F29SO - Software Engineering
Stage 1 - The Bid

Bid for Restaurant Ordering System by Buzzword

November 25th 2017

Buzzword (Group 5)

Kerr Brydon	-		Wai Chong	-	
Craig Duffy	-	H00235298	Tommy Lamb	-	H00217505
Alastair Nibloe	-		Mamta Sofat	-	
Michael Steventon	-		Michael Lones	-	Manager

Contents

Chapter 1

Requirements Specification

Contents

1.1 Introduction

This document outlines the requirements, scope, and boundaries placed upon the project based on an interpretation of the document "Year 3 Group Project Specification - Restaurant Ordering System"

1.2 Purpose

The purpose of this system is to support restaurant staff in taking and fulfilling food orders from customers. The system is intended to make the process of relaying orders to the kitchen and fulfilling them more efficient, whilst increasing the information available to customers regarding their order. The aim is to allow servers to take quickly take complex orders from a customer or group of customers and have this automatically relayed to kitchen staff. The kitchen staff will then be able to provide certain feedback on the order, such as estimated time to delivery, which will be viewable by customers through this system.

1.3 Scope

The Restaurant Ordering Support System (henceforth "The system", or "ROSS"), will be designed to receive orders from waiting staff and relay this information to staff working in the kitchen. ROSS will also allow waiting staff to amend orders, kitchen staff to provide feedback on orders (for example, the time until delivery), and customers to view some or all of this feedback alongside their order details. This is to achieve the aim of increasing operational efficiency at the restaurant and customer experience.

The system will also be designed to handle a good variety of data to support the client's need to deploy the system across multiple differing restaurant environments.

The system will not be designed to provide detailed instructions on the preparation of food, only what dishes have been ordered (with any modifications). It will also not implement a stock management system, though some emulation of such a system may be implemented.
No aspect of the system will process payments from customers, though price information about items on the menu may be represented in the system. The system will also not be explicitly designed to integrate with any other technologies the restaurant may be using, for example, Point of Sale or Stock Control systems.

1.4 Overview

1.4.1 Context

The system will interact with 3 existing groups of people:

1. Waiting Staff
   These are the people who converse with customers to elicit instructions for the kitchen staff to act upon. These instructions are usually in the form of food and/or drinks orders, or amendments to previously made orders. In the existing system, they are also responsible for relaying information from the kitchen staff to a customer. Waiting staff may or may not deliver food from the kitchen to customers.

2. Kitchen Staff
   This group consists of any person responsible for the preparation of food or drink for customer consumption.

3. Customers
   Anyone who enters the restaurant's premises with the intent to purchase food or drink is considered a customer. The System is only concerned with formalising a customer's request, not with encouraging or eliciting purchases.

These groups of people (collectively "users") will interface with ROSS through one of five distinct software components:

1. A Kitchen component to display information on current, unfulfilled orders to kitchen staff. The component will also be responsible for processing input to the system from the kitchen staff.

2. A Waiting component to allow waiting staff to enter, view, and amend customers' orders into the system.

3. A Customer component allowing customers to view the current status of their order. Any other direct interaction between the system and any customer (I.E. not through another user) will be the responsibility of this component.

4. A Database component responsible for storing any persistent information required by the system. For example all current unfulfilled orders, the restaurant's menu, and the availability of items on it. These examples will not necessarily be required by or implemented in the system.

5. A Server component which will manage the flow of information between the four previously mentioned components.

The general process of interaction expected is for a customer to converse with a member of Waiting staff to place or amend an order, or to request an update on their existing order. This member of Waiting staff will then interact with the system to formalise the customer's request through the Waiting component. The staff member will then receive some information from that same component regarding the request; for example, "Accepted", "Denied", details of an order, etc. These details can then be acted upon by the staff member or customer.
When a request has been formalised in this manner, the request is passed from the Waiting component to the Server component, which will act upon the request and interact with other components as required. Interaction with the Database component will usually be related to either recording the request for possible future use within or outwith the system, or to confirm such a request is valid or feasible.

In the case of new orders being placed or existing orders amended, the relevant details from the request will be passed from the Server component to the Kitchen component.
This component, in relaying order details to the kitchen staff, allows them to prepare the food or drink as requested by the customer. It will also allow them to provide information related to the request to ROSS, which will then act accordingly.

The customer's direct interaction with the system will include viewing the status of their order. That is the formalisation of their order recorded by ROSS, and some or all of the related information provided by the kitchen staff. Further interaction may or may not form part of the system as finally implemented.

Interaction with the aforementioned software components will take place through hardware owned and operated by the customer (for the Customer component) or provided by the restaurant in all other cases.
Interfaces between the components will utilise standard web technologies and practices as described in Section 1.8: System Interfaces.

1.4.2 Functions

System functions are discussed generally in Subsection 1.4.1: Context and specifically in Section 1.5: Functional Requirements

1.4.3 User Characteristics

Expanding on the user groups defined in Subsection 1.4.1: Context, their use of the system can be further defined as follows:

1. **Waiting Staff**
   This group will interact with the system through a mobile wireless device, most likely in the form of a tablet computer. Each member of the group will have a device. Given the number of staff members varies by restaurant, the size of this group is unknown; however it is not expected for there to be any more than 20 such people at any given time. This group consists only of Waiting staff who are working at the time in question.

   It is expected that with their regular usage this group will become familiar with the system's interface design (the Waiting component particularly), its functionality and limitations. As such the Waiting component can rely more heavily on implicit knowledge, making it more efficient to use through a simpler interface.
   There is a distinct possibility that certain aspects of the interface could be operated without looking at it given enough experience, and the design will aim to maximise this possibility.
   Using the system both on a regular basis multiple times a day and in relation to employment, this group likely place ease of use and functionality significantly higher than visual aesthetics. They likely do not want to enjoy the interface per se, rather they want it to be second nature and make no investment - physical or mental - in its operation.

2. **Kitchen Staff**
   The kitchen staff also will make regular usage of the system multiple times a day in their employment. However their requirements are subtly different. In the potentially hectic and high-pressure environment of a kitchen, the interface (Kitchen component) must require a minimal amount of time and thought to operate. The aesthetics must also be carefully considered in order to convey as much information as possible without requiring appreciable concentration - it should work well with only glances from the staff.
   While implicit knowledge can be assumed through experience, 'blind' operation will likely not feature due to the changing nature of the information being displayed.

   This group will utilise a number of touch-screen devices of a size roughly according to an average desktop PC monitor. These will be installed in the kitchen itself in fixed locations, and may be shared amongst members. As such the size of kitchen is the determining factor rather than the number of staff. An amount from 1 to 5 would be considered normal, with anything up to 10 devices being reasonably expected.

3. **Customers**
   Unlike other groups, the customers will not make regular usage of the system, and will not likely use it constantly throughout the day. As such they likely place a relatively high value on the aesthetics and enjoyment from using the system, and are willing to sacrifice some aspects of usability or functionality. That is not to say those aspects can be neglected.
   As a consequence, customers will require much more prompting and signposting from the system to help formalise their intent. Given their intermittent usage, customers will willingly pay more attention to the interface and afford more deliberation in their actions. This reduces the requirement for a slick and intuitive interface relative to the Kitchen and Waiting staff members.

1.5 Functional Requirements

F-UR-1 **Allow waiting staff member to place customers' orders**
Priority: Variable
Highest Priority: Must Have
Source: "Group Project Specification Document"

F-UR-1.1 **Allow staff member to enter unique identification**
Priority: Must Have
The system will accept from the staff member 1 or more pieces of information that will allow the order to be uniquely identified by both staff member and customer. These details will be some combination of:

- Any part of the customer's name. Forename, surname, nickname, pen name and similar are all acceptable.
- Group Number. A number determined by the restaurant to group relevant orders together. For example, using a table number allows multiple separate orders by different people to be logically grouped.
- Unique Order Number. A number specific and unique to the order to which it is assigned. This is the only detail allowed to be used by itself.

The unique ID should be easily memorable by the customer. It is preferred if the customer has some reference to the Group Number or Unique Order Number eg. Number marked on the table.

F-UR-1.2 **Present staff member with restaurant menu**
Priority: Must Have
Show to the staff member the menu of the restaurant in a hierarchical manner.

The position of elements displayed to the person should be consistent for any number of orders where the menu hierarchy has not been modified, and for any location in that hierarchy.

F-UR-1.3 **Allow staff member to navigate the menu and select an item**

Priority: Must Have

The staff member must be able to navigate the menu hierarchy in an arbitrary order, able to revisit and/or return to a previous location in the hierarchy.

They must also be able to select any item from the menu, and to return to the menu from that selection. The location in the menu they return to should be the same location from which they made the selection.

F-UR-1.4 **Allow staff member to record arbitrary details regarding selected item**

Priority: Must Have

After selecting an item, the person must be able to record any arbitrary requests made by the customer regarding the item. This may include, but is not limited to:

- Requests to omit items (eg remove cheese from a burger)
- Requests to add or substitute items (eg swap chips for curly fries)
- Acknowledgement of dietary or allergy requirements (eg peanut allergy)
- Selection from options listed on menu (eg cake with choice of custard or ice cream)

It must also be possible to record the amount of said item to be ordered. Where some arbitrary detail/request has been recorded, for example to omit cheese, that detail should be applied to **all** of the items specified by the amount recorded. So for example if the request "No cheese" and the amount "5" are recorded for an item "burger", the order should consist of 5 burgers of which **none** have cheese.

F-UR-1.5 **Forming and placing order for fulfilment**

Priority: Must Have

The staff member must be able to select and record details for a number of different items and have the system remember all previous selections and their details. Additionally they must be able to view this list of selections (defined as an order), and remove items from it.

They must be able to place this order for fulfilment, such that information recorded by the system is displayed pursuant to item F-UR-3.

The exact number of items an order may contain is specified in item NF-UR-15.

F-UR-1.6 **Automatically grouping items in an order**
Priority: Could Have
The system could automatically group certain menu items together in the order and show these groups as part of item F-UR-3, with the aim of all the items in a group being complete and delivered to the customer at approximately the same time. For example, it could group starters, mains, and desserts into separate groups such that each group is individually prepared and served sequentially.

F-UR-1.7 **Specify item groupings in an order**
Priority: Should Have
Similar to item F-UR-1.6 and in addition to item F-UR-1.4 above, the system should separately and explicitly record groupings of items within an order. Any specified grouping must override any automatic grouping that may otherwise be made by the system under item F-UR-1.6.

F-UR-2 **Allow waiting staff to amend existing orders**
Priority: Must Have
Source: "Group Project Specification Document"

F-UR-2.1 **Retrieve order details**
The system must request the unique identification details entered as part of item F-UR-1.1 from the user.
If these details are invalid, the system must notify the user and prompt again for input.
There is no expectation for the system to check if the details refer to the intended order.

Once a valid (but not necessarily 'correct') set of ID details has been received, the system must display the details of the order (to which the entered ID details refer) to the user. The system is not required to display all stored information regarding the order, only that which is considered pertinent.

F-UR-2.2 **Amend or Remove order details**
With an order being displayed to the user, they must be able to select any item in the order and change any associated details recorded as part of item F-UR-1.4.
The user must also have the ability to remove any number of items from an order.
The system will also provide functionality to remove the entire order from the system.

The waiting staff member may remove an order marked complete under item F-UR-4.2, but may not modify its details.

F-UR-2.3 **Confirm amended order**
Once the user has finished modifying the order (including removing it entirely) this change in information must be propagated through the system, and specifically notify the Kitchen staff as per item F-UR-3.2.

F-UR-3 **Display state of orders to Kitchen Staff**
Priority: Must Have
Source: "Group Project Specification Document"

F-UR-3.1 **Display all unfulfilled orders**
The system must produce a complete list of all orders placed under item F-UR-1 for display to Kitchen Staff with regards to item NF-UR-6 and any other relevant requirements. All relevant information for an order must be displayed. This should be as a collection of menu items, and must include any requests made in accordance with item F-UR-1.4.

F-UR-3.2 **Notify of amendments to orders**
The system must explicitly draw the attention of any Kitchen staff to changes in the order state which result from actions taken under item F-UR-2. This includes where entire orders have been removed from the system.

F-UR-4 **Add or Update order information from Kitchen**
Priority: Variable
Highest Priority: Must Have
Source: "Group Project Specification Document"

F-UR-4.1 **Provide an estimate of time to complete an order**
Priority: Must Have
The system must allow Kitchen staff to provide an estimate of the time required to complete preparation of an entire order. This must be recorded by ROSS and made available for users to view as part of item F-UR-2.1.
The system must also allow for this estimate to be changed at any time.

F-UR-4.2 **Mark an order as complete**
Priority: Must Have
Kitchen staff must be able to mark an order as complete within the system, after which it cannot be amended - though it must be able to be removed from the system.

F-UR-4.3 **Provide an estimate of time to complete an individual item**

Priority: Should Have

The system should allow Kitchen staff to specify an estimated preparation time for each individual item in an order, and have these individual timings displayed as part of item F-UR-2.1. From these estimates the system should derive an overall estimate for the entire order.

The system must also allow for these estimates to be changed at any time if made.

F-UR-5 **Kitchen staff can acknowledge information changes**

Priority: Should Have

Source: Tommy Lamb

F-UR-5.1 **Kitchen staff can acknowledge order amendments**

With respect to item F-UR-3.2, kitchen staff may (but are not required to) acknowledge amendments to orders. Where an amendment is acknowledged the interface must display the modified order in the same manner as any other order which is in the same state. For example, if an order currently being prepared is amended the acknowledgement should cause it to be represented in the same manner as other in-progress orders. Likewise for orders awaiting preparation, and any other states that may occur. This requirement does **not** include where orders have been cancelled/removed form the system. See item F-UR-5.2.

F-UR-5.2 **Kitchen staff must acknowledge order cancellations**

With respect to item F-UR-3.2, kitchen staff **must** acknowledge any order which has been cancelled. Until this acknowledgement the interface must continue to draw attention to the cancelled order. Only after this acknowledgement is received may the system consider the order cancellation completed and act accordingly.

F-UR-6 **Allow kitchen staff to remove orders**

Priority: Could Have

Source: Tommy Lamb

F-UR-6.1 **Remove order from system**

The system could allow kitchen staff members to cancel an order, removing it from the list of unfulfilled orders in the system.

The system should record a reason for the cancellation as provided by the kitchen staff member cancelling the order.

F-UR-6.2 **Notify Waiting staff member of cancellation**
When an order is cancelled by the kitchen, the system should notify at least one member (and preferably only one) of the waiting staff who can relay this information to the customer in a personable manner.

F-UR-7 **Allow customer to view order status**
Priority: Must Have
Source: "Group Project Specification Document"

F-UR-7.1 **Retrieve order details for customer**
This should function in the same manner as item F-UR-2.1, without specifying that the user interface be the same.
This requirement need not show the same details as item F-UR-2.1, only those details considered relevant to the customer.
The minimal set of details that must be shown are:

- Items ordered, alongside any specific requests made under item F-UR-1.4.
- Estimated preparation times as recorded under item F-UR-4.
- If the order has been marked complete by the kitchen as under item F-UR-4.

F-UR-8 **Allow customer to make, amend, and remove orders**
Priority: Could Have
Source: "Group Project Specification Document"

F-UR-8.1 **Customer placing an order**
This would execute in the same manner as item F-UR-1. However the specific implementation of those requirements (including User Interface elements) should be considerably different, respecting the differing user needs outlined in Subsection 1.4.3: User Characteristics

F-UR-8.2 **Customer amending or removing an order**
This would execute in the same manner as item F-UR-2. However the specific implementation of those requirements (including User Interface elements) should be considerably different, respecting the differing user needs outlined in Subsection 1.4.3: User Characteristics

F-UR-9 **Generate report based on previous, fulfilled orders**
Priority: Could Have
Source: "Group Project Specification Document"
The system could generate reports based on the orders made through ROSS for use by restaurant management in business decisions. The exact form or usage of these reports is unspecified.

F-UR-10 **Intelligent recommendation system for order items**
Priority: Will Not
Source: "Group Project Specification Document"
ROSS will not implement functionality to display to a user recommendations for items to add to the order being made or amended.

1.6 Usability Requirements

NF-UR-1 **Quick for waiting staff to enter orders**
Priority: Variable
Highest Priority: Must Have
Source: Tommy Lamb

NF-UR-1.1 **Single item orders must take no more than 1 minute**
Priority: Must Have
From start to finish, it must be possible to take less than 1 minute for a member of waiting staff to record a single item as part of an order. This time does not include recording any specific requests under item F-UR-1.4, nor any time taken for the order to be propagated through the system (eg passing the order to the server component). This does include entering unique information (item F-UR-1.1), but not generating the information (eg asking the customer their name).

This requirement does not stipulate that all such orders must take less than 1 minute, only that it is feasibly achievable with the User Interface being used.
The requirement also makes no specification for the experience of the staff member. It is acceptable for the staff member in question to be experienced both in using the interface and menu structure in meeting this requirement.
Reasonable accommodation should be made for the complexity of ID information being entered and the size/complexity of the menu.

NF-UR-1.2 **Single item orders should take no more than 30 seconds**
Priority: Should Have
The same conditions as item NF-UR-1.1, except that the process should take no longer than 30 seconds. In meeting this requirement consideration must be made for the size and complexity of menu. It may not be (and is not expected) that all items in the menu could be used in meeting this requirement. Likewise a certain level of experience may be required in meeting this requirement.

NF-UR-2 **Menu information must be up to date**
Priority: Must Have
Source: Tommy Lamb
With respect to item F-UR-1.2, the menu information displayed by the system should represent the menu at that time. The menu information must not be outdated (but may become outdated in the time it is being displayed to the user).

NF-UR-3 **Orders placed must be validated**
Priority: Must Have
Source: Tommy Lamb
With respect to item F-UR-1.5 and item F-UR-2.3, any orders or amendments recorded by the system must be confirmed to be valid. Where it is invalid, the system must notify the user.

NF-UR-4 **Kitchen estimates should be quick to enter**
Priority: Must Have
Source: Tommy Lamb
With respect to item F-UR-4, it must take no longer than 30 seconds to enter a time estimate or mark an order as complete. The user must be able to complete this task using their non-dominant hand. The criteria must also be met if the user is using their dominant hand. In either case, the task must be completable using only a single hand, not both.
This does not include the time taken to come up with the estimate or to realise the order is ready, only the time to register the information in the system.
This does not preclude a user using two hands to operate the system, just that it must be fully operable using a single hand.

NF-UR-5 **Kitchen estimates are not required by system**
Priority: Should Have
Source: Tommy Lamb
While the system must support preparation time estimates as per item F-UR-4, the system should also cope with orders which have no estimate provided. This should hold true for the entire life-cycle of an order, from when it is first placed through to completion and including any modifications. When displaying order details as per item F-UR-2.1, a meaningful message must be displayed (if pertinent) regarding the lack of estimate. This applies particularly to item F-UR-7.

NF-UR-6 **Kitchen displays must be easily read**
Priority: Must Have
Source: Tommy Lamb

NF-UR-6.1 **Displays understandable from a distance**
With respect to item F-UR-3, all information displayed (in whatever form that may take) must be understandable by a user from a distance of at least 150 centimetres. The user is assumed to have good vision (does not require corrective lenses) and an unobstructed view of the interface. It is also assumed they are familiar with the interface and can therefore recognise certain graphical elements eg. symbols, colour coding.

NF-UR-6.2 **Displays must support wide viewing angles**
With respect to item F-UR-3, any hardware used to display visually the information must support wide viewing angles in both the horizontal and vertical planes. Standard display technologies such as LCD are expected to suffice for this requirement.

NF-UR-6.3 **Displays easy to understand and unambiguous**
With respect to item F-UR-3, a user must be able to understand any **single** piece of information being displayed within 5 seconds. They must have understood the information well enough to accurately relay it to another person at least 5 seconds after viewing the display.
The user is assumed to have a view of the display meeting the assumptions and constraints of item NF-UR-6.1. This includes interface experience.

NF-UR-7 **Kitchen interface must not rely on audio cues**
Priority: Must Have
Source: Tommy Lamb
With respect to item F-UR-3, no piece of information should be solely communicated to users through sound of any sort. This does not preclude the use of audible cues within the interface, but any such cue must be used in combination with a visual element of some description.

NF-UR-8 **Kitchen interface should not make unnecessary notifications**
Priority: Should Have
Source: Tommy Lamb
With respect to item F-UR-3.2, the system is only required to draw attention to amended or cancelled orders for which it has reason to believe the kitchen has actively begun preparation of any item within it. Any order which the system does not believe is being prepared currently does not have to have a notification of any sort when amended or cancelled.

NF-UR-9 **Order status must remain up to date**
Priority: Must Have
Source: Tommy Lamb

With respect to item F-UR-2.1, the information displayed to the user must reflect the most recent information held in the system. Any changes in the information as held by the system must be reflected in the interface within 2 minutes of the change occurring.
This time does not include any delays caused by technical issues, such as those which may affect communication between the various system components. Rather this reflects a normal operating environment.

NF-UR-10 **Customer interface cannot require training**
Priority: Must Have
Source: Tommy Lamb
For any functionality involving a customer (the group as defined in Subsection 1.4.1: Context), the interface must be fully usable without training or instruction external to the interface itself. In other words, the customer must be able to execute any function of the system given the interface and no other information.
It is assumed however that the customer in question has awareness of the context of usage, and a basic level of technological competence. Explicitly, an inability to use a standard web browser (or similar event) would not count has a failure against this requirement, even though it is part of the overall interface.

NF-UR-11 **Customers do not require an account to view order status**
Priority: Should Have
Source: Craig Duffy
A customer should not be required to register details with the restaurant in order to view the details of their order. This does not preclude asking for the unique order ID under item F-UR-1.1, however any information entered must not be indefinitely stored.
This requirement explicitly does not apply to item F-UR-8.

1.7 Performance Requirements

NF-UR-12 **ROSS must handle at least 500 orders an hour**
Priority: Must Have
Source: Craig Duffy
ROSS must be able to handle anything up to and including 500 orders being placed for each and every continuous hour of operation without error.
This requirement does not preclude the system failing due to an excess number of orders in a period of time shorter than one hour.

NF-UR-13 **ROSS must be capable of handling 10 orders in one minute**
Priority: Must Have

Source: Tommy Lamb
The system must have the capability to handle at least 10 orders being placed within one minute without error. This does not require that 10 orders be handled every minute for the duration of operation, rather that within one minute of **independent** operation 10 orders could be handled. It is acceptable that conditions preceding the minute in question prevent 10 orders being processed without error.
It is acceptable for the system to fail should more than 10 orders be placed within one minute.

NF-UR-14 **Orders must be propagated quickly**
Priority: Must Have
Source: Tommy Lamb
The system must be able to fully propagate a new order or amendment to an existing order through all components within 1 minute of confirmation. Explicitly, any capable component must be able to display to a user the details of the order within 1 minute of confirmation.
Confirmation here refers to the completion of item F-UR-2.3 or item F-UR-1.5.

NF-UR-15 **System must support orders up to 100 items**
Priority: Must Have
Source: Tommy Lamb
The system must support the placement of orders consisting of up to (but not including) 100 items. The order may contain any combination of unique or duplicate items, which may or may not have specific requests made under item F-UR-1.4. The value of 100 does **not** include any amounts specified under item F-UR-1.4. As such the order may request more than 100 portions of food or drink from the kitchen when the order as entered into the system does not list more than 100 items.
This is a technical requirement placed on the system and does not preclude a lower operational limit being imposed by the restaurant.

1.8 System Interfaces

See also requirements specified under Section 1.6: Usability Requirements.

NF-UR-16 **Intercomponent communication must use web standards**
Priority: Must Have
Source: Tommy Lamb
In communicating between components, with the exception of between the server and database components, HTTP and underlying networking standards must be used. These may include, but are not limited to:

- Ethernet
- 802.11 (commonly called Wi-Fi)
- IPv4 / IPv6
- TCP / UDP

In addition to this, the usage of HTTP must be through standard protocols and practices.
The precise form of data being transferred should be some combination of the following, complying with W3C standards:

- HTML
- CSS
- JavaScript
- JSON

Communication between server and database components should be carried out using the best available standard compatible with the specific database implementation. For example (which is non-binding), SQL.
The exact versions of any technology listed here is explicitly unspecified at this time.
The mandated use of these standards does not preclude the use of non-standard software, where that software itself conforms to or makes use of these standards. For example, JavaScript libraries or What-You-See-Is-What-You-Get editors.

NF-UR-17 **Human-Computer interfaces must be touchscreen compatible**
Priority: Must Have
Source: Tommy Lamb
The hardware interface between Kitchen, Waiting, and Customer components and their users must be fully compatible with touchscreen technology. Any and all functionality or features of these components should be accessible through a touch screen interface.
This does not preclude the potential use of other Human-Computer interface technologies, such as a mouse and keyboard.

NF-UR-18 **Hardware-Software interfaces should be through standard web browsers and operating systems**
Priority: Should Have
Source: "Group Project Specification Document"
With the exception of Server and Database components, all components should operate on their hardware and be fully functional through any of the following browsers:

- Google Chrome
- Mozilla Firefox
- Apple Safari
- Microsoft Edge

Additionally those components which run on browsers should be fully functional on any of the following operating systems:

- Microsoft Windows
- Apple MacOS
- Apple iOS
- Android
- Certain Linux Distributions, including
  Ubuntu
  CentOS

No specification is made for what hardware the system should operate on, as this is considered in the domain of the operating system being used.

1.9 System operations

1.9.1 Human System Integration Requirements

No specific requirements are made under this section. Other requirements relevant to Human-System integration may be found in Section 1.8: System Interfaces and Section 1.6: Usability Requirements.

1.9.2 Maintainability

NF-UR-19 **Standard components should be easily maintained**
Priority: Should Have
Source: Tommy Lamb
Those components of the system which have not been specifically designed or implemented as part of this project should be reasonably maintainable by a single person who has appropriate domain-specific knowledge and experience. Certain issues may arise with which the person requires assistance from an outside party. This requirement does not preclude that possibility, nor is this requirement considered unmet under such circumstances. It is only required that such a person be able to maintain the specified components to a reasonable degree. The components to which this requirement applies include, but are not necessarily limited to:

- Operating System Software
- Device drivers
- All hardware owned and operated by the restaurant, including
  - Waiting staff devices
  - Kitchen devices
  - The server hardware
  - Networking hardware
- Web server software

This requirement explicitly does not apply to bespoke software written as part of this project.

NF-UR-20 **Bespoke software components should be highly independent**
Priority: Should Have
Source: Tommy Lamb
All bespoke software components written as part of this project should be designed and implemented in a way which minimises coupling between components. As a consequence, most maintenance of those software components should be able to be carried out in isolation without interfering with other components.

1.9.3 Reliability

No specific requirements are made under this section. Related requirements may be found under Section 1.7: Performance Requirements.

1.10 Physical Characteristics

1.10.1 Physical Requirements

NF-UR-21 **Mobile hardware components should be light**
Priority: Must Have
Source: Tommy Lamb
Any mobile hardware operated by the restaurant as part of this system must not weigh more than 1 kilogram. Mobile here refers to any hardware which is intended to carried regularly by any user.

1.10.2 Adaptability Requirements

NF-UR-22 **Easy to add new component instances**
Priority: Must Have
Source: Tommy Lamb
The system must be designed to make it as easy as possible to add new component instances. For example, increasing the number of Waiting staff components, or displays in the kitchen.

1.11 Environmental Conditions

NF-UR-23 **Server must be adequately protected**
Priority: Must Have
Source: Tommy Lamb
The environment in which the server hardware is to be installed must be suitable for sensitive electronic equipment. This necessitates as a minimum:

- Free from liquids and excess humidity
- Free from excess dust and other fine particles
- Free from combustible materials (including gases) that may be ignited by moderate heat or an electrical charge
- Free from or protected against electromagnetic interference
- Protection against power supply fluctuations (including power surges)
- Free from corrosive materials
- Free from children and animals
- Free from external vibrations and risk of physical shock
- Adequate ventilation

NF-UR-24 **Kitchen hardware should operate under hazardous conditions**
Priority: Should Have
Source: Tommy Lamb
Hardware used in the kitchen must be able to operate fully for prolonged periods in any combination of the following conditions:

- High temperatures
- Moderate levels of airborne particles
- High humidity
- Risk of liquid splashes

NF-UR-25 **Waiting and customer hardware require low electromagnetic interference**
Priority: Must have
Source: Tommy Lamb
Due to the wireless nature of communication, the environment for the customers and waiting staff must have low levels of electromagnetic interference to ensure performance. In particular, signal noise in the 2.4GHz, 5GHz and surrounding frequencies must kept to a minimum. This requires only that the communication signal being used be significantly stronger than any other electromagnetic radiation in the stated frequencies. This may be achieved through any balance of increasing signal strength or reducing background radiation.

1.12 System Security

NF-UR-26 **System must not be internet accessible**
Priority: Must Have
Source: Tommy Lamb
No aspect of the system can be accessible from outside of the restaurant's Local Area Network.
This applies only to the system as installed within a specific restaurant. It explicitly does not apply during any phase of this project prior to installation at a restaurant.

NF-UR-27 **Software components should be hardware locked**
Priority: Must Have
Source: Tommy Lamb
Any and all software components of the system designed for use by any staff member should be limited to operating on specific hardware devices. Specifically, the Kitchen and Waiting components should only be accessible through the hardware devices installed in the kitchen and given to waiting staff.
This requirement makes no specification regarding access, physical or otherwise, to these hardware devices.
This also applies only to the system as installed within a specific restaurant. It explicitly does not apply during any phase of this project prior to installation at a restaurant.

1.13 Information Management

No specific requirements are made under this section. Relevant requirements may be found under Section 1.5: Functional Requirements and Section 1.8: System Interfaces

1.14 Policies and Regulations

NF-UR-28 **System must be able to adapt to differing ID representations**
Priority: Must Have
Source: "Group Project Specification Document"
With respect to item F-UR-1.1, the system as it is installed in a restaurant needs only to support one form of unique ID during use. This may or may not consist of 1 or more pieces of individual data. The exact form of ID to be used by the system will be specified by the specific restaurant's organisational policies. As the system is to be designed for installation "across [a] diverse range of restaurants" (**Specification**), the system must allow the form of ID used to be specified during installation and setup.

The system does not have to support any arbitrary form of ID specified during installation, but must minimally offer:

- The combination of Table Number and Customer Name.
- A System-wide Unique Identification Number (equivalent to an order number)

This requirement does not apply to any internal representation used by the system, only to how the identity of an order is represented in interacting with any users.

1.15 System Life Cycle Sustainment

No specific requirements are made under this section at this time.

1.16 Packaging, Handling, Shipping, and Transportation

No specific requirements are made under this section at this time.

1.17 Verification

No specific plans have been made with regards to verification of the system. Suffice to say that extensive testing of software components will take place both individually and as integrated in the system. Additionally testing will take place with mock users, both familiar and unfamiliar with the system, to verify and quantify how those requirements placed on usability and human element interfaces are met.

1.18 UML Diagrams

1.18.1 Use Case Diagram

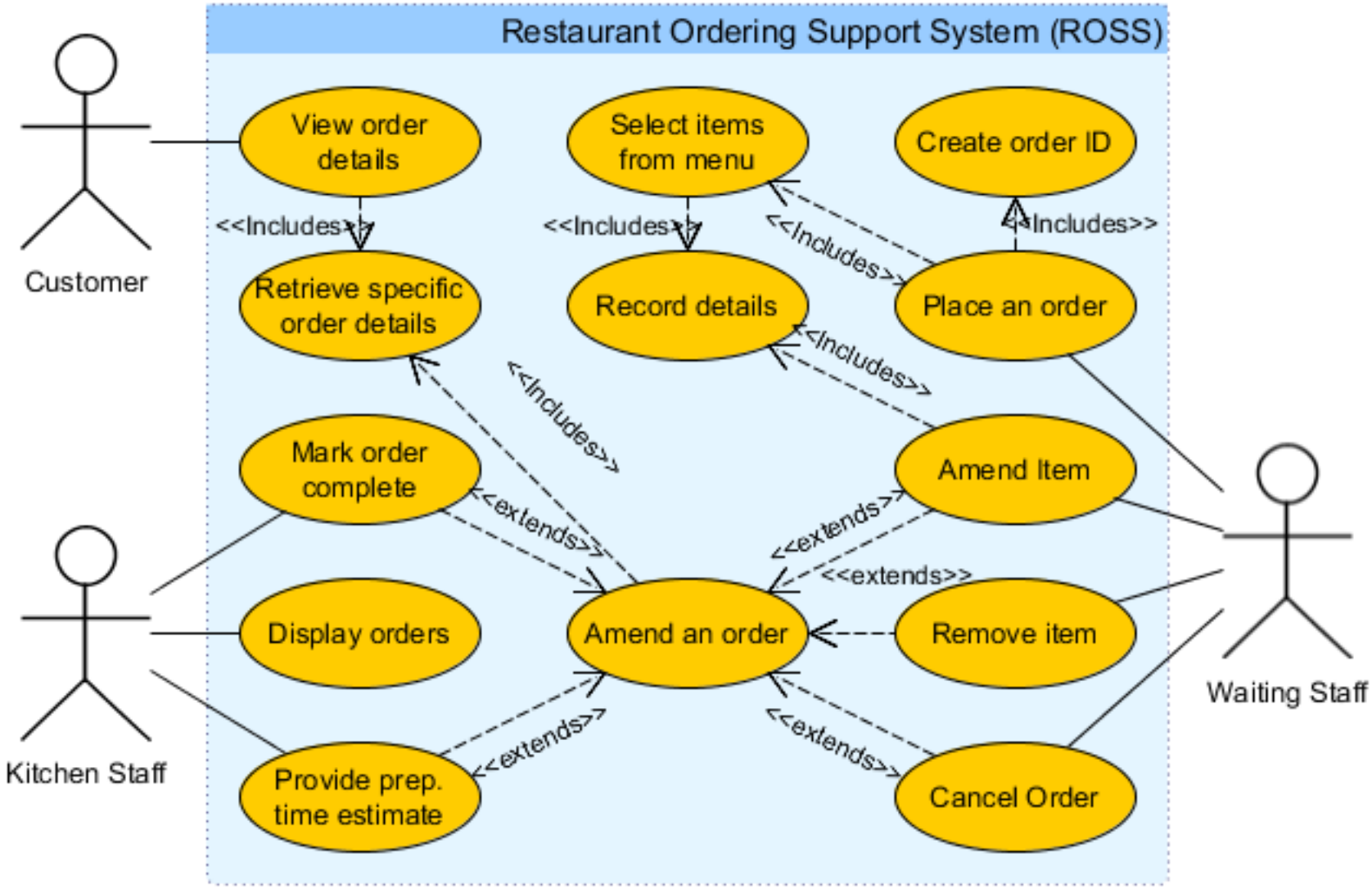


Figure 1.1: Use case diagram for the Restaurant Ordering Support System.
For clarity and simplicity a one-to-one relationship with specific requirements is not enforced, and specific use cases ignore component boundaries.

1.18.2 Activity Diagrams

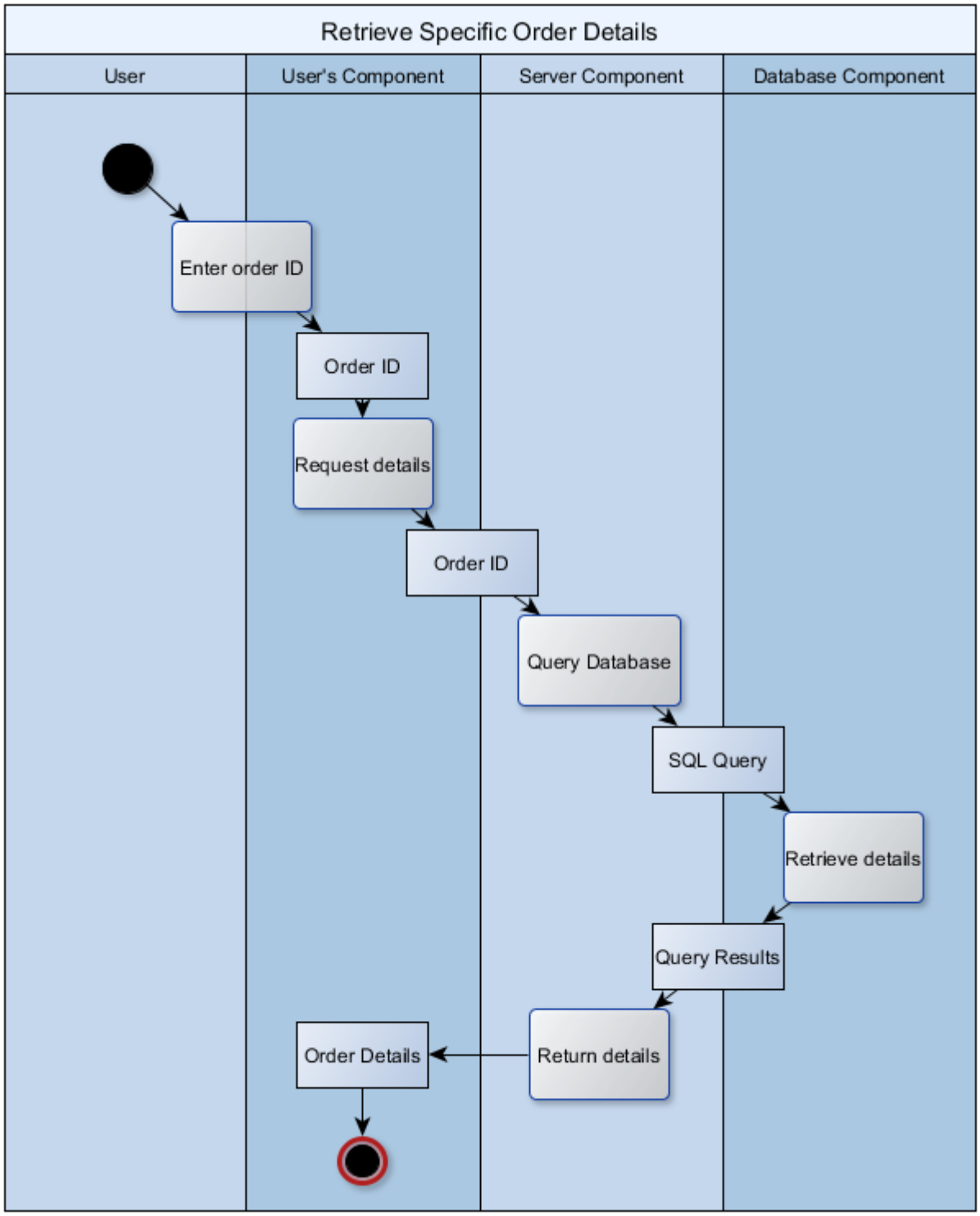


Figure 1.2: A sub-activity to retrieve details of a given order.

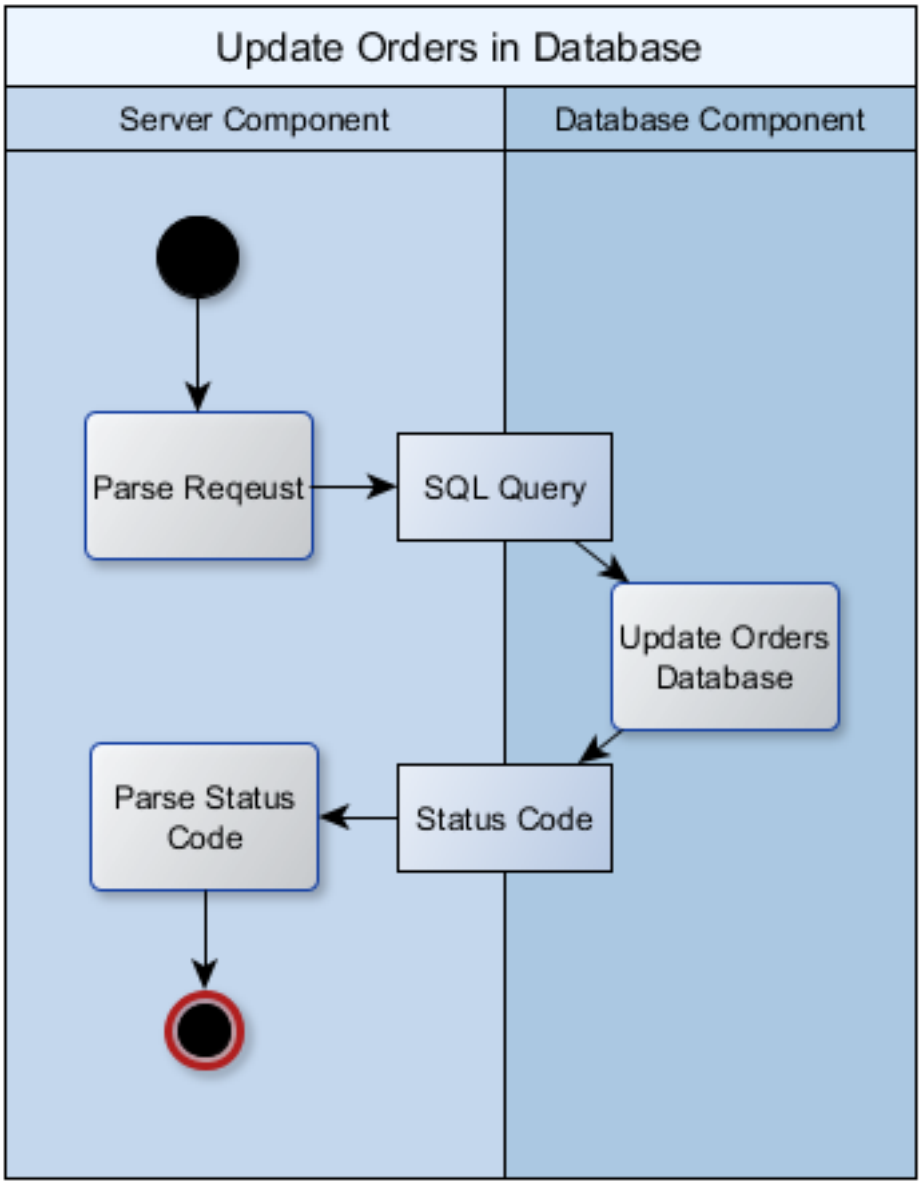


Figure 1.3: A sub-activity to perform an update to an order in the Database component.
Updates can include adding, amending, or removing an order as well as adding prep. time estimates or marking as complete.

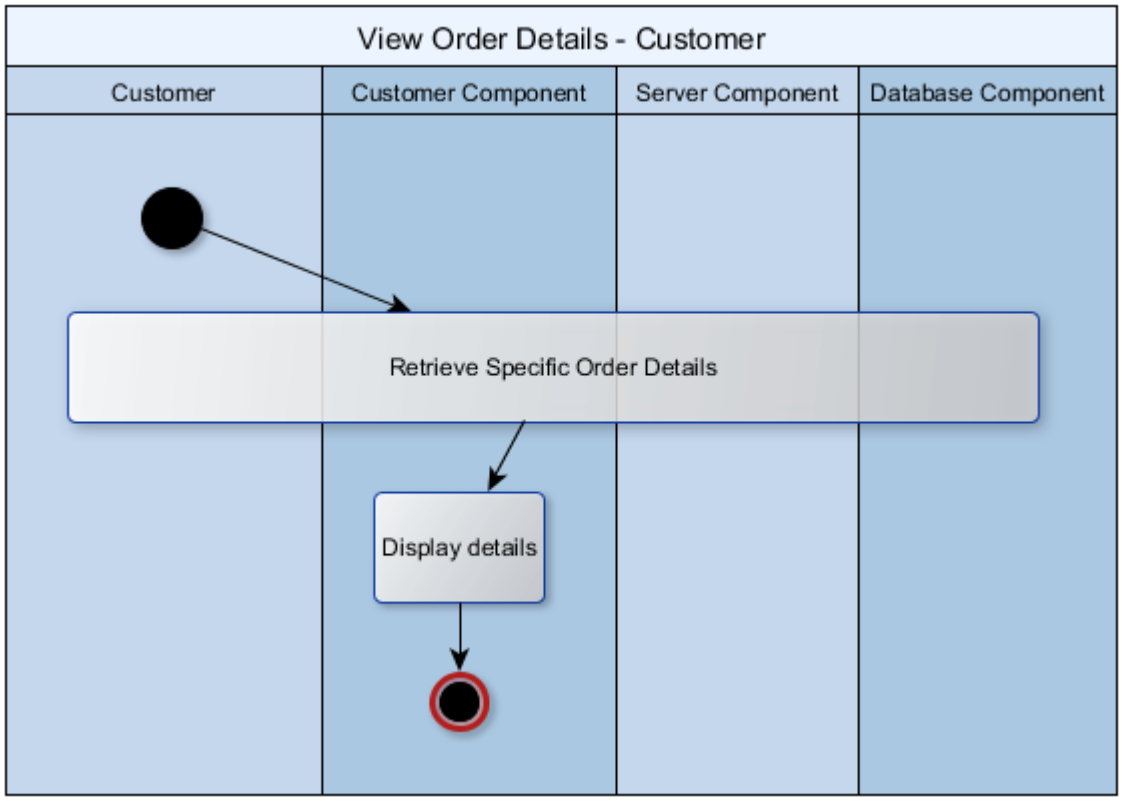


Figure 1.4: The activity diagram for a customer to view their order status.

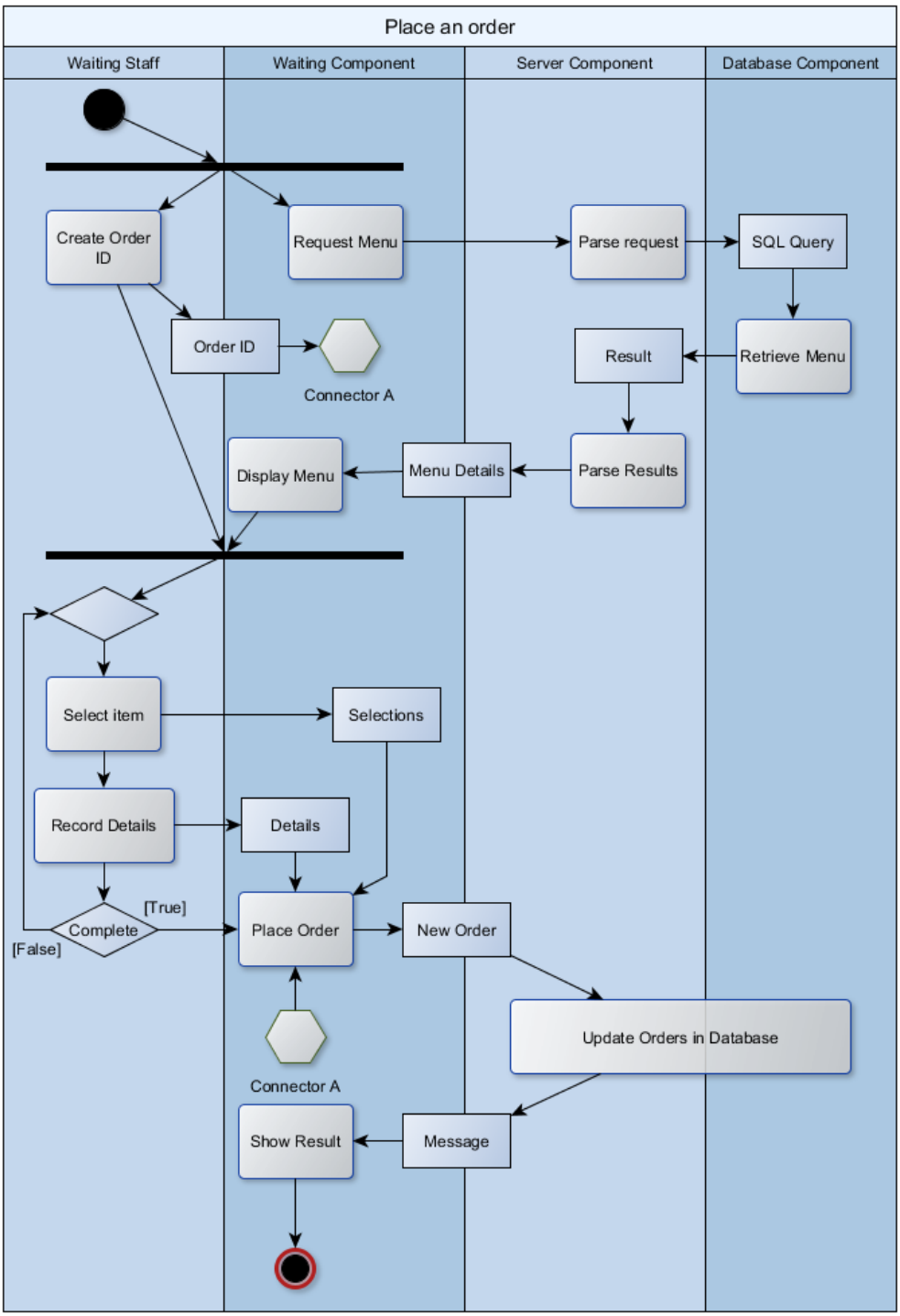


Figure 1.5: The activity diagram for entering an order into the system.

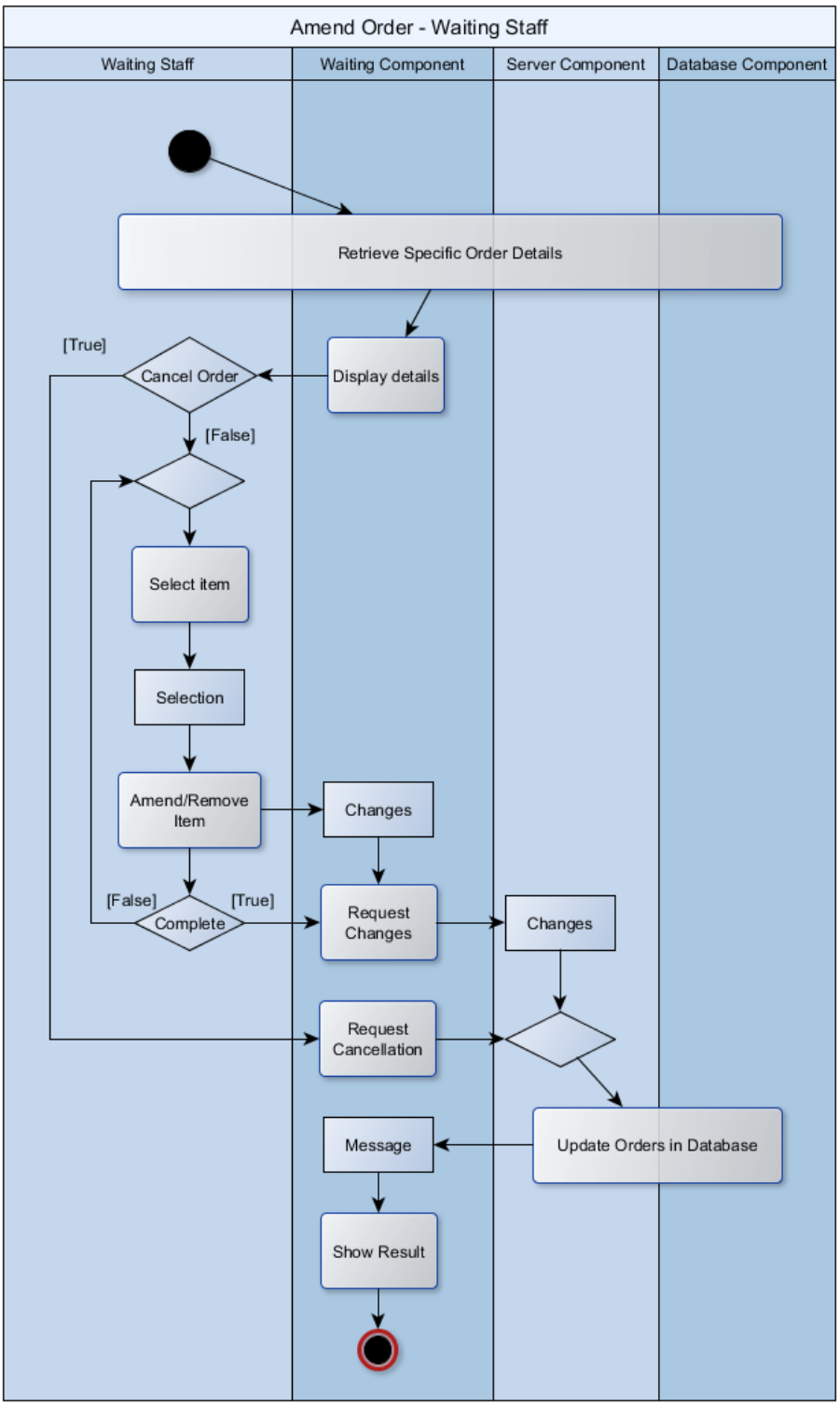


Figure 1.6: The activity diagram representing waiting staff amending an order.

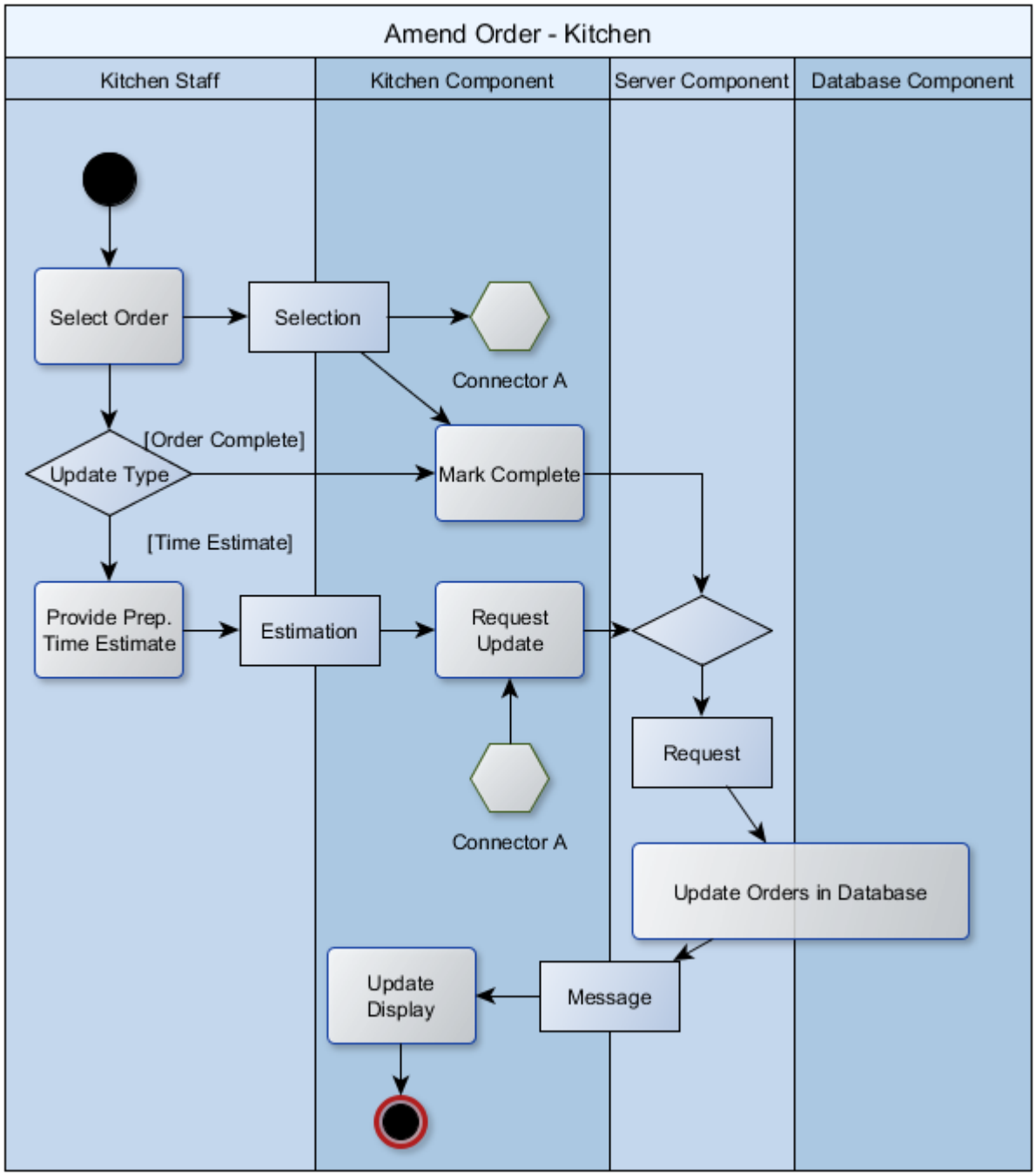


Figure 1.7: The activity diagram representing kitchen staff amending an order.

1.19 Assumptions and Dependencies

It is assumed that the restaurant is capable of implementing and maintaining a reliable wireless Local Area Network pursuant to any relevant requirements specified in this document.
It is further assumed that the restaurant has or will have an active wireless Local Area Network implemented for the installation of this system.

It is assumed that the restaurant has or will have the ability to maintain the server hardware and software - where said software has not been implemented as part of this project - by the end of this project. This may or may not be through a Service-Level Agreement with Buzzword.
As a corollary, no assumptions have been made regarding the maintenance of any bespoke software written for this project - primarily the 5 components mentioned in Section 1.3: Scope.

It is assumed that the restaurant will take all reasonable steps to manage access to human-system interfaces, and as such little to no security measures must be implemented in this project. Human-system interfaces here refers primarily (but not exclusively) to the hardware devices to which item NF-UR-27 applies. That is those hardware devices through which access-limited software components can be accessed. For example, the devices carried by waiting staff.

1.20 Definition of Terms

Term	Definition
Server	The hardware and software environment which will provide copies of the Kitchen, Waiting, and Customer components to connected client devices. Also provides the hardware and software for execution of the Server and Database components. Server does **not** refer to a waiter, waitress, or other member of staff.
Client device	The device being used to interact with the system. This will be the computer used by Kitchen staff, mobile or tablet device used by waiting staff, and any customer's device which has received the Customer component of the system from the server.
Local Area Network (LAN)	The collection of computer networking devices within the restaurant's premises.
Prep.	Short for Preparation. In the context of the phrase "prep. time" it refers to the time taken from starting to prepare the order/item (depending on usage) to being completed.

Chapter 2

Risk Analysis

2.1 Risk Identification

2.1.1 Background

Risk identification is the initial stage of the process. It looks to identify any possible risks that may come up during the project. We identified 4 different categories of risk that may come up.

1. **People Risks:** Risks linked to members of the project team.
2. **Technology Risks:** Risks applied to the work tools used to help create the system.
3. **Knowledge Risks:** Risks related to the background knowledge of the team members.
4. **Project Risks:** Risks related to the project itself.

2.1.2 Documentation

Risk ID	Risk	Type	Strategy Type
RK1	Team member drops out	People	Contingency
RK2	Bereavement	People	Contingency
RK3	Coursework for other modules	People	Avoidance
RK4	GitHub being unavailable	Technology	Contingency
RK5	Team Members being unavailable	People	Minimisation
RK6	Equipment/server crash	Technology	Contingency
RK7	Project is underestimated in terms of difficulty	Technology	Avoidance
RK8	Loss of Data	Technology	Contingency
RK9	Lack of Communication	People	Avoidance
RK10	Members unfamiliar with aspects	Knowledge	Minimisation
RK11	Team Members not doing fair share	People	Avoidance
RK12	Changes to Project Specification	Project	Contingency

2.2 Risk Analysis

2.2.1 Background

After we identified all the possible risks to our project, analysis of each risk can be done. Each risk can be grouped into their **probability** and **impact**.

Probability is measured by the following terms: High, Medium and Low.

Impact is also measured by the following terms: High, Medium and Low.

2.2.2 Documentation

Risk ID	Risk	Probability	Impact
RK1	Team member drops out	Low	High
RK2	Bereavement	Medium	Medium
RK3	Coursework for other modules	High	Medium
RK4	GitHub being unavailable	Low	Low
RK5	Team members being unavailable	Medium	Low
RK6	Equipment/server crash	Low	High
RK7	Project is underestimated in terms of difficulty	Low	Medium
RK8	Loss of Data	Low	High
RK9	Lack of Communication	Low	Low
RK10	Members unfamiliar with aspects	Medium	Medium
RK11	Team members not doing fair share	Low	Medium
RK12	Changes to Project Specification	Low	High

2.3 Risk Planning

2.3.1 Background

After each risk is identified and categorised, appropriate plans can be prepared for each risk.

Each risk is given the most appropriate strategy, so to not waste time selecting which one is best

2.3.2 Strategies

1. **Avoidance:** This strategy is used to reduce the chance of the risk becoming a problem at all.

2. **Contingency:** This strategy is based on if the risk becomes a reality, what is the best course of action.

3. **Minimisation:** This strategy is used to reduce the overall consequences if the risk becomes a problem.

2.3.3 Documentation

RK1	**Contingency Plan:** The workload is to be spread evenly across all members, to ensure lost work can easily be caught up on.
RK2	**Contingency Plan:** The work is to be assigned to other group members so that no work is lost.
RK3	**Avoidance Plan:** Everyone should be able to time manage effectively. If too caught up in other courses they should inform the group manager.
RK4	**Contingency Plan:** Backups should be kept on personal computers so that in the event of servers being down, files can be transferred via other means e.g. USB Stick.
RK5	**Minimisation Plan:** Team Members are to let the group manager know so that work can easily be reassigned and no work is lost.
RK6	**Contingency Plan:** Work is to be saved regularly so that loss of work is avoided in the event of any equipment crashing.
RK7	**Avoidance Plan:** Tasks should be given out to the appropriate team members. Members should also contact the manager if struggling as soon as possible.
RK8	**Contingency Plan:** Multiple members should have backups of others work in case a member loses data. All data should also be kept on GitHub as a centralized back up.
RK9	**Avoidance Plan:** Regular meetings should be scheduled to ensure all tasks are being dealt with and all questions are being answered.
RK10	**Avoidance Plan:** Tasks should only be given to the appropriate team members to ensure all work is completed correctly.
RK11	**Avoidance Plan:** Members should regularly upload documents to GitHub to show progress. The manager should keep members updated with tasks so they can complete their respective tasks on time.
RK12	**Minimisation Plan:** Work should be easily changed so that it can follow the updated to the new specification. No work should be submitted early.

Chapter 3

Project Decisions and Plan

Contents

3.1 Project Decisions

3.1.1 High-level system architecture

The system is to be designed as a web application, as per the specification document (**Specification**). As a result the system will consist of 5 main software components to be created, 3 client-side and two server-side. The client-side applications, namely the Waiting, Kitchen, and Customer components are intended to interact only with the Server component and only in a minimal fashion. It is intended that only communication will be limited to the initial loading of the client application and passing data only as required. For example, the Waiting component is intended to only communicate with the server to request an up-to-date menu, details on a specific order, and to place or amend an order. As much as possible this communication will be through lightweight AJAX and Server-Sent Events.
The server-side components consist of the code required to handle the data transfer between components and the database. The necessity of the database is debatable, but is used to reduce the load on the client devices by providing a central ACID storage location for data. The database also reduces the coupling of components to the data representation across the system whilst easing the addition of possible future functionality. For example, the recording and statistical analysis of orders placed throughout a day/week/month.

Although designed and implemented as a web application, the system is not intended to be used in any capacity over the internet. In order to ease the development process the system will be internet accessible during that period, with the intent to disallow internet connections when installing the system in a restaurant. This would be achieved through restricting access to local IP addresses through the server and networking equipment.
The reason for this is because the system as specified does not specifically require any over-the-internet features, and doing so massively reduces the security and authentication requirements for the software we are implementing. Without this stipulation, some sort of account management sub-system would need to be implemented to limit access to the Waiting and Kitchen components.

Architecture Diagram

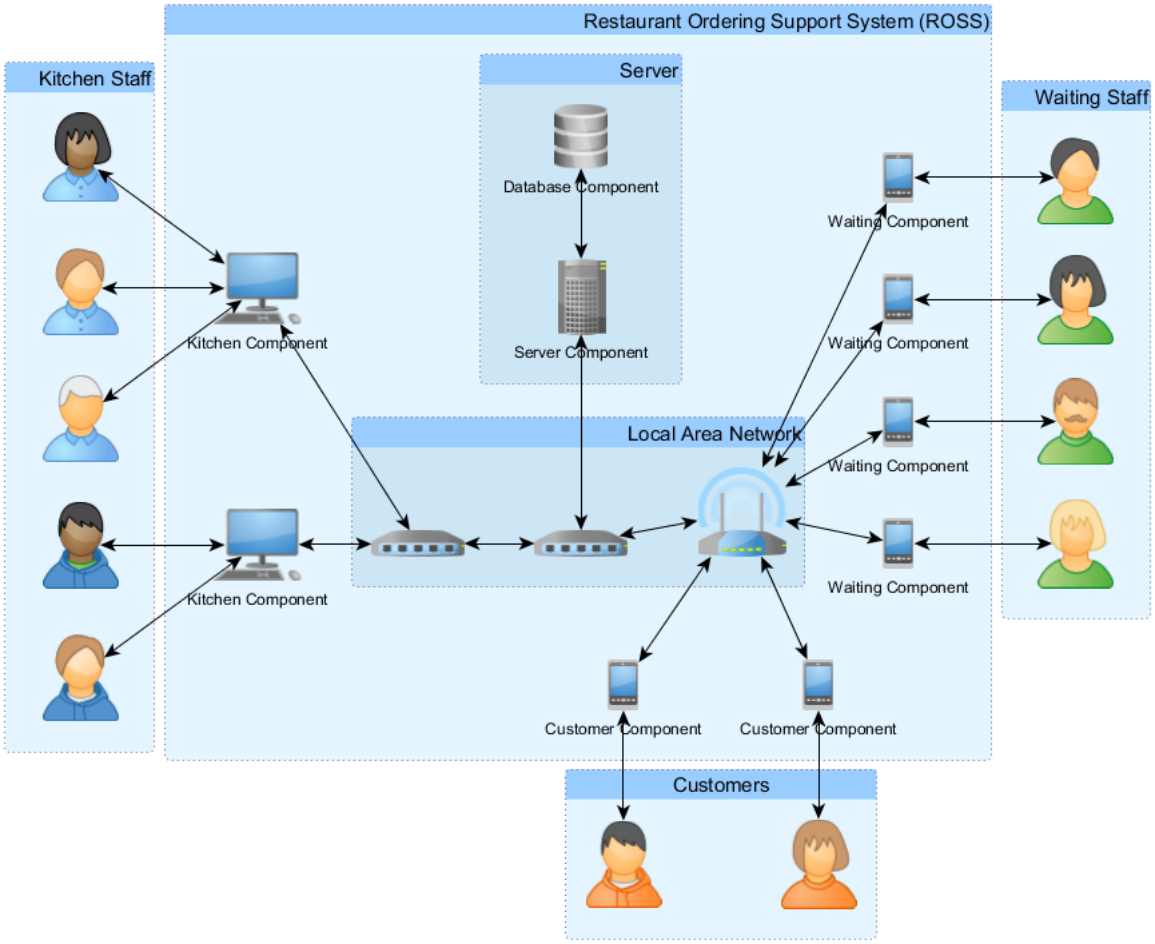


Figure 3.1: Architectural overview of ROSS.

As a web app, the exact number of devices and people connected to the Local Area Network is flexible.
No specification is made as to the Local Area Network topology.

3.1.2 Existing software and infrastructure usage

During the software development phase of the project, the software will be implemented on the existing LAMP server provided to students within the MACS department. This is to remove the need to implement and maintain a server to test on and allow us to focus on the web app itself. The same software stack would be utilised in the final product when installed within a restaurant, though not necessarily the same software versions.
This stack is chosen for its commonness and free open-source nature. This makes it cheap and easy to implement and maintain, especially when the client may not have a sizeable or experienced support system for web technologies.

In terms of other software incorporated into the product, the decision has been made to use the React JavaScript library in order to ease the development of the User Interface (UI). The library makes it significantly easier to associate functionality and states with specific visual components, bringing JavaScript much closer to an object-orientated paradigm. The library also has functionality to react to changes in state and update the relevant UI elements, implementation specifics of which are not yet fully researched. No explicit plans are made to use any other libraries or frameworks not built-in to technologies used (e.g. the standard built-in PHP library). The project remains open to the potential offered by other libraries which may come up during development. These will be judged on, amongst other things, their ease of incorporation to the existing code base and any advantages they provide in the immediate and long-term future. Specific efforts will be made to avoid incorporating too many libraries and bloating the product with largely unused code.

3.1.3 Development technologies

Whilst each team member is to be allowed to use whatever development environment they desire, specific technologies will be used to facilitate co-operation.

Git and GitHub will be used to provide strong version control and serve as a backup of all work, both complete and in progress. They also provide excellent tools to manage the integration of multiple peoples' work into a single cohesive unit.

Documentation is primarily to be written using LaTeX to take full advantage of Git. Since Git cannot work with binary files (for example PDF, DocX), the fullness of its version control cannot be exploited unless a plain-text file format is used. LaTeX is written in plain text, allowing full use of version

control. It also has the advantage of automating significant amounts of document formatting, making it easier to produce documentation in parallel and maintain a consistent aesthetic and format. It also makes the merging or documents easier.

In terms of communication within the team, WhatsApp was decided as the method of choice. This is because email was felt too slow and easily missed, and of the various communication applications out there it was the one most members had in common. For more formal communication within the team and with external parties (such as line manager), email is used. This is because, although slower, email gives a clear record of communication both in attributing statements to people and specifying who received a given message.

3.2 Risk Analysis

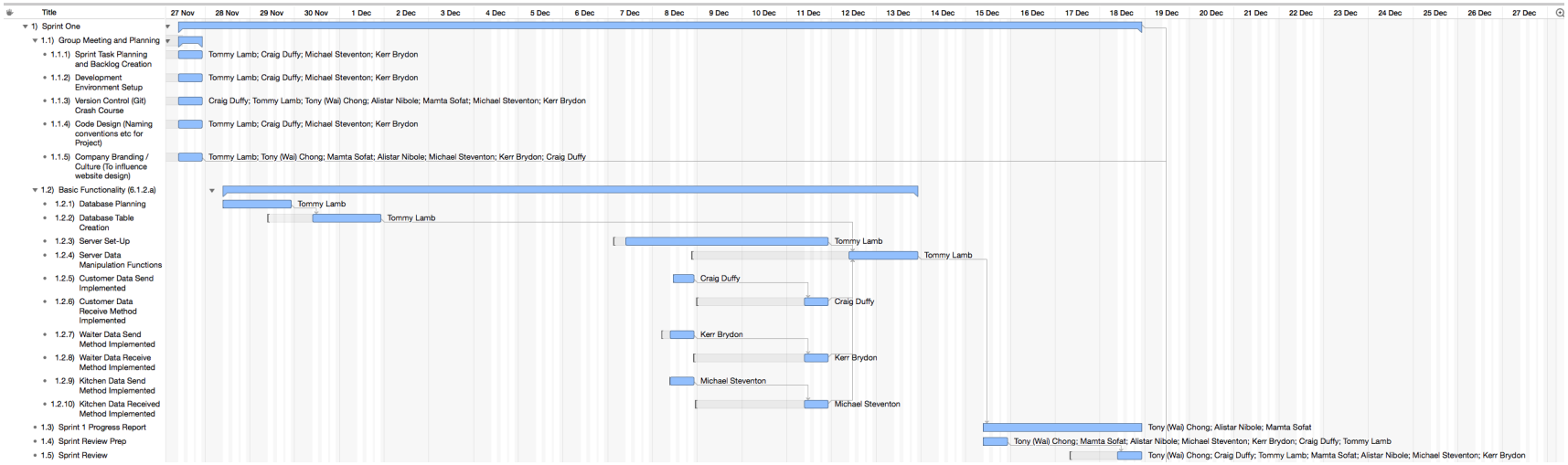
3.3 Project Plan

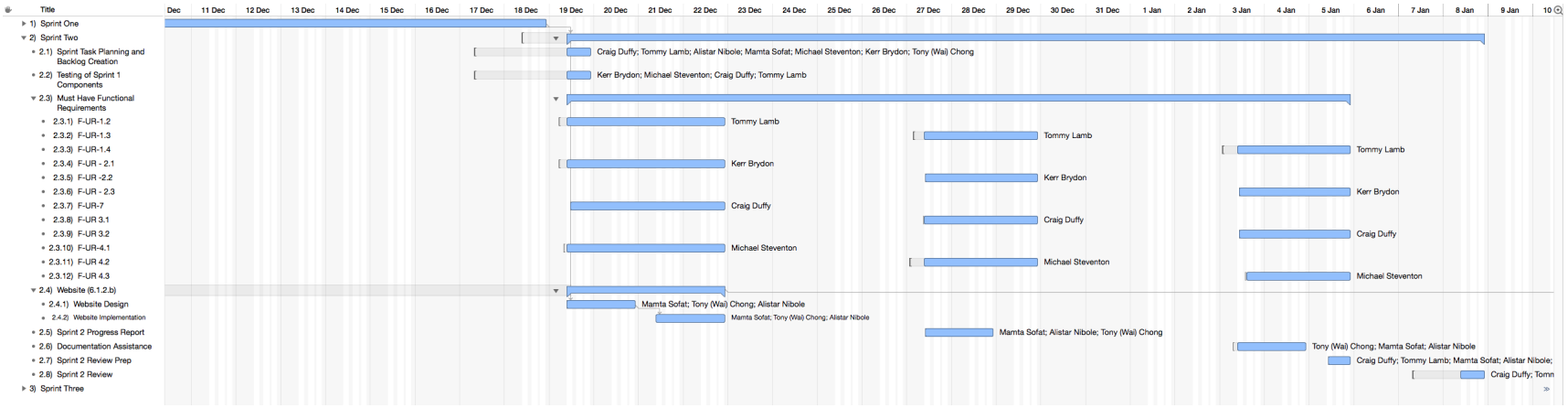
The project plan carries forward from Stage Two. It was unclear if the specification wished us to include stage one but as this document was produced during this stage it seemed illogical to do so.

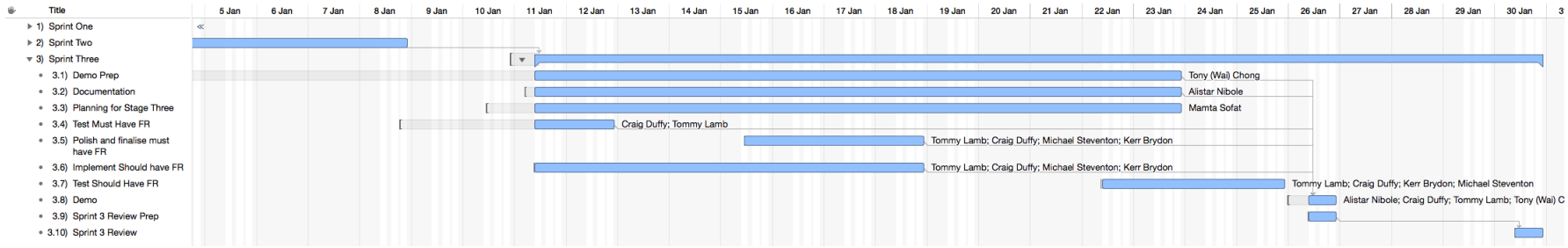
3.3.1 Stage 2

Stage Two is split into three sprints.
The first sprint follows on immediately from the submission of this document and lasts till just after exams. During this time it is expected that the core-dev team will set-up the database and server, as well as writing code for the components to talk to the server. The rest of the team will work on company branding and culture to dictate the website design.
The second sprint follows a couple days after that, and spans the holiday period. During this time it is expected that the core-dev team will begin implementing the most important functional requirements while the rest of the team works on the website.
The final sprint follows a couple days after the second sprint, and spans the first few weeks of semester two. During this time it is expected that the core-dev team will work on implementing the rest of the most important functional requirements, and have something to demo at the submission date. Meanwhile, the rest of the team will document the progress so far.

Figures begin on the next page.


Title
27 Nov
28 Nov
29 Nov
30 Nov
1 Dec
2 Dec
3 Dec
4 Dec
5 Dec
6 Dec
7 Dec
8 Dec
9 Dec
10 Dec
11 Dec
12 Dec
13 Dec
14 Dec
15 Dec
16 Dec
17 Dec
18 Dec
19 Dec
20 Dec
21 Dec
22 Dec
23 Dec
24 Dec
25 Dec
26 Dec
27 Dec
1) Sprint One
1.1) Group Meeting and Planning
1.1.1) Sprint Task Planning and Backlog Creation
Tommy Lamb; Craig Duffy; Michael Steventon; Kerr Brydon
1.1.2) Development Environment Setup
Tommy Lamb; Craig Duffy; Michael Steventon; Kerr Brydon
1.1.3) Version Control (Git) Crash Course
Craig Duffy; Tommy Lamb; Tony (Wai) Chong; Alistar Nibole; Mamta Sofat; Michael Steventon; Kerr Brydon
1.1.4) Code Design (Naming conventions etc for Project)
Tommy Lamb; Craig Duffy; Michael Steventon; Kerr Brydon
1.1.5) Company Branding / Culture (To influence website design)
Tommy Lamb; Tony (Wai) Chong; Mamta Sofat; Alistar Nibole; Michael Steventon; Kerr Brydon; Craig Duffy
1.2) Basic Functionality (6.1.2.a)
1.2.1) Database Planning
Tommy Lamb
1.2.2) Database Table Creation
Tommy Lamb
1.2.3) Server Set-Up
Tommy Lamb
1.2.4) Server Data Manipulation Functions
Tommy Lamb
1.2.5) Customer Data Send Implemented
Craig Duffy
1.2.6) Customer Data Receive Method Implemented
Craig Duffy
1.2.7) Waiter Data Send Method Implemented
Kerr Brydon
1.2.8) Waiter Data Receive Method Implemented
Kerr Brydon
1.2.9) Kitchen Data Send Method Implemented
Michael Steventon
1.2.10) Kitchen Data Received Method Implemented
Michael Steventon
1.3) Sprint 1 Progress Report
Tony (Wai) Chong; Alistar Nibole; Mamta Sofat
1.4) Sprint Review Prep
Tony (Wai) Chong; Mamta Sofat; Alistar Nibole; Michael Steventon; Kerr Brydon; Craig Duffy; Tommy Lamb
1.5) Sprint Review
Tony (Wai) Chong; Craig Duffy; Tommy Lamb; Mamta Sofat; Alistar Nibole; Michael Steventon; Kerr Brydon

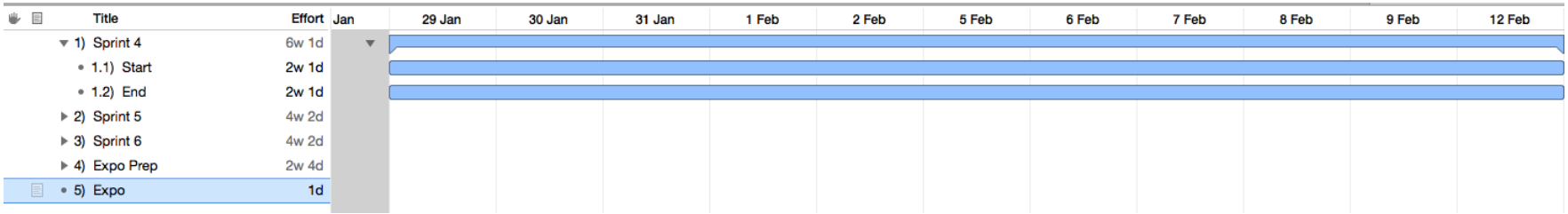

Title
Dec
11 Dec
12 Dec
13 Dec
14 Dec
15 Dec
16 Dec
17 Dec
18 Dec
19 Dec
20 Dec
21 Dec
22 Dec
23 Dec
24 Dec
25 Dec
26 Dec
27 Dec
28 Dec
29 Dec
30 Dec
31 Dec
1 Jan
2 Jan
3 Jan
4 Jan
5 Jan
6 Jan
7 Jan
8 Jan
9 Jan
10
1) Sprint One
2) Sprint Two
2.1) Sprint Task Planning and Backlog Creation
2.2) Testing of Sprint 1 Components
2.3) Must Have Functional Requirements
2.3.1) F-UR-1.2
2.3.2) F-UR-1.3
2.3.3) F-UR-1.4
2.3.4) F-UR - 2.1
2.3.5) F-UR -2.2
2.3.6) F-UR - 2.3
2.3.7) F-UR-7
2.3.8) F-UR 3.1
2.3.9) F-UR 3.2
2.3.10) F-UR-4.1
2.3.11) F-UR 4.2
2.3.12) F-UR 4.3
2.4) Website (6.1.2.b)
2.4.1) Website Design
2.4.2) Website Implementation
2.5) Sprint 2 Progress Report
2.6) Documentation Assistance
2.7) Sprint 2 Review Prep
2.8) Sprint 2 Review
3) Sprint Three
Craig Duffy; Tommy Lamb; Alistar Nibole; Mamta Sofat; Michael Steventon; Kerr Brydon; Tony (Wai) Chong
Kerr Brydon; Michael Steventon; Craig Duffy; Tommy Lamb
Tommy Lamb
Tommy Lamb
Tommy Lamb
Kerr Brydon
Kerr Brydon
Kerr Brydon
Craig Duffy
Craig Duffy
Craig Duffy
Michael Steventon
Michael Steventon
Michael Steventon
Mamta Sofat; Tony (Wai) Chong; Alistar Nibole
Mamta Sofat; Tony (Wai) Chong; Alistar Nibole
Mamta Sofat; Alistar Nibole; Tony (Wai) Chong
Tony (Wai) Chong; Mamta Sofat; Alistar Nibole
Craig Duffy; Tommy Lamb; Mamta Sofat; Alistar Nibole;
Craig Duffy; Tomn

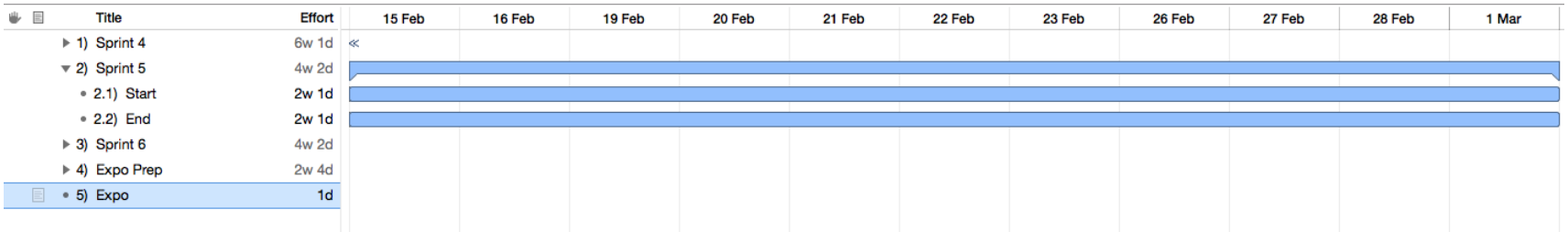

Title
5 Jan
6 Jan
7 Jan
8 Jan
9 Jan
10 Jan
11 Jan
12 Jan
13 Jan
14 Jan
15 Jan
16 Jan
17 Jan
18 Jan
19 Jan
20 Jan
21 Jan
22 Jan
23 Jan
24 Jan
25 Jan
26 Jan
27 Jan
28 Jan
29 Jan
30 Jan
1) Sprint One
2) Sprint Two
3) Sprint Three
3.1) Demo Prep
3.2) Documentation
3.3) Planning for Stage Three
3.4) Test Must Have FR
3.5) Polish and finalise must have FR
3.6) Implement Should have FR
3.7) Test Should Have FR
3.8) Demo
3.9) Sprint 3 Review Prep
3.10) Sprint 3 Review
Tony (Wai) Chong
Alistar Nibole
Mamta Sofat
Craig Duffy; Tommy Lamb
Tommy Lamb; Craig Duffy; Michael Steventon; Kerr Brydon
Tommy Lamb; Craig Duffy; Michael Steventon; Kerr Brydon
Tommy Lamb; Craig Duffy; Kerr Brydon; Michael Steventon
Alistar Nibole; Craig Duffy; Tommy Lamb; Tony (Wai) C

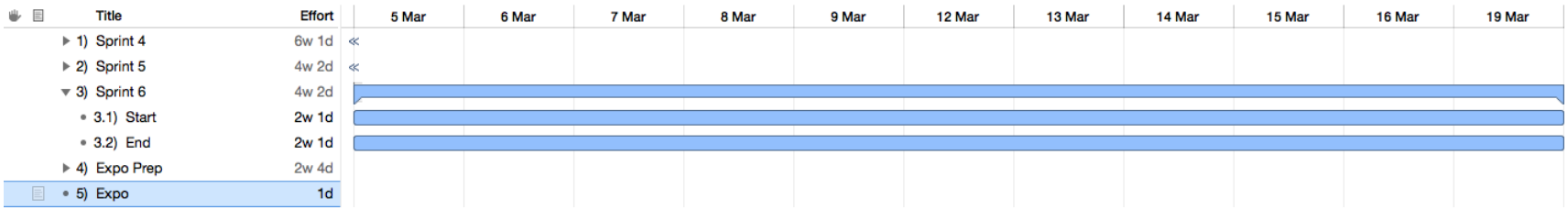
3.3.2 Stage 3

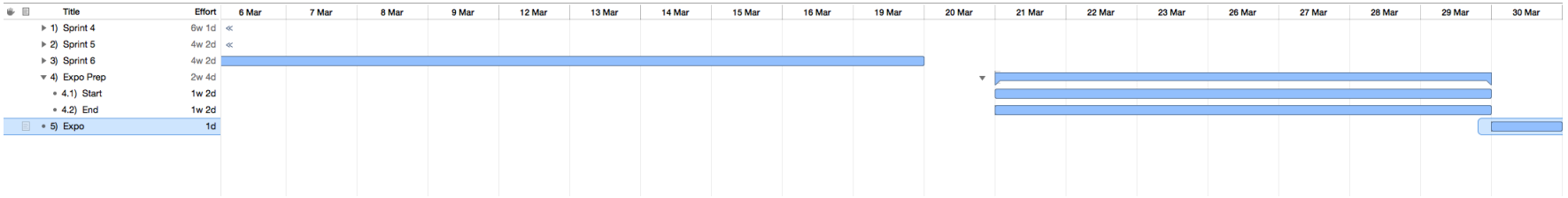
Stage three is very briefly planned with only the start and end dates of the sprints outlined.
It is unclear how far through the project we will be at this stage so it is best to allow for flexibility.

Figures begin on following page.


Title
Effort
Jan
29 Jan
30 Jan
31 Jan
1 Feb
2 Feb
5 Feb
6 Feb
7 Feb
8 Feb
9 Feb
12 Feb
1) Sprint 4
6w 1d
1.1) Start
2w 1d
1.2) End
2w 1d
2) Sprint 5
4w 2d
3) Sprint 6
4w 2d
4) Expo Prep
2w 4d
5) Expo
1d


Title
Effort
15 Feb
16 Feb
19 Feb
20 Feb
21 Feb
22 Feb
23 Feb
26 Feb
27 Feb
28 Feb
1 Mar
1) Sprint 4
6w 1d
2) Sprint 5
4w 2d
2.1) Start
2w 1d
2.2) End
2w 1d
3) Sprint 6
4w 2d
4) Expo Prep
2w 4d
5) Expo
1d


Title
Effort
5 Mar
6 Mar
7 Mar
8 Mar
9 Mar
12 Mar
13 Mar
14 Mar
15 Mar
16 Mar
19 Mar
1) Sprint 4
6w 1d
2) Sprint 5
4w 2d
3) Sprint 6
4w 2d
3.1) Start
2w 1d
3.2) End
2w 1d
4) Expo Prep
2w 4d
5) Expo
1d


Title
Effort
6 Mar
7 Mar
8 Mar
9 Mar
12 Mar
13 Mar
14 Mar
15 Mar
16 Mar
19 Mar
20 Mar
21 Mar
22 Mar
23 Mar
26 Mar
27 Mar
28 Mar
29 Mar
30 Mar
1) Sprint 4
6w 1d
2) Sprint 5
4w 2d
3) Sprint 6
4w 2d
4) Expo Prep
2w 4d
4.1) Start
1w 2d
4.2) End
1w 2d
5) Expo
1d

3.4 Definition of Terms

Term	Definition
LAMP	Stands for Linux, Apache, MySQL, PHP. A typical, fully open-source software stack for web servers.
MACS	Stands for Mathematical and Computer Sciences The phrase "MACS department" is the common-usage name for the School of Mathematical and Computer Sciences within Heriot-Watt University.
App	Abbreviation of application.

Chapter 4

Project Costing

4.1 Costing

4.1.1 Hardware

Hardware: 1. Assuming we need different type of devices for each one of our software and web developer for testing. 2. A customized desktop for kitchen to keep tracking the orders. 3. Two monitors for kitchen 4. A customized desktop for server 5. One monitor for sever 6. A POS Receipt Printer for the bill.

4.1.2 Software

Software: Licenses 2 Operating system Antivirus/Security (Business)

4.1.3 Staff

Staff: Hourly rate of pay Project Manager (£35/hour) 2 Software Developers (£20/hour) UI Designer (£17/hour) Quality assurance staff (£15/hour) Business and Marketing Staff (£16/hour)

4.1.4 Training and Support

Initial installation after delivery Require both our project manager and software developer. Traveling cost. Client's staff training Require both our project manager and software developer. Traveling cost. Maintenance First six months free. 1, 3, 5 year(s) contract after. 24/7 online support services free

4.1.5 Reserves

20% of the total estimate.

People Cost	Hours	No. of Staff	Cost per hour	Wage Per Person	Total Cost
Project Manager	150	1	£35	£5,250	£5,250
Software Developer	150	2	£20	£3,000	£6,000
UI Designer	150	1	£17	£2,550	£2,550
Web Developer	150	1	£15	£2,250	£2,250
Quality Assurance Staff	50	1	£15	£750	£750
Business and Marketing Staff	150	1	£16	£2,400	£2,400
				Total	**£19,200**

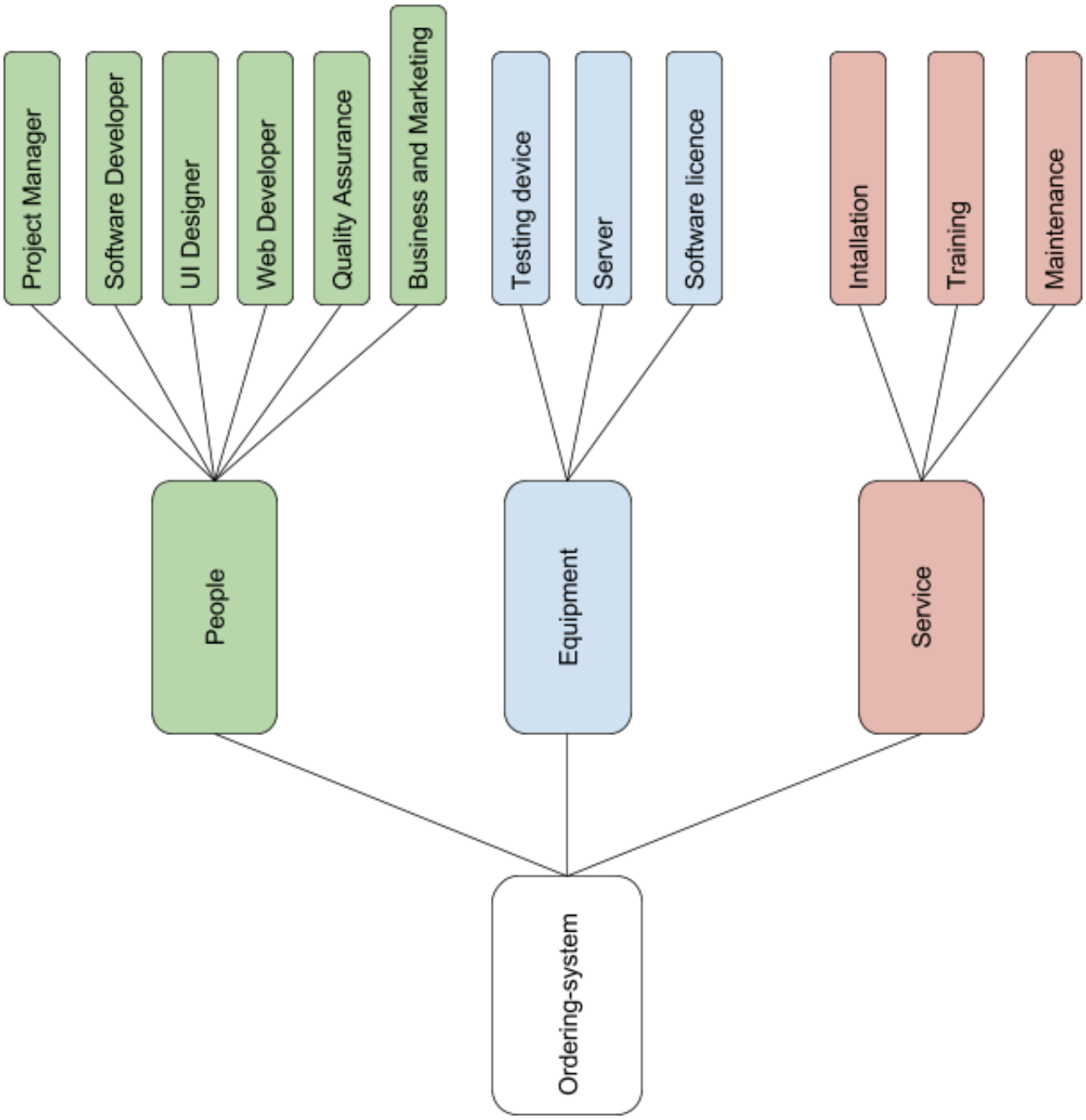
Equipment	Units	Cost per unit	Type	Total Cost
Server	1	£1,000	Bought	£1,000
IPhone 6 (testing)	4	£300	Bought	£1,200
samsung s6 (testing)	4	£200	Bought	£800
ASUS 24" Monitor	3	£155	Bought	£465
Canon iP7250 Printer	1	£40	Bought	£40
Office Depot A4 2500 sheets	1	£15	Bought	£15
Custom PC for kitchen	1	£300	Bought	£300
POS Receipt Printer	1	£69	Bought	£69
Operating system	2	£87	Bought	£174
Antivirus	2	£10	Bought	£20
			Total	**£4,083**

Service	Length	Cost per unit	Cost
Installation			
1. Project manager	3hr	£35	£105
2. Software developer	3hr	£20	£60
3. Travel	2/times	£15	£30
		Total	**£195**
Training			
1. Project manager	3hr	£35	£105
2. Software developer	3hr	£20	£60
3. Travel	2/times	£15	£30
		Total	**£195**

Support	Length	Cost
Updates/Maintain	6 Months	Free*
Updates/Maintain	1 Year plan	

Reserves	Length	Cost
20% of total estimate	20%	£23,673
	Total	**£4,734.60**

Total Project Cost	**£28,408**


Project Manager
Software Developer
UI Designer
Web Developer
Quality Assurance
Business and Marketing
Testing device
Server
Software licence
Intallation
Training
Maintenance
People
Equipment
Service
Ordering-system

Chapter 5

Usability Evaluation of Mock-Ups

5.1 Usability

5.1.1 Introduction

This document outlines the process we took in creating and executing a usability testing plan for our Restaurant Ordering Support System (ROSS). Said testing was carried out with the aim of identifying any flaws in our initial design and to gather a rough idea of user performance with the system. The system is split into 3 parts. There is the app the waiter will use to take the customer's' order through, the kitchen display for relaying the orders, and the order update system providing an up-to-date status to the customer on their order. The app will be installed on a tablet available for the waiter to use at any table. The app will allow them to input an order and request any necessary modifications e.g. those caused by dietary restrictions. The kitchen display will present a clear summary of the order to the kitchen and will allow the kitchen staff to interact by providing an estimate on the time it will take for the order to be complete. The order update system will provide the customer with a summary of their order alongside its current status, as given by the kitchen staff.
This system is designed to replace the current ordering system the client has in place and will be installed in all their locations once complete.

5.1.2 Test Goals

The objective is to test the usability of our ordering system and whether it is preferable to the standard ordering system currently in place at the restaurants. We need to know if the essential actions of the system can be carried out easily by various users if the system is to be successful. We also aim to prove the 3 following hypotheses:
1. A mobile application is easy for a waiter to place orders through.
2. A kitchen display is easy for staff to monitor and interact with.
3. Providing the customer with constant feedback on their order leaves them feeling more satisfied.
If all three hypotheses are proven true, then we believe we have created an ideal system to be placed into use in all of our clients restaurants.

5.1.3 Participants

We expect to have between six and eight participants in our study. The subjects should have a range in both age and computer literacy. We are looking for varied participants as there is no one type of customer that may enter the client's restaurants. Customers may come from any background, therefore our system must be easily usable by all.

The participants will be asked to complete one questionnaire before and another questionnaire after they have attempted to perform multiple tasks using the mock-up UI of the system we have created. The participants will not have any prior knowledge as to the workings of the system other than a quick briefing before carrying out the tests, and will have an investigator present to answer any questions they may have, alongside providing any potential assistance required. All subjects will be required to sign a consent form prior to their participation in this study.

5.1.4 Experiment Design

Participants will be asked to complete a series of tasks on each part of the system. These tasks are representative of what we believe will be some of the most common actions taken by users once the system has been fully implemented. These ideas were extracted from our requirements document to help us identify which features to test most strenuously.
However, our prototype is nothing more than a shell user interface. Therefore we cannot test certain requirements such as 'F-UR-1.5 Forming and Placing Order for Fulfilment'. Since the three sections of the system are not interlinking in the prototype, we cannot currently test the successful processing of orders and their recording in the system. The confirmation of a successful order would be its appearance in the kitchen display.

Tasks (Wait Staff App)

1. Navigate to the main course menu.
2. Add 'Spaghetti Bolognese' to the order.
3. Add 'Ice Cream' to the order with amendment 'Extra Scoop'.
4. Find what soft drinks are for sale.
5. Place the order/check out.

Tasks (Kitchen Display)

1. Check status of table 2.

Tasks (Customer Feedback Display)

1. Enter an Order Reference.
2. Check Order Status.
3. Summon Waiter to request amendment to an order.

5.1.5 Findings

There were five (5) participants in this study.

Pre-questionnaire Results

1. How often do you visit a restaurant?

Daily	0
Few times a week	1
Once a week	2
Once a month	2
never	0

2. Have you ever had difficulty placing an order in a restaurant?
   80% No, 20% Yes

3. What common issues have you come across in restaurants?

   - Poor Service
   - Slow service bringing food. Taking a long time to bring the bill.
   - Rude Staff. Long Waits.
   - Bad service and bad food.
   - Forgetting parts of orders.

4. Have you ever ordered food online through a website or an app?
   Yes 100%

5. Did you mostly enjoy the service?

Yes	3
Mostly	1
No	1

6. What didn't you enjoy about the experience?

   - The wait.
   - Slow Delivery.
   - No way to know order has been prepared correctly until it arrives.
   - Long delivery times
   - Lack of information about the food

7. Do you prefer making purchases face-to-face or online?

   - I enjoy both

- Face-to-Face as you can ask questions, it is usually quicker. Makes the experience more enjoyable (sometimes).
- Face-to-Face
- Both are fine.
- Online

Test Results

1. Navigate to the Main Course Menu.
   All Participants managed this task without issue.

2. Add 'Spaghetti Bolognese' to the order.
   All Participants managed this task without issue.

3. Add 'Ice Cream' to the order with amendment 'Extra Scoop'.
   A couple of participants hesitated as they were unsure whether to enter the amendment in the text box before or after they selected the item.

4. Find what soft drinks are for sale.
   All Participants managed this task without issue.

5. Place the order/ check-out.
   Some participants took a few seconds to find the checkout button but all participants completed the task.

6. Check status of table 2.
   Lots of uncertainty on what could actually be done and what was on the display with only one participant correctly pressing the status button on the first attempt.

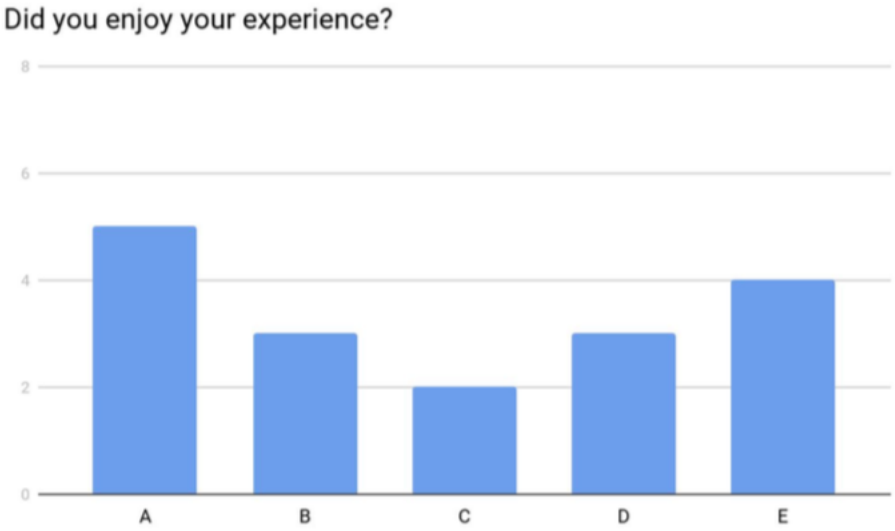
7. Enter an Order Reference
   Confusion at the appearance of the order reference. However, all participants were able to identify the correct steps to take.

8. Check Order Status
   Mostly understood, some were unhappy with the lack of clarity in the summary.

9. Summon waiter to request amendment to the order.
   All participants managed this without an issue.

Post-Questionnaire Results

1. How each participant (A - E) Rated Their Experience



2. Did you find the system intutitive to use? (i.e. Did all items appear under the correct headings/ menus you expected them to?)

Extremely Intuitive	0
Very Intuitive	0
Somewhat Intuitive	3
Not Very Intuitive	2
Not at all intuitive	0

3. Did you find yourself relying on the investigator to help often?

Very Often	1
Quite often	2
Occasionally	1
Rarely	1
Never	0

4. Are you satisfied with the look of the system?

Extremely Satisfied	0
Very Satisfied	0
Somewhat Satisfied	2
Not Very Satisfied	2
Not at all Satisfied	1

5. Would you like to use a system like this one in a restaurant?

    - Yes, it appeared fairly straightforward.
    - I would if it was programmed correctly and everything was visible + simple to use.
    - Not really. It seemed to over complicate the whole thing
    - Maybe, if some improvements were made.
    - The Idea is good. Should be a good system once errors have been fixed.

6. Do you have any thoughts on how to improve the system?

    - Tighten up on Presentation. A few careless errors with spelling and layout.
    - Spell things correctly. A feature that lets you know how long your food will be. A chance o get drinks refills easily.
    - Make things clearer and simpler. Too many parts are confusing
    - Provide more options. Provide some more explanation for things like the kitchen orders.
    - Too complex for how little it does

5.1.6 Conclusions

Features

All but one of the participants pointed out the lack of features of the system in its current state, questioning the viability of such a system. Although all who gave such feedback did think the idea had potential, if only the system had more capabilities.
In this case a repeat of the testing, featuring a prototype containing all the must have requirements, will assist us in coming to a conclusion on the quality of our system. This initial test has shown that while we have some promising ideas, we need more if we are to create a complete and successful system.

Navigation

Both the wait staff app and the customer feedback display were found to be very quick and simple to browse by all participants, but they all had struggles with navigating the kitchen display. The confusion stemmed from not being able to tell what was and was not a button. The assortment of bright colours with no explanation also proved to be misleading. Overall the layout proved to be a mess of colours and squashed together tables.
The layout proved to be clear for two thirds of the system and should provide a suitable framework to build upon from here. The kitchen display will absolutely need to be overhauled for future work. In its current state it is disastrous and borderline unusable. Without significant change it could become the achilles heel of the project.

Visuals

The main positive that was repeated about the look of the system was the lack of clutter on screen. Another negative point that was shared by multiple participant was that colours were not used very well, with a lack of colour and life in some sections, and the overdone and unhelpful use of colours in others getting in the way of functionality. The kitchen display again came in for heavy criticism in this aspect. The wait staff app had mixed feedback. The participants liked the clarity of all objects but found it very bare looking, due to lack of options, colours, and images.
The feedback from the participants has shown that there is improvement to be made to make the system much more visually appealing. There is a balance that needs to be met however, as shown by the contrasting feedback on the separate parts of the system. Steps must be taken to make the system look more professional and this will take more than a splash of colour here and there.

5.2 Mock-Ups

These interfaces are meant for print outs only. There's not much interaction using proto.io. The explanation of screens given below.

5.2.1 Customer Interface

Customer places order by mentioning their order reference number in the text box and sends the request for the order, they can view their order and then call for the waiter who will place the order.

5.2.2 Waiter Interface

1. Table Status [WScreen1]
Red: Empty
Blue: Table order received when they clicked 'call waiter'
Yellow: 'Edit Order' request from the table
Green: Table served
2. Taking orders [WScreen 2]: a. Click table 2: See their order summary, confirm to place order. i. Confirming order: Click 'Place order' that leads to 'Pay' and execute payments. Turns the table 2 button green. b. Click table 3: i. Editing order: click 'Edit order' that leads to 'menu'. Tick/Untick options and press 'Check out'. Leads to order summary and then waiter confirms it and places order leading to steps mentioned in part a(i). [Menus in pdf files] c. Payment: Can be done in cash or by card. With card they can swipe it and the waiter can print receipts for the customer

5.2.3 Kitchen Interface

One screen with Order Status and 2-4 Screens with Order details. I tried keeping it very simple like the orders that have been served they are not visible anymore so as to keep minimum confusions in a busy environment. List of colour codes: 1. Nothing ordered: white 2. canceled order/empty tables: red 3. Getting prepared: yellow 4. Order edit: orange 5. Order sent/table served: blue and sky blue 6. Table being served: grey