Team Six Deliverable Three

Andrew Clifford (19798632) Anzac Morel (66261880) Connor Macdonald (41647584)
George Stephenson (97277741) Hamesh Ravji (43832772) Rchi Lugtu (89933448)
Taran Jennison (67420293)

October 14, 2019

Contents

1 Executive Summary

2 Business and System Context

3 Stakeholders

This section aims to outline all the stakeholders the project has, and their respective concerns and what the team can do to relieve them. This helps the team see the point of view of the potential users and adjust course to best suit them and keep the project on track. Below there is a table which outlines the types of stakeholders, their needs, possible impact, priority, and action required.
Types of stakeholders:

- SP: Sponsors. People who commissioned project to be completed, including the intended people using the application.
- IN: Internal stakeholders. Employees/managers creating the product.
- EX: External stakeholders. People not directly involved but affected by outcome.

ID: SP0
Priority: High
Impact: High
Persona: Front of house employee
Front of house employees will be working directly with customers and the order management part of the application.
Stakeholders concerns:
Ease of use and speed when selecting options or possibly doing slightly customised orders.
Solution:
Create a clear GUI without unnecessary information.

ID: SP1
Priority: High
Impact: Medium
Persona: Truck manager
Manages the employees in the truck. Will likely update menus and stock levels in the system. Possibly ordering more stock when necessary.
Stakeholders concerns:
Ease of use when updating items in the application. Clarity of information shown so that they can make informed decisions based on stock levels, sales, etc. They might also want the ability to export/import recipes/stock for use in multiple food trucks.
Solution:
Create an easy to use GUI for displaying information and adjusting stock levels/recipes/prices.

ID: SP2
Priority: Medium
Impact: Medium
Persona: Truck Owners
Truck owners who are not also truck managers will not necessarily use the software but will rely on their employees to use it.
Stakeholders concerns:
Truck owners will need the software to work reliably for their employees to ensure their business has little downtime.
Solution:
High coverage of automated tests to ensure high quality and stability.

ID: SP3
Priority: Medium
Impact: High
Persona: Stock managers
Stock managers will only be interacting with the management side of the system to view and change stock levels.
Stakeholders concerns:
Ease of use with the management system. Access to financials, and stock. If the system fails the stock management will likely become inaccessible.
Solution:
Create an easy to use GUI for displaying information and adjusting stock levels/recipes/price. High coverage of automated tests to ensure high quality and stability.

ID: SP4
Priority: Medium

Impact: High
Persona: Chef
The chef will be relying on the system to give them orders. Likely won't interact with the GUI directly.
Stakeholders concerns:
Ability to see orders easily whether by notes or screen. If by screen, then an ability to dismiss or mark as complete is necessary.
Solution:
Create an output after each order is finalized that can be sent to the chef and displayed or printed in a given form.

ID: EX0
Priority: Low
Impact: Medium
Persona: Event Managers
Stakeholders concerns:
Event managers want the food truck to ensure everyone gets their order so reliability is their priority.
Solution:
Ensure orders are as easy to manage and consistent.

ID: EX1
Priority: Low
Impact: Low
Persona: Council
Stakeholders concerns:
The council wants the food truck to abide by food standards.
Solution:
Add a feature which allows tracking of food expiry dates.

ID: EX2
Priority: Low
Impact: Low
Persona: Stock suppliers
Stakeholders concerns:
Reliable stock order requests.

ID: EX3
Priority: Low
Impact: Medium
Persona: Accountant
Stakeholders concerns:
Access to organised financials.
Solution:
Make the management side easy to read and simple to export.

ID: IN0
Priority: Medium
Impact: High
Persona: The team
The team working on the project
Stakeholders concerns:
Meeting required standards and creating a product that fills the needs of the clients.
Solution:
Frequent interaction with stakeholders and good project management.

4 Quality Requirements

5 Use Cases

6 Functional Requirements

7 Acceptance Testing

This section aims to outline all the acceptance tests that were implemented into the system. This helps the team see if the system meets requirement specifications. Below are all acceptance tests written in the Gherkin test format.

ID: UC1
Use case: Add money to the cash register
Given:
$200 in the cash register
When:
$50 is added
Then:
There is now $250 in the cash register

ID: UC2
Use case: Remove money from the cash register
Given:
$200 in the cash register
When:
$50 is removed
Then:
There is now $150 in the cash register

ID: UC3
Use case: Too much money is removed from the cash register
Given:
$30 in the cash register
When:
$50 is removed
Then:
Error is thrown

ID: UC4
Use case: Add an item to the current order
Given:
One burger in the current order
When:
Chips are added to the current order
Then:
The current order now contains one burger and one chips

ID: UC5
Use case: Remove an item from the current order
Given:
One burger and one chips in the current order
When:
Chips are removed from the current order
Then:
The current order now contains one burger

ID: UC6
Use case: Cancel the current order
Given:
The current order contains one burger and one chips
When:
The order is cancelled
Then:
The current order is now empty

ID: UC7
Use case: Refund the total cost of an order
Given:
The cost of the order that is to be refunded was $15 and there is $50 in the cash register

When:
$15 is removed from the cash register
Then:
$35 in the cash register

ID: UC8
Use case: Customise a menu item
Given:
The current order contains one burger
When:
Tomatoes are removed
Then:
Tomatoes are removed and the chefs order and receipt reflect this

ID: UC9
Use case: Add a new menu to the system
Given:
There are zero menus in the system
When:
New menu is added
Then:
There is one menu in the system

ID: UC10
Use case: Add an item to an existing menu
Given:
A menu contains zero items
When:
Burger is added to the menu
Then:
Menu contains one item, Burger

ID: UC11
Use case: Remove an item from an existing menu
Given:
A menu contains one item, Burger
When:
Burger is removed from the menu
Then:
Menu contains zero items

ID: UC12
Use case: Edit an item
Given:
Burger doesn't contain tomatoes
When:
Tomatoes added to burger
Then:
Burger contains tomatoes

ID: UC13
Use case: Add a recipe to a menu item
Given:
Burger doesn't have a recipe
When:
Recipe is added to burger
Then:
Burger now has an associated recipe

ID: UC14
Use case: View the recipe for a menu item
Given:
Recipe for a burger needs to be viewed

When:
Burger selected AND View recipe selected
Then:
Recipe for burger displayed

ID: UC15
Use case: Add stock items
Given:
5 buns are currently in stock
When:
10 buns are added to stock
Then:
15 buns are currently in stock

ID: UC16
Use case: Update stock when stock is used
Given:
2 chips are currently in stock
When:
Chips are ordered
Then:
1 chips are currently in stock

ID: UC17
Use case: Item with no stock is ordered
Given:
0 chips are currently in stock
When:
Chips are ordered
Then:
Error is thrown

ID: UC18
Use case: View available stock
Given:
Levels of stock need to be viewed
When:
Stock screen is opened
Then:
Available stock is displayed

ID: UC19
Use case: View daily sales
Given:
Daily sales need to be viewed
When:
Past orders is selected
Then:
Past orders are displayed

ID: UC20
Use case: Generate sales report
Given:
Sales report is needed
When:
Generate sales report is selected
Then:
Sales report containing financial information is generated

ID: UC21
Use case: Change price of item
Given:
Price of burger is $10

When:
Price changed to $12
Then:
Price of burger is now $12

ID: UC22
Use case: Save menus to external file
Given:
Menus need to be exported
When:
Save menus to external file is selected
Then:
Menus.csv saved in local directory

ID: UC23
Use case: Load menus from an external file
Given:
Menus need to be imported
When:
Load new menu is selected
Then:
File containing menus is loaded in the correct format

ID: UC24
Use case: View sales from previous days
Given:
Previous days sales need to be viewed
When:
Past orders is selected AND date is changed to 12/10/2019
Then:
Orders made on 12/10/2019 are displayed

ID: UC25
Use case: Confirming an order
Given:
Current order contains one burger and one chips, total cost is $15
When:
Order is selected and stock is available to create order items
Then:
Chef's order and receipt are printed and $15 added to register

8 Deployment Model

9 Data Modelling

10 Technical Design

11 GUI Prototypes

Hello i am gui

Table 1: Summary of exposures to different risks that could occur during development

Team Risks				
ID	Description	Likelihood %	Impact 1 - 10	Exposure
R-01	Team Conflict	10%	1	0.1
R-02	Unfamiliar Development Tools	20%	2	0.4
R-03	Unfamiliar APIs/ libraries, various programming skill levels	90%	5	4.5
R-04	Miscommunication with lecturers	80%	5	4
R-05	Loss of data, problem with import of data	20%	6	1.2
R-06	Product does not agree with stakeholders expectations	90%	10	9
R-07	Code written by individual team members not readable by others	70%	4	2.8
R-08	GitLab, Google Drive, etc. become unavailable	2%	7	0.14
R-09	No Internet	5%	6	0.3
R-10	Bug in software e.g. bug says something gluten free when actually it isn't	90%	10	9

Table 2: Summary of consequences to different risks that could occur during development

ID	Consequences	Justification˜of
R-01	Reduced Productivity	Team has rules
R-02	Reduced Productivity as time spent learning how to use dev tools	Members of the
R-03	Some members limited to certain tasks, may mean some membes have to do more work than others	This is the tean
R-04	Doing tasks incorrectly and will have to redo or get a bad mark	The team has v
R-05	Have to re-complete work / manually import	All members of
R-06	Software won't sell (fake world) / bad mark from lecturers (real world)	It is very unlike
R-07	Reduced Productivity / Code has to be re-written	With new tools
R-08	Reduced Productivity, Can't commit new changes, may have to switch platforms	These services a
R-09	Can't commit changes to GitLab	Internet is prov
R-10	People could get sick	It is very unlike

12 Risks

12 Risks

The risk assessment module analyzes a number of different risks that both the team as well as the operator (user of the software) must be aware of during development and use of the software. Each risk is analyzed by multiplying its likelihood of occurring by the impact of the consequences on the group/user. This allows (low-likelihood, high impact) risks to be compared to (high-likelihood, low impact) risks. Most importantly, the last column of the table indicates how the risk can be avoided altogether, so this table should be referenced regularly.

12.1 Team Risks

12.2 User Risks

12.3 Risks Discussion - talk about what risks we have encountered and how we prevented / mitigated their effects, and justify why some values changed

Based on the feedback from the 1st deliverable and the 2nd deliverable it was clear that the risks section was not as extensive as it should have been, and some of the values were not correct e.g. We had the likelihood of team members being unfamiliar with libraries at 30% when really it should have been at 90%. In hindsight, we should have had more than one person deciding on the values for the risk assessment module as it resulted in biased and less thought through values. However time constraints near the end of the deliverable didnt allow for this. Time management was something that definitely held our grade back in the 1st and 2nd deliverable and this is a clear example of that.

For deliverable 2 we changed some of the likelihood values as you noticed and added a 'Justification of liklihood percentages' column too. Below are some examples of some of the risks we actually encountered and how they affected the development of the project.

R-03 - Unfamiliar libraries: This became clear to us very early on in deliverable 2 when using libraries such as JavaFX. Not many members of the group were familiar with it to begin with, even after having completed the JavaFX Lab. This hindered development in some areas where basic GUI functionality actually took a lot longer than expected to get up and running without any bugs.

Table 3: Summary of exposures to different risks that could occur during use of the software

User Risks				
ID	Description	Likelihood˜%	Impact˜1 - 10	Ex
R-11	Human error (misuse of software)	80%	9	7.2
R-12	Program freezes while processing customer's orders	20%	10	2
R-13	Screen showing the cooks what orders to make is inconsistent with actual order	20%	10	2

Table 4: Summary of consequences to different risks that could occur during use of the software

ID	Consequences	Justification˜of likelihood˜percentages
R-11	People could get sick	It is very likely that a user makes a mistake as mistakes happen frequentl
R-12	Angry customers lines get long, lose order	There is a low chance that the system will have a bug that will crash the
R-13	Angry customers	There is a low chance that the system will have a bug with such an integ

R-02 - Unfamiliar development tools: Most members of the team had only used Eclipse from SENG201 for Java projects. Switching over from Eclipse to IntelliJ was hard for some members of the team as the Project and Module SDK settings were playing up, however once we got it working there were no more problems and we concluded that IntelliJ is a lot better than Eclipse. SceneBuilder was also very new for most people, Taran did a lot of the design work for our GUI, so he had to learn how to use it by himself but he got the hang of it pretty quickly and produced an appealing GUI. We used Google Drive to store all of our design documentation as everyone was familiar with it, however we eventually had to switch over to LaTeX which was a new development tool for all of us. As this switch happened towards the end of deliverable 3, in hindsight we should have used LaTeX from the beginning as it is better than Google Drive.

R-08 GitLab, Google Drive, etc. become unavailable: We had the likelihood of this risk at 2%, as it seemed so unlikely as the development tools we were using such as GitLab and Google Drive are run by large companies. We were proven wrong when Google Drive crashed on us. Our design doc was 60 pages and when we had seven people trying to edit it at once, it crashed. Hence we switched our design documentation over to LaTeX which was much more friendly and has much better formatting tools. Connor's laptop also crashed two days before deliverable 2 was due. This was unfortunate, however we mitigated its consequences by making sure we always met where there was a lab computer available for Connor to work on. Hamesh also had a spare laptop that he kindly lended to Connor when we had to meet where there were no lab machines.

13 Testing Protocol

1	RowCol1	Row1Col2
2	Row2Col1	Row2Col2
3	Row3Col1	Row3Col2

Table 5: First Table

14 Project Timeline

Deliverable 1:
The first week of the deliverable was spent getting to know members of the team, establishing a team policy and setting up tools that would be used throughout the project (design document, trello board etc). After this initial phase had been completed a meeting was held to create and distribute tasks. This was done by creating a trello card for each task that needed to be complete. Then each group member assigned them selves to a card or cards and they were to complete the associated task before the next meeting. This same method was used for the entire first deliverable and with continual updating of tasks each week meant that the team was able to complete all tasks required without a large push close to the deadline. One issue with this method is that with team meetings being used primarily for task distribution all tasks were completed by individuals. The problem with this is that most if not all of the tasks required collaboration and therefore the quality of our deliverable suffered as a result.

Deliverable 2:
No work was completed during the first week of the deliverable as other commitments were priority at this time. After this a meeting was had to discuss the second deliverable and again create and allocate tasks. This was heading into the mid semester breaks which meant that meeting up and communicating as a whole group was going to be more difficult as some people would not be in Christchurch during the break. To combat this those still in Christchurch had meetings as per usual while a summary was created and send to the other members of the group and they were advised to check trello and assign themselves tasks. No main functionality of the system was implemented over the break only base classes. Once break was over it was time to tackle some of the larger classes and main functionality of the system. After a week it was found that our current system for completing tasks was not going to work any more as the tasks were larger and a lot more communication was required. Therefore we adapted by adding more meetings where we would all come together to work on the project. These peer programming sessions proved to be successful as we again were able to complete all the tasks required without a large push close to the deadline. One thing that we learnt from this deliverable is that even though the project code was well underway the design document should not be left behind. During the second deliverable the design document was just an after though and this wasn't good enough as it should be an active document that accurately reflects our system and we would have to make up for this in during the third deliverable.

Deliverable 3:
No work was completed during the first week as other commitments were priority at this time. After thus a meeting was held to discuss what tasks to prioritize. This was important as we needed to make sure that it was clear what tasks were going to maximise our number of marks while being achievable in the time remaining. The tasks that were decided upon were made into trello cards and were to be completed within the week. During the third week the final trello cards completed and final tweaks and bug fixes on the project completed. Overhaul of the design document begun, moving from google docs to latex as well as updating each section from the design document. This was thought to be necessary as our design document was severely lacking in quality and google docs was not stable enough to allow all members of our team to work on it at the same time. A lot of effort was put in during the final week which was to be expected. During week two, although we were able to complete a lot of work towards the project it was not enough. We did not follow the deadline we set for ourselves strictly enough and this caused us to have to put in a lot more effort in the final week then we had initially planned.

Overall:
Overall the team worked well together and we were able to complete the project to a good standard. Looking back there is a lot to be improved on in terms of planning. One things that was discussed in the lab times is getting the most out of the tools we are using. It was not until the third deliverable that we were adding anything more than just a name to a trello card. Adding deadlines and check boxes in earlier deliverables would have been beneficial as it would help to get tasks done correctly the first time and to help solidify the deadlines that we set for ourselves. Also taking an entire week of was not a good idea especially considering that we did it multiple times. This time should have been used to at least plan ahead and complete smaller tasks. Finally the peer programming sessions that we implemented during the second deliverable should have been something that we did from the very start as they increased the quality of our work as well as keeping everyone on the same page.

15 Document Changelog

Appendices

A Class Diagram