the customer has successfully selected their patties, they must then be able to select from a variety of other ingredients (lettuce, tomato, cheddar cheese etc.). The customer should be allowed to finalise their mains without electing any of the Other Ingredients. * If the number of any of the constituent items exceeds the maximum allowable limit, the customer must be prompted to select an allowable amount. * If any item is out of stock the customer should be notified. * The customer should finally be able to add the order to their checkout. | |
| Priority | 1 |
| Size | 60 SP (1 SP = 2 hrs) |

|  |  |
| --- | --- |
| ID | US2 |
| Name | Adding sides and drinks |
| User Story Description:  As a Customer, I want to be able to optionally add sides and drinks to my order, so that I can complete my meal order.  Acceptance criteria:   * The customer should be given the option to add sides and drinks from the main page. From here, the customer will be redirected to a page with all the sides and drinks offered (nuggets, chips, coke, sundaes, orange juice). * The page should list all the drinks, sides and sundaes in their respective sides offered as a selectable item. * The customer must specify which side/drink they want, along with the quantity. * The customer should have the option to not order a side/drink with their main if they don’t want one. * Should the customer not want a main, the option to just order a side/drink should be provided. * If any item is out of stock the customer should be notified and the item should not be added to the order. | |
| Priority | 1 |
| Size | 60 SP (1 SP = 2 hrs) |

|  |  |
| --- | --- |
| ID | US 4 |
| Name | Order checkout |
| User Story Description:  As a customer, I want to be able to checkout my order, so that my order can be processed.  Acceptance Criteria:   * Once the customer has selected their mains, sides and drinks they should be able to add their order to the checkout. * The customer should be provided with a summary of the items to be purchased and the total price of the meal. * There should be a button on the screen, giving them the option to checkout their order. * Once the customer checks out, they should receive confirmation of the order being received and an online order id, along with a summary of the order and the process of the order. | |
| Priority | 2 |
| Size | 48 SP (1 SP = 2 hrs) |

|  |  |
| --- | --- |
| ID | US 5 |
| Name | Order status |
| User Story Description:  As a customer, I want to be able to check the status of my order, so that I know if it’s ready for collection.  Acceptance Criteria:   * Once an order is placed, the customer should be provided with an order-id. * There should be a link to view the status of the order on the website. * Here the customer should enter their id number and press submit, before the order, and the status of the order is displayed. * When the order is ready to collect, the status of the order should be changed to “order ready for collection”. * If the order isn’t ready yet, the status of the order should say “we are processing your order”. | |
| Priority | 3 |
| Size | 48 SP (1 SP = 2 hrs) |

**Epic Story 2:** As a staff member, I want to be able to service online orders.

|  |  |
| --- | --- |
| ID | US 6 |
| Name | View orders |
| User Story Description:  As a staff member, I want to be able to view online orders, so that I know what I need to prepare.  Acceptance criteria:   * The staff should be able to see all unfulfilled orders at any point in time, in the order that they were placed. * Once an order comes in, the staff orders menu should be updated, and the staff should be able to see what mains, sides and drinks have been ordered, with details of the customisations made and quantities ordered. * Each order should have its corresponding order-id displayed next to the meal details as well as the current status of the job. | |
| Priority | 1 |
| Size | 36 SP (1 SP = 2hrs) |

|  |  |
| --- | --- |
| ID | US 7 |
| Name | Update orders |
| User Story Description:  As a staff member, I want to be able to update online orders, so that I can inform customers as to the status of their order.  Acceptance criteria:   * As an order is completed, the staff should be able to press a “finished” button to notify the customer on the website via their order-id that their order is ready to be picked up. * The order should remain on the system until the customer has collected the order. * Once collected, there should be a “remove” button to remove the details of the order from the staff panel. After this, the order-id should not have a status. | |
| Priority | 2 |
| Size | 36 SP (1 SP = 2 hrs) |

Epic Story 3: As a staff member, I want to be able to maintain my inventory.

|  |  |
| --- | --- |
| ID | US 8 |
| Name | View inventory |
| User Story Description:  As a staff member, I want to be able to view my inventory, so that I know when I need to replenish my stock.  Acceptance criteria:   * The staff should always be able to view the remaining stock of any items or ingredients. * If an item has zero stock, it should also be displayed. | |
| Priority | 3 |
| Size | 36 SP (1 SP = 2 hrs) |

|  |  |
| --- | --- |
| ID | US 9 |
| Name | Update inventory |
| User Story Description:  As a staff member, I want to be able to update my inventory, so that I can inform customers what items are in stock.  Acceptance criteria:   * The staff should be able to update the inventory to reflect new stock levels in cases such as new stock arriving or perishables expiring. * As a customer’s order is completed, the inventory levels should change according to what was sold. * Every time an order is processed, the system must correctly deduct the right amount of inventory. * Should the staff member try to deduct too many ingredients, the system should raise an error informing them that the quantity can’t be less than zero. | |
| Priority | 2 |
| Size | 36 SP (1 SP = 2hrs) |

CLASS DIAGRAM

A close up of text on a white background

Description automatically generated

ER DESIGN

A picture containing text, map

Description automatically generated

COMP1531 Group Project – Logbook

Week 2

(28/02):

* Groups formed in tutorial groups
* Group members are Derrick Foo, Rui Wang and Henry Ye
* Group name: Team wanna one (Ruis favourite K-Pop Group)
* Teams Registered and Team Repository created in GitHub.
* Worked through Affordable Car Rentals User Stories (Lab02)
  + Progress can be checked on GitHub
* Group members tasked with reading the Assignment Specifications over the week, and to understand what is required of us in the first Milestone.

Week 3

(7/03):

* Worked on Requirements Analysis and Use-Case Diagrams for Affordable Rentals as a group.
* After Lab, we stayed behind in class to get started on our User Story Template for the Project.
* Problems we encountered:
  + User Story Priority – we had trouble ascertaining which user stories were more important than others, as well as the amount of story points to assign each user story.
  + Granularity Problems – our epic stories and user stories often overlapped, and we struggled to find meaningful ways to fix them at first.
* After our meeting, we had a list of user stories we were happy with and were each tasked with completing the acceptance criteria for the user stories.

|  |  |
| --- | --- |
| Name | User Story to complete |
| Derrick | Customising Main, Adding Sides + Drink, View Orders |
| Henry | Update Orders, Order Checkout, Order Status |
| Rui | View Inventory, Update Inventory, Pricing |

* Planned to meet at Uni on Saturday (10/3) to present our Acceptance Criteria and fix problems as a group.

(09/03):

* Meeting at UNSW
* Everyone had done their acceptance criteria, but to varying degrees of success
* Henry had quite vague acceptance criteria, whilst Derrick went overboard and wrote things that were not included within the specifications.
* The rest of the meeting was dedicated to understanding what really needed to be in an acceptance criterion and worked together to make our Acceptance criteria more detailed and more in line with the task specs. (this meant cutting stuff out of derricks and adding stuff into mine)
* Meeting was adjourned and our task at home was to compile and format our 3 docs together and edit to meet the milestone 1 requirements.

(10/3)

* Milestone 1 was due today, whole group proofread user stories, made sure there was no grammar/spelling errors
* Submitted Milestone 1

Week 4

(14/03)

* User Stories marked by Tutor and Lab Demo.
* Feedback:
  + Epic Stories mostly OK, no granularity issues.
  + Minor issues with first user story in that the <goal> and the <reason> were too similar. Resolved with the help of tutors.
  + Unnecessary User Story (Pricing), pricing elements of the project should be placed into their respective places in other user stories. Need to be fixed later.
  + Story Points were way too big (1 Day), need to find a more reasonable story point size (maybe like 2 hours?)
* During Lab, we decided that our responsibilities due before our next meetup is to familiarise ourselves with the lecture content, and to read and understand what needs to be done for Milestone 2.

A picture containing indoor, wall

Description automatically generated

CRC Cards

Week 5

(20/03)

* Google Hangouts to brainstorm ideas for our Class Diagram
* Used AffordableCarRentals Class Diagrams to help us with our inheritances and overall structure and flow of the diagram
* Constantly referred to the User Stories to make sure we stayed on track.
* Problems encountered:
  + Had problems with inheritances (what the classes <know>, and get from other classes)
  + This led to problems with our arrows and cardinality
  + Problems distinguishing if certain classes were classes (is patties a class or should It belong in the ingredients class?)
* Delegated Tasks: iron out problems and make sure Class Diagram is finished by the end of the week, to start working on the implantation soon (24/03)

(24/03)

* Through Conference Call, we reviewed the tasks set out to finish from last meeting
* Although we still had some problems finalising everything, we were behind schedule, so we needed to flesh out our implementation before coming back and finishing the diagram
* delegated and worked on creating classes in python:
  + Henry: Ingredients, Inventory
  + Derrick: Mains, Order, Staff Order Menu

Problems encountered:

* + Initially had trouble translating information from our class diagram into our implementation
  + Had confusing names for our classes that made it hard to conceptualise what we were coding (e.g. using vague titles like ingred, cust, and often getting confused between ingred vs inventory and main vs order)
* After meeting, delegated to fix our class diagram to better reflect information and make it clearer and more succinct as we implement it.

(25/03)

* Received news that deadlines for our implementation have been extended.
* Decided to stop current assigned work to focus on Class Diagrams, schedule a meeting at Uni after next week’s lab to complete it.

Week 6

(28/03)

* Stayed after Lab to complete Class Diagrams.
* As we had started to implement some of our classes, it made it easier to conceptualise the inheritances of our Classes in our Class Diagram.
* Also made it easier to figure out which arrows we should use.
  + For example, we debated which inheritance arrow we should use between Item and Inventory, but since we already coded a part of the Class Objects of Item and Inventory, we realised our inheritance in the code was how an Aggregation Relation is coded.
* Minor issues with readability of our class diagram, but quite happy with the contents of the diagram.
* Allows us to a lot more time into implementations and testing.
* Tasks to complete before our next conference call:
  + finish initialising classes
  + write out functions into the Class files (functions from the <methods> section)
* Tasks to complete in the next conference call:
  + make sure our Class Diagram is presentable
  + work through issues raised from initialising classes.

(30/03)

* Deliverables were achieved, but we realised we did not think about how the Classes would interact with each other and how we could access functions and objects from one class in other classes
* We underestimated the amount of work still left to do with completing the functions in our classes and overestimated the amount of time we thought we needed to complete the deliverables before this conference call.
* Therefore, we shifted our goals of the conference call to:
  + Continue working through our classes - Derrick
  + creating a basic test file to test our current functions to see how they ran – Derrick and Henry
* Problems encountered during conference call:
  + Realised the way we structured our classes meant it was more efficient to store our objects in dictionaries rather than lists, this meant changing most of our implementation which took a lot of time
  + Struggled with accessing fields within the dictionary, did not have much experience with using and implementing dictionaries. For example, our print function would work for a list, however for the dict, it would start to print <Object object at 76d873r (random address code)>
  + Test File was useful but quite basic, and didn’t do what we wanted to do for some exception cases
  + Because we changed the way our objects were stored, the way our functions were coded had to be completely revamped.
  + This meant completing our Classes to an acceptable degree took a lot longer than anticipated

(31/03)

* Conference Call, goals to be achieved:
  + Finalise our Class Diagram and Submit - everyone
  + Finalise implementation of Classes, and make sure all functions are working within our basic test file - everyone
* Small formatting tweaks to Class Diagram, no significant changes made
* Submitted, Milestone 2 Part 1 completed!
* Implementation of classes went quite smoothly, however, we were worried about the complexity of our code, as our Order Menu <online system> was a List of Dictionaries <order>, and that Dictionary was also composed of dictionaries<main>, <drink>, <sides>, with the <main> dictionary within the <order> dictionary being a list<buns>, <patties>.
  + Makes sense in our brains but may make things more difficult in the future.
  + [ {{sesame bun: 3, beefpatty: 2}, {tomato: 2}, {cheese: 3}}, {coke: 1}, {chips: 2}},

{{brioche bun: 4, chicken patty: 3}, {tomato: 1}, {cheese: 2}}, {sprite: 2}, {nuggets: 2}}] 🡪 pretty much what it looks like

Week 7

(1/04)

* Test Files to be completed today:
  + Basic testing of Functions within Classes:
    - Item - Derrick
    - Inventory - Henry

Recurring issues with debugging our code. The way we stored our code (list of dictionaries etc.) made it harder to resolve issues found.

(3/04)

* Test Files to be completed today:
  + Basic testing of Functions within Classes:
    - Order – Derrick
    - StaffOrderMenu - Henry
* Problems encountered:
  + Some minor issues with logic but solved quite quickly
  + Maybe testing was a little too basic, should look for more extensive testing for all the cases within the functions.
  + This meant we decided to go back to Class tests completed 2 days ago to increase the scope of testing in those files.
* Increased workload means we did not have time to finish all our testing files.

(4/04)

* Presented Class Diagrams today in class.
* Feedback:
  + The way we store our objects is quite different with how most people would, but tutor understood our reasoning and methods.
  + <methods> and <attributes> were fine
  + Only iffy thing was the number of things inherited from items.
  + We explained that we thought it was necessary from looking at the specs of the assignment
  + No changes needed to be made to our Class Diagram
* After Lab, group stayed behind to finish our test cases.

(6/04)

* Conference Call to get started on Exceptions Handling
* Tasks to do today:
  + Exceptions Handling for:
    - Item - Derrick
    - Inventory - Henry
* Problems:
  + Group struggled with understanding the concept of Exceptions handling and how it differed from normal pytesting.
  + Through the exceptions handling, we found numerous issues with our Class Objects, which we struggled to fix.

(7/04)

* Tasks to do today:
  + Exceptions Handling for:
    - Order, StaffOrderMenu – Derrick
    - Main – Henry
  + Finalising all our Code Files to commit

Week 8

(11/04)

* Presented code to tutors to mark
* Main feedback:
  + Use of Git bad, because we did most of our work through conference calls, we did not make the proper branches and commits that they needed to check even workload amongst group members
  + Class inheritance were different to how most groups would approach the task
  + Exceptions handling good.
* Next Task at hand was to learn and work on flask and frontend.
* Group Members tasked with learning flask through the Lab presented this week, and to watch and understand all of Hussein’s videos on flask.

(14/04)

* Convened through a conference call to build a shell of what our website would look like
* Built simple run.py with a couple of buttons, but none of backend linked to the website yet.
* Problems encountered:
  + Understanding of flask very vague, Hussein’s video a little basic, knew what to do but not how to apply it
* Main Take-aways:
  + Need a stronger grasp of working with Jinja and Flask in order to carry out our website to the fullest degree
  + Should learn the fundamentals and application by completing the lab with Car Rental services
  + Spent the remainder of the conference call working through our labs.

Week 9

(18/04)

* Rui, after not doing anything for the backend, has told us she would work hard to do most of the frontend to make up for it.
* Me and derrick graciously accepted, but we had already started on our frontend, so we decided to continue working a little bit on our website so we could offer help whenever Rui asked.
* Tasks:
  + Henry: learn flask, continue working on building website to help Rui
  + Derrick: learn html, continue working on building website to help Rui
  + Rui: Finish all the frontend.
* Problems encountered:
  + Every refresh appended a new order to our staff order menu, struggled to fix this.
  + Struggled to integrate inventory backend within our frontend, ran into many problems, we think the underlying issue is within our backend, but it is demoralising to go back through what we’ve done to try fix it.
* Talked to tutors after class, recommendations
  + Use redirects to help with form submissions
  + Much of the exceptions handling can take place in frontend coding
  + What we have so far is on the right track.

(21/04)

* After many attempts to contact Rui, she did not get back to us.
* This left us with very little to work off, and essentially had 1 day to get off our butts and work hard to get at least something presentable together for the assignment.
* XP Programming, we planned to work together very closely to make sure we keep each other productive and on track.
* Task to achieve:
  + Main Task: finishing “Creating an Order”, making sure that our website can at least do that without breaking on us
* Problems encountered:
  + Inventory always causing us problems, decided to deal with it when we start coding staff decrementing order.
  + Environment very draining, but our trust in our group member left us no choice.

Week 10

(22/04)

* Met at UNSW, to finish up the frontend for assignment.
* Responsibilities (Both Henry and Derick)
  + Finish up “Staff able to update inventory”
  + Make sure “Customer able to place an order”, “Customer able to track progress”, and “Staff able to view placed and clear orders” are correctly doing their jobs
  + Review code with Class Diagrams to make sure we have stuck to out Design
  + If we have time, deal with persistence.
  + If we have time, add in SUNDAES to the list of sides you could order.
* Problems
  + Persistence was too hard, we gave up
  + Our Website does not give too many options, but at its core it does technically achieve all the “requirements” set out in the specs.

(27/04)

* Responsibilities
  + Henry: finish ER Design, compile report to be submitted
  + Derrick: Review Class Diagram, User Stories
* Problems
  + Nothing, we were glad to get this assignment over and done with and we r proud of ourselves for this 2-man job.

VELOCITY CHARTS

* 1 SP = 2 Hours
* Our Story Points Actual included work done on tasks not directly attributed to achieving the Acceptance Criteria, e.g. Class Diagrams, ER Diagrams, Logbook
* Total Projected Story Points of Assignment: 360 Story Points
  + This was an estimate given at the beginning of the assignment, we have not edited it, but a more accurate estimation could have been given when more knowledge was gained.