

# Assessment 1: Software Requirements Specification

Software Architecture & Design  
Semester 1 2020

Luke McWha & Cory Mifsud  
#101 092 956      #101 484 964

## Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Project Background</b>	<b>3</b>
2.1. Overview	3
2.2. Goals and Objectives	3
2.3. Domain Vocabulary	3
<b>3. Requirements</b>	<b>4</b>
3.1. Functional Requirements	4
3.2. Assumptions	4
3.3. Scope	4
<b>4. Problem Domain</b>	<b>5</b>
4.1. Pain Points	5
4.2. Domain Entities	5
4.3. Actors	5
4.4. List of Tasks	5
4.5. CRUD Check	6
4.5.2. Missing Tasks / Justification	7
4.6. Entity Relationship Model	8
4.7. Workflows	9
4.7.1. Online Menu Viewing	9
4.7.2. Payment Processing	9
4.7.3. Ordering	10
4.7.4. Online Reservation Creation	10
<b>5. Task &amp; Support</b>	<b>11</b>
5.1.	11
5.2.	11
5.3.	12
5.4.	12
5.5.	13
5.6.	13
5.7.	14
5.8.	14
5.9.	15
5.10.	15

5.11.	15
<b>6. Quality Attributes</b>	<b>17</b>
6.1. Usability	17
6.2. Correctness	17
6.3. Modifiability	18
6.4. Reliability	18
<b>7. Possible Solutions</b>	<b>19</b>
7.1. Solution 1 - Remote Infrastructure	19
7.1.1. Overview	19
7.1.2. Advantages	19
7.1.3. Disadvantages	19
7.2. Solution 2 - Local infrastructure	20
7.2.1. Overview	20
7.2.2. Pros:	20
7.2.3. Cons:	20

## 1. Introduction

This specification document has been created to define the software requirements, provide an outline and scope for this software project. This document will cover a range of points on the project lifecycle and intends to define all functionality, assumptions, problem areas, tasks, relationships and possible solutions. Further, this document will incorporate thoughts and a priority of attributes that should be taken into consideration in planning and development.

## 2. Project Background

### 2.1. Overview

The Cuddly Wombat (client) is a café/restaurant that is planning to expand its business. In expanding, it will go from a capacity of about 50 people, to allow the cafe to host approx 150 customers.

The Cuddly Wombat's system is lowtech and requires many hours of manual labour/ intervention. The tasks completed include taking orders from guests, passing the orders on to the kitchen, accounting etc. To further expand the business the Cuddly Wombat realise they need to improve their processes which includes the introduction of new software to streamline these processes.

### 2.2. Goals and Objectives

The new system will support a number of features: Table reservations, taking customer orders, informing the kitchen about the orders, informing staff about customers who have waited a long time for their order, customer receipts, handling payments. There will also be an online ordering solution incorporated for takeaway meals. These aspects combined complete the goal of scaling the cafe's processes to accommodate more customers.

### 2.3. Domain Vocabulary

Below defines various words used throughout this document:

Word	Definition
Quality Attributes	Aspects, when considered together, that define the quality of the project.
CRUD	Create, Read, Update, Delete function for a database
FOH	Front of House cafe staff members
Front of House	Staff members who interact with customers
Accounting processes	The movement of money from the customer to cafe.
Standard Web Search	Searching for website through a search engine eg. Google
Standard system data exchange	Data (such as an order) moving from one section of the Applications to another section
Standard Database saving	Saving data from the application to an external data store
UX	User Experience
UI	User Interface

## 3. Requirements

### 3.1. Functional Requirements

- Create a table reservation for a customer
- Create a customer order
- Create an online customer order for takeaway
- Notify and give details of a customer's order to the kitchen
- Notify front of house staff whether a customer has waited a long time
- Create customer receipts for a customer's order
- Handle monetary transactions between a customer and the cafe
- Logging data to a database

### 3.2. Assumptions

- When paying the customer will interact with a staff member to fix up payment.
- Accounting processes are external to the system.
- External entities processing speed will be adequate for eftpos purposes.
- Website does not facilitate remote orders, only viewing of the menu.
- Required hardware will be procured.
- System will not have more than 150 concurrent users.

### 3.3. Scope

The system shall support reservations, taking orders from customers, informing the kitchen about these orders, informing the staff about customers who have had to wait for longer than a predetermined time for their meal (to allow these customers to be reimbursed), creating invoices and receipts for customers, and handling payments. The gathering of statistics will be incorporated to give a better overview of menu items customers are ordering. Menus will be available online to inform customers of the food that they offer and allow customers to order from a takeaway menu.

## 4. Problem Domain

### 4.1. Pain Points

- Manual work
- Current processes are not scalable
- Speed of orders
- Accuracy of orders
- No way to record

### 4.2. Domain Entities

- Menu Item - A single drink or dish that a customer can order.
- Menu - A set of menu items. Presented on the website.
- Order - A set of menu items ordered by a customer.
- Invoice - Entails the price, quantity, and table details of an outstanding order.
- Table - Corresponds to a physical table in the restaurant.
- Table Assignment - Corresponds to a customer's occupation of a table. A weak entity between tables and reservations, differs by having allocated customers in time periods and associated orders.
- Reservation - A future table assignment allocation.
- Payment - A completed payment process corresponding to an invoice.

### 4.3. Actors

There are 5 actors in this system. Each of these interact with the system in a different way. These actors include:

- Customer
- Kitchen
- FOH Staff
- External Banking/Accounting System
- Statistic Database

### 4.4. List of Tasks

- Make reservation through website
- Manage reservation
- View menu through website
- Manage menu through website
- Generate invoice for order
- Make payment on invoice and generate receipt.
- Allocate customer cover to table.
- Place or change order on table.
- Log information to database.
- Notify FOH staff on order wait exceeded.
- Receive and action order.

## 4.5. CRUD Check

Task/Entity	Order	Invoice	Payment	Table Allocation	Table	Reservation	Menu	Menu Item
Make reservation through website						C		
Manage reservation						RUD		
View menu through website							R	R
Place order	C			R				R
Generate invoice for order on table	R	C						
Receive and action order	R							
Make payment on invoice		R	C					
Allocate customer cover to table				CR	R	R		
Log order statistics	R							
Notify FOH staff on order wait exceeded	R							

Task/Entity	Order	Invoice	Payment	Table Allocation	Table	Reservation	Menu	Menu Item
Manage Menu							U	CUD
Missing?		UD	UD	UD	CUD		C	

#### 4.5.2. Missing Tasks / Justification

As shown above, Table Allocations and Tables are missing Update and Delete, and Create, Update and Delete operations respectively. These are required to support movements of customers and changes in the restaurant's layout. This results in new tasks:

- Move or remove customer table allocation.
- Edit table layout.

There will never be more than one menu, and it will never be deleted, so create and delete operations are unneeded.

Invoices and payments are immutable records and thus do not require update or deletion operations. In the event of a customer refund, new records are created to balance the old ones.

The lack of a read operation on payments highlights a need for a new task to view historical payment data.

- View payment history.



## 4.6. Entity Relationship Model

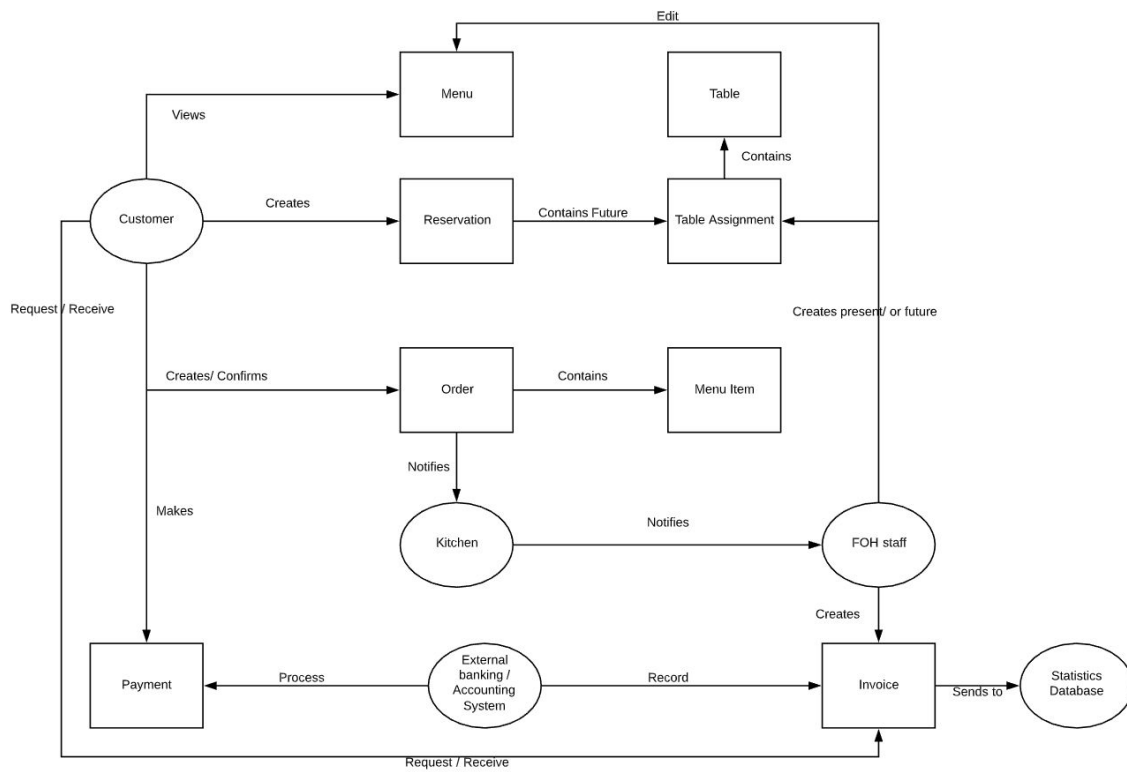


Fig. 1, ERM

## 4.7. Workflows

### 4.7.1. Online Menu Viewing

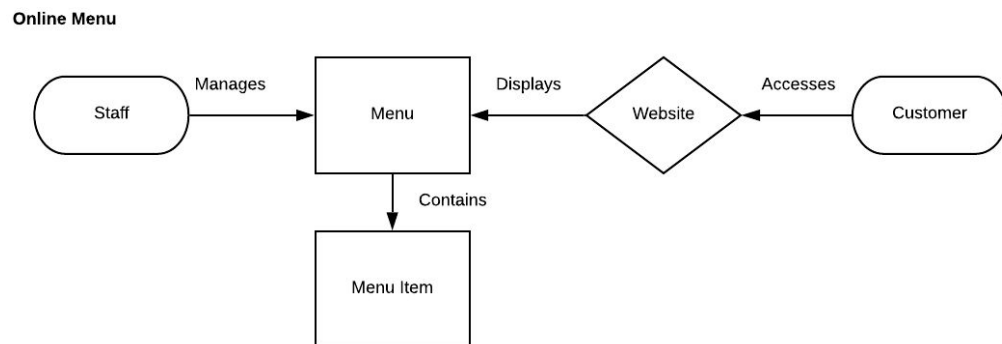


Fig. 2, Workflow 1

### 4.7.2. Payment Processing

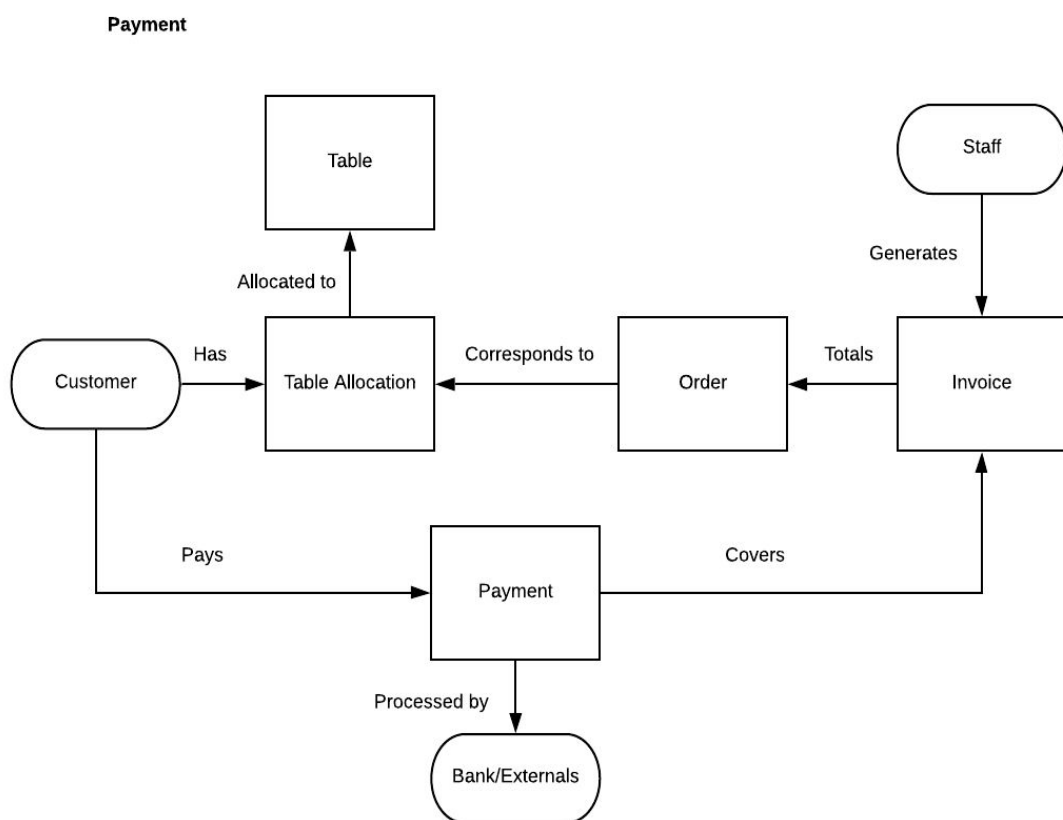


Fig. 3, Workflow 2

### 4.7.3. Ordering

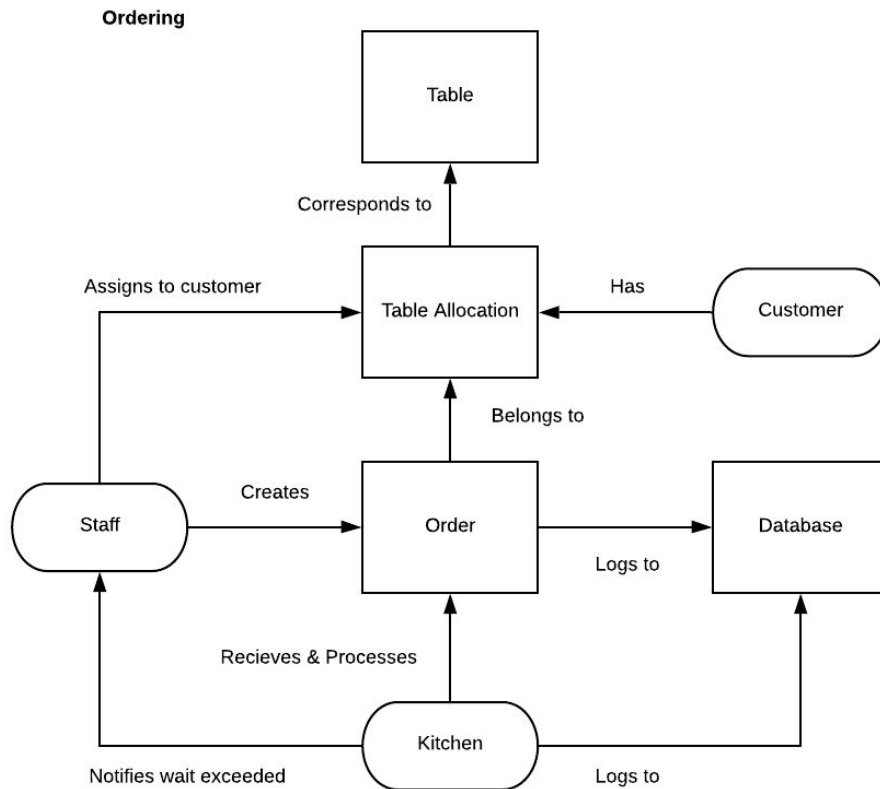


Fig. 4, Workflow 3

### 4.7.4. Online Reservation Creation

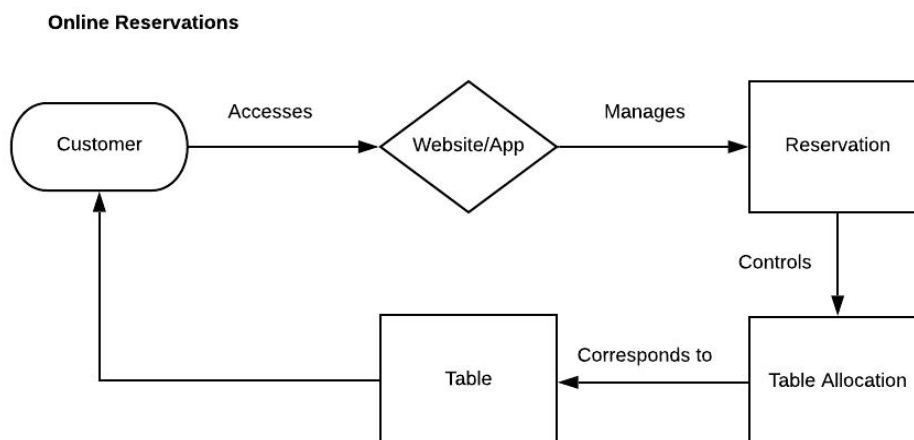


Fig. 5, Workflow 4

## 5. Task & Support

### 5.1.

<b>Task</b>	Make table reservation
<b>Actor</b>	Customer
<b>Purpose</b>	To reserve a table at the venue
<b>Trigger/ Precondition</b>	Customer decides to make a reservation
<b>Frequency</b>	High
<b>Critical</b>	Medium
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Find website	(Standard web search)
2. Create reservation <b>Problem:</b> Cannot find reservation section on website	Have good UI design to make finding the section easy.
3. Enter requested reservation details	(Standard data entry)
4. Confirm reservation <b>Problem:</b> Customer does not have email	Allow for mobile/text confirmation.
<b>Variants</b>	
1a. Customer books over the phone 1b. Customer books in person	Staff members may process the booking.

### 5.2.

<b>Task</b>	Views menu through website
<b>Actor</b>	Customer
<b>Purpose</b>	See the cafe's menu
<b>Trigger/ Precondition</b>	Customer decides to look at menu
<b>Frequency</b>	High
<b>Critical</b>	Low
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Find website	(Standard web search)
2. Find menu on the website	Ensure a good UI/UX to guide the user.

<b>Problem:</b> Cannot find menu on website	
3. Open menu to be viewable <b>Problem:</b> User does not have a PDF viewer	Embed a PDF viewer within the website itself to present the menu; Assume the user is using Chrome, Edge or Firefox and use the native PDF viewer in the browser;

## 5.3.

<b>Task</b>	Place order
<b>Actor</b>	Customer/Staff
<b>Purpose</b>	Place an order for the cafe to make
<b>Trigger/ Precondition</b>	Customer decides what they would like.
<b>Frequency</b>	Very High
<b>Critical</b>	Very High
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Views items on menu	
2. Add menu items to the order	Click button on page
3. View items for order	
4. Confirm Order and send to kitchen	Click button on page
<b>Variants</b>	
3a. Change order 4a. Change of mind, cancel order	Items can be removed from the order. Orders can be cancelled.

## 5.4.

<b>Task</b>	Manage menu through website
<b>Actor</b>	Staff / Management
<b>Purpose</b>	Update menu
<b>Trigger/ Precondition</b>	Management want to change cafe menu
<b>Frequency</b>	Low
<b>Critical</b>	Moderate
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Management login to management portal	Access admin login screen through website

2. Navigate to the “Update Menu” page	
3. Make menu changes	Be able to edit text on a menu document
4. Save menu changes	Button for user to click to save changes

## 5.5.

<b>Task</b>	Generate invoice for order on table
<b>Actor</b>	External Banking/Accounting System
<b>Purpose</b>	Generate invoice for an order
<b>Trigger/ Precondition</b>	Customer paid for order
<b>Frequency</b>	Very High
<b>Critical</b>	Very High
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Get the customer order details	(Standard system data exchange)
2. Sends order data to statistics database	(See task: “Log order statistics”)
3. Create invoice based on the order details	System adds order detail to invoice template
4. Presents invoice to FOH and Customer <b>Problem:</b> User does not agree with the invoice, or there is a mistake.	Displays invoice on screen

## 5.6.

<b>Task</b>	Receive and action order
<b>Actor</b>	Customer/Staff/Kitchen
<b>Purpose</b>	Move order to kitchen to be made
<b>Trigger/ Precondition</b>	Completes order and confirms
<b>Frequency</b>	High
<b>Critical</b>	High
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Order is confirmed	Button to confirm order
2. Present order to the kitchen	(Standard system data exchange)

3. Action the order	Order appears on screen in kitchen to be made
---------------------	---

## 5.7.

<b>Task</b>	Make payment on order
<b>Actor</b>	Customer
<b>Purpose</b>	Pay for order made
<b>Trigger/ Precondition</b>	Customer contacts staff to pay
<b>Frequency</b>	High
<b>Critical</b>	High
<b>Sub Tasks</b>	<b>Example Solution</b>
1. View generated invoice	System displays invoice on a screen
2. Exchange money to pay for invoice	Through cash or card payment
3. Confirm and verify payment	A prompt comes on the screen asking whether complete payment has been made
<b>Variants</b>	
2a. Cash payment	Till is opened and cash is received by customer Change is given if required
2b. Card payment	Card payment is made on an EFTPOS machine. Transaction is sent to the external accounting system. Confirmation of payment is given by an external accounting system.

## 5.8.

<b>Task</b>	Generate receipt for payment
<b>Actor</b>	External Banking/Accounting System
<b>Purpose</b>	Give customer receipt for purchase
<b>Trigger/ Precondition</b>	Payment for order is made
<b>Frequency</b>	High
<b>Critical</b>	High
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Order invoice is created	(See Task: "Generate invoice for order on table")

2. Print receipt	Receipt for payment is printed.
<b>Variants</b>	
1a. No receipt required	Receipt printing is optional.

## 5.9.

<b>Task</b>	Allocate customer cover to table
<b>Actor</b>	FOH
<b>Purpose</b>	Allocate number of people on table
<b>Trigger/ Precondition</b>	Customer is seated at table
<b>Frequency</b>	High
<b>Critical</b>	Low
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Declares number of customers on table	FOH clicks on table in app and fills out form fields
2. Allocates FOH staff member to service the table	FOH adds team member to service table through drop down list of active team members

## 5.10.

<b>Task</b>	Log order statistics
<b>Actor</b>	External Banking/Accounting System
<b>Purpose</b>	Log data for statistical analysis
<b>Trigger/ Precondition</b>	Receipt created for order
<b>Frequency</b>	High
<b>Critical</b>	Moderate
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Receive invoice details	(See Task: "Generate invoice for order on table")
2. Save invoice data between respective data tables	(Standard Database saving)
3. Send response that data has been logged	A response indicating that data has been logged.



## 5.11.

<b>Task</b>	Notify FOH staff on order wait exceeded
<b>Actor</b>	Kitchen
<b>Purpose</b>	Notify FOH on long order wait time
<b>Trigger/ Precondition</b>	Order timer exceeds certain time
<b>Frequency</b>	Moderate
<b>Critical</b>	High
<b>Sub Tasks</b>	<b>Example Solution</b>
1. Sends out notification of order time	Automatic when timer reaches certain time
2. Receive detail that order time has expired	Popup on the screen indicating the order waiting time.
3. Offer customer reimbursement	FOH offer some form of goods as reimbursement

## 6. Quality Attributes

### 6.1. Usability

Usability is a core attribute that will define the quality of this product. The software is to be used in a fast paced cafe thus human-understanding caused errors should be minimized.

The main aspects of usability in this scenario are: User Experience (UX), User Interface (UI) and Performance.

User Experience refers to the workflow and use case adherence of the software. It focuses on how the software will be used while ordering a meal, staff are taking payment or the kitchen is receiving orders. User experience meeting user expectations reduces the risk of errors.

User Interface refers to the graphical representation of the application interface. Good UI design promotes a pleasant experience and subtly highlights component importance hierarchy, reducing user frustration and improving usage fluidity.

Performance refers to the responsiveness of the application during use. In a fast paced, real-time environment low performance results in low customer satisfaction and loss of profit due to late and comped meals, and frustrated staff due to an unresponsive experience. Simple navigation events should be near-instant, while heavier processes should not take longer than 2 seconds to complete. Low frequency processes may exceed this limit.

### 6.2. Correctness

Correctness is another core attribute that must be considered throughout this project. The key areas where correctness must be adhered are:

1. The customer ordering processing, the kitchen receiving the order.

Incorrect orders result in angry customers, longer waits and lost profits while the correct orders are prepared.

2. The customer paying the correct amount for their order with an accurate invoice.

An incorrect invoice amount or payment process can result in over or under charging one or more customers, and undesirable audits.

3. Reservation timing and accuracy.

Clashing or null reservations may result in over or under booked tables, resulting in customer dissatisfaction or loss of profit.

These scenarios all affect both customer and company satisfaction and must be taken seriously.

### 6.3. Modifiability

The system must be designed with expansion in mind. The owners considered features outside those defined in the current scope which may be incorporated into a second phase. Thus to avoid unnecessary blow-out of project resources, the system must be built in a modifiable manner.

Scalability also falls under modifiability. It is not unreasonable to expect that in the future the cafe may expand and require the ability to handle more customers, tables, restaurant locations, menu selections etc. as it has on this occasion.

Interoperability also falls under this category somewhat. Future integration with common 3rd party entities such as MenuLog, UberEats is to be considered throughout the design phase.

### 6.4. Reliability

A full, non-recoverable failure during business hours would majorly hamper earning potential and dissatisfied customers. The system must function as intended for 98% of opening hours. In the case of operational failure, the system must be able to recover functionality within 3 minutes, without loss of important data such as invoices.

## 7. Possible Solutions

### 7.1. Solution 1 - Remote Infrastructure

#### 7.1.1. Overview

Our recommended solution involves a web-app acting as a single source of operation. A number of tablet devices are dispersed throughout the cafe for staff use.

Upon arrival of a customer, a staff member opens and logs in to the app, which defaults to a cafe layout view. They select a table suitable for the customers, and assign a customer cover to the table.

When the customer would like to order, the staff selects the table to navigate to an order screen. They then input the menu items and confirm the order, notifying the tablets in the kitchen that a new item has arrived.

The kitchen staff, who have selected the 'Current Orders' view from the app menu, can then set the state of the order, and view other orders in the queue similarly. When too much time elapses without a state change, a front-of-house device notifies the staff members to attend to the waiting table.

Upon completion of their meal, the customer moves to the till, where the staff member selects their table and presses the invoice button, displaying an invoice on the screen. The customer can then pay with cash, or with the attached (or built in) EFTPOS utility on the tablet. The customer can decide whether to print a receipt or not. Either way, the interaction is recorded in the application's data-store.

The staff can then visit 'History' in the app menu to view order statistics and historical payment data.

The staff can also select 'Update Menu' in the app menu, and make the required changes which are pushed to the customer-facing website.

The customer website will contain three main pages:

1. An attractive homepage.
2. A menu page.
3. A reservation page.

The reservation page allows customers to leave their details and a preferred reservation time, which will notify the staff on a tablet, who can review and either accept or reject reservation requests under the 'Reservations' menu. The customer will then be notified of the result via their contact details.

#### 7.1.2. Advantages

- Single point of development. Flow of data clearer.
- Less redundancy in processes (eg. changing menus)

#### 7.1.3. Disadvantages

- Control of data deferred to external party. Reliant on availability.
- Reliant on connection to the internet. Service drops halt functionality.

## 7.2. Solution 2 - Local infrastructure

### 7.2.1. Overview

This solution further removes staff intervention and interaction with the customer moving towards a more complete technology approach.

The customer can access the cafe's menu through the website and can reserve a table at the cafe. Through entering the customers name, email, phone number, selecting a time and how many people will be on the table, they will then create a reservation and get a confirmation email for that reservation.

When a customer arrives at the cafe they are greeted by a FOH staff member who will show them to their table if they have a reservation or allocate them a table if they don't.

Each table will have an integrated iPad or Tablet device which will act as their menu. Here a customer can look through menu items and add items to their order. When they are happy with their order they can press 'Confirm' and this action will send the order to the kitchen to be made.

A timer is kept with each order to track the length of time it takes to get the order to the customer. If this timer elapses a certain time, FOH staff will be notified to compensate the customer for this wait time. When an order is made by the kitchen, they will notify staff to take the order to the customers table.

When the customer is finished, they will go to the front desk where they will make a payment. The system will ask whether the payment is cash or EFTPOS. It will process the payment and ask whether the customer wants a receipt.

Certain staff will be able to access the admin portal of the website. Here they can change the viewable menu on the website.

The staff can also select 'Update Menu' in the app menu, these changes are device only and will need to be done for each table device.

The staff can also visit 'History' in the app menu to view order statistics and historical payment data.

### 7.2.2. Pros:

- All internal processes are handled internally. Removes reliance on service providers for core business functionality.
- Customers handling orders themselves removes reliance on staff, a limited business resource.

### 7.2.3. Cons:

- Mix of tech, part online part local, increase in complexity.
- Requires investment in local infrastructure.
- Customers can't be trained to use software.