# Schedule & Notes

Evaluation

Inform the details for the evaluation

Transcribe the video so that we cant see the face

Give a few promiting questions necause the interviewee will not speak

Think about what interfaces to evaluate depending on their expiernce

Cognitive walkthorhg usability expert not user s

Consent from users to record and distribute the videos for academic

Identify the task for the user – 4 people ordering

Rehearse – refine evalution form

Phase 1 (evening)

1. Refine your evaluation form
2. Perform Evaluation with Family
3. Refine Digital Prototype
4. Refine evaluation form
5. Perform Evaluation with Callum
6. Refine Digital Prototype
7. Refine evaluation form

Phase 2 (now)

Together

1. coding manual + actual coding + user stories + acceptance criteria + MOSCOW + testing + deployment

* see Django documentation

v1 = bullet point v2 = essay v3 = refinement v4 = proofread

v5 = final/ ready to publish p = priority

//

It will raise criticism if you collect customer phone number when they scanned your QR code

Permission statement. Mobile ID

# Argumentation Patterns

General

* argument – logical reasoning (what, why, describe how)
* reference – background sources
* experiment – using results you found

Evaluation

* define a problem/ hypothesis
* define general question
* specific question
* narrow a problem to implementable
* implement it
* evidence of experiment

# Abstract

Motivate, Set Aims, Describe, Explain results, Contact email

# Project Proposal v4

Laser Mate is a £3.75-billion-yearly-profit software empire with the primary goal to enable restaurant customers to order and pay for food and drinks using a mobile phone.

The unique feature of the restaurant ordering platform is the lower waitering cost (50\%) and transaction fee (1\%). Restaurants that employ two waiters will attract a saving of half the waitering cost - £16,800 each year. Using this app, restaurant waiters will save around 50\% time and effort as they will no longer need to take, record, and deliver orders, and also give and take payment. Restaurant owners will also save additional workloads on dealing with staff rota, training and supervision and salary payment.

Another prospect of the mobile web comes from the transaction fees. Online web payment (0.39\%+2p per transaction) is more cost-effective than card machine payment (1.75\% per transaction).

The cost of business deployment and ongoing scaling is comparatively lower than the revenue potential. The operation of the company is mainly composed of the software, post advertisement and the menu update. Since post advertisement may only take a few days, assuming that each employee can register 15 restaurants a day, we will reach 5500 restaurants (£27 millions) in a year at a low cost. Another advantage of this business model is that we do not need to spend much time to maintain the operation of each restaurant owner after they have signed up to the system - we only need to update their menus and answer their enquiries.

The projected profit estimation is composed of the service charge and the difference in transaction fee. Upon taking 1\% service charge per customer transaction, we will obtain £3,000 for each £300,000 restaurant sit-in annual revenue. The cost, £3,000, is reasonable compared to the waitering cost of around £16,800. Furthermore, the fact that online web transaction is 1\% lower than card machine transaction means that our business will take another 1\% revenue. Accounting this 1\% service charge with the difference in transaction cost (1\%), we will have 2\% revenue for each restaurant (£6,000). Consequently, assuming that we will earn £5,000 for each restaurant (due to tax and other negligible costs), given that there are around 1.5 million restaurants in the EU and U.S, it is estimated that with 50\% market penetration, we will have a profit of £3.75 billion per year (750,000 restaurants x £5,000).

# Final Software Product – care after coding

* QR code link of software product demo
* diagrams for final product

# Business Executive Blueprint v3

## Phase 1 - Pre-Launch Groundwork

Once this software is sufficiently programmed and tested, we will crowdsource the system components that we cannot develop. The primary reason we don’t develop these software parts ourselves is because we don’t want to be legally liable in the case of failure. The software components we want to outsource include the software security, payment portal and legal and accounting [?] responsibilities. Please note that our graphic design (the motorcycle design) is copied from UpLab.com, however, we can legally use that as our final design because the developer states that we can use this design for commercial purposes[?]. We must also set up a company type [?] and business name (trademark). We may want to limit our liability so that company debts cannot affect our personal assets. We also need to set up the terms and conditions, tax protocols, insurance and recruitment policies, employment contract terms and legal protections. We may need legal consultations because our project involves a large-scale systematic monitoring and processing financial transaction activities. We may want to seek business funding and advice to scale the business internationally.

<https://www.accountingweb.com/practice/clients/where-do-accountants-fit-in-with-startups>

https://www.uplabs.com/posts/food-delivery-app-design-ui

<https://www.netlawman.co.uk/ia/types-uk-company>

https://www.termsfeed.com/blog/5-reasons-need-terms-conditions/

Furthermore, we may want to establish a company support manual to ensure that all employees follow the same set of organisational protocols for every action taken.

## Phase 2 – Software Product Marketing & Commercialisation

Once all the pre-launch groundwork is setup, we will start to promote and establish the client base. First, we want to write a marketing post template (see Appendix -> Software Post Marketing) so that we can mass advertise our service in a few days. The reason we use post is that it guarantees that the restaurant owner will at least read it. The cost of a post (colour-printing, papers, envelopes, stamps) is around £1 per restaurant owner.

The restaurant owner will then receive the template mail we sent them (see Appendix -> Software Post Marketing). If they want to become our member, they will be guided to go onto our home page and complete our registration process in less than 1 minute (see Restaurant Owner Interface -> Registration Page). We will then receive a member request [?] via our business email. Based on the information they have provided in the email, we will set up a restaurant owner account for them (upload the basic info, financial data, menu). Afterward, we will send them a template email (see Appendix -> Member Registration Confirmation Email) to notify them that the account is set up successfully and will send them the QR code menus by post. We will send them three times the number of QR code menus they need so that they have enough replacement when they lost it. Now, the restaurant owner can start using our service within 3-7 days.

## Phase 3 – Business Scaling

This marketing and client acquisition process will be continued until we have sufficient number of restaurant owner sign-ups. After that, we will expand the geography scope of the business, first in Scotland and the rest of the UK, then either the U.S. or other EU countries whose primary language is English. We want to have further legal and accounting consultations to ensures legal compliance. We may want to position ourselves strategically, by conducting marketing research on whether there is a need for this app and their costs of living (to see if we need to adjust our service charge).

Another issue is human resource - with the rising member sign-ups within the app, we want to ensure that we have sufficient staff level to cope with the consultation demand (menu update, enquiries).

From then onwards, the business is about repeating the processes for phase 2 and 3.

# Design Principles v3

<https://razorware.wordpress.com/2012/01/04/task-3-the-fundamental-principles-of-hci/>

GU HCI course notes

Checklist of design principle to ensure that every page adhere to that

Online Heuristic evaluation

## Captivating Design Principles

Web design is an essential software component because users will only want to reuse an app if it is comfortable and engaging to use. Our software design considers the importance of images over texts and the overall information representations (shape, spacing, colour, font). In our customer meal ordering page, we use circles rather than rectangles for the image frame because with rectangles, it feels that all the information is crammed together with insufficient spacing. We display four different types of food per screen and use the minimal textual descriptions for information understanding. We take Fitts’ Law [?] into considerations (spacing between software components and their sizes) so that the users will not feel overwhelming with the amount of text and image on the screen.

We adapted the colour palette [?] and ensure that all the colours are complementary with each other pleasantly. In our CEO interface page, all the different colours in the database rectangle headings are compliant with the principles in the colour palette. We use both professional and warmth colour to contrast the visual effects and enhance information memory. For example, for the company staff interface, we use competence colour at the top and the back of the rest of the web pages and warmth colours for the database rectangular headings. Another design criterion we followed is the consistency and the predictability of the information hierarchy structure. Information with higher importance are placed at a more outer areas of the mobile web and they will have a larger font size, to guide users understanding the app.

## Device Options

Our project idea considers the use of different types of devices (phone, laptop, phone) for the type of software users. The customer interface will use the mobile phone because almost everyone has a phone in their pocket. We will use the tablet layout for the chef & waiter and the restaurant owner interfaces because we want to have a larger screen size. We want to use a tablet stand for each tablet so that the chefs and waiters don’t need to lean down to see the orders and to prevent the contact between the tablet and the table. The restaurant owner interface is also designed in the tablet form so that they can use the tablets for the chefs and waiters and they don’t need to purchase an additional laptop (due to cost issue).

The CEO and the company employee platform will be in the laptop interface. Laptop is portable and it is easier to work with a laptop that has a keyword.

## Dynamic Information Management

Our databases adapt a click-to-change approach. Traditionally, to alter information in the database, you must click an add button and fill in a form to change it. To enable minimal effort and completion time in this data-driven system, we customise our database so that users can change the database contents by just clicking the data.

## Ethical Design

We rigorously follow the ethical principles set out by the University to protect the wellbeing and the rights of our evaluation participants and app users. You can see that in our ethical consent form for our second phase semi-structured interview evaluation.

We also obtain informed consent through our terms and conditions and legal policies to set out mutual agreement through our liability limiting statements.

Our company also abide to strict regulatory requirements to ensure the duty of care to our employees. These include health and safety practice to ensure that our staff are protected under long term screen exposure and the long number of working hours.

## Multimodal Interaction

The chef and waiter interface will have a sound notification every time a customer successfully pay for their meals. Restaurant staff cannot possibly look at the meals all the time. This system ensures that only when there is a new order will the staff look at the interface.

## Inclusive Design

Our mobile app takes into account users with specific difficulties, such as colour blind, dyslexia, eyesight problems and mental and physical disability. Over 0.038% of the world population [?] suffers from colour blind. To overcome this barrier, we ensure that we follow a checklist of colour-blind design criteria online [?], all our customer interface designs are high contrast, particularly for essential information that must be standout, such as prices and menu descriptions.

<https://www.colourblindawareness.org/colour-blindness/>

<https://www.designmantic.com/community/website-design-guide-color-blind.php>

We also incorporate simple English word choices for all the interfaces so that all people, regardless if they are linguistically or mentally disadvantage, can use the interface effectively.

All our interface components consistently accommodate for eyesight issue. All the texts are at least 16px [?]. Secondary texts are about 2 sizes smaller than the primary ones.

<https://learnui.design/blog/mobile-desktop-website-font-size-guidelines.html>

## System Feedback Mechanism

Our app provides system feedback when new data is inserted into the database, i.e. for the customer interface, we have a system feedback when a menu item is added to the system; when payment is performed (success/ failed); when bank details is added to the system.

## Design for Automation & Infinity

Our search and result functionality in our company employee and CEO databases incorporate the principles of the design infinity. Our quick search feature will only display data entries that are exactly the same as the input. While the app users type in their search keywords, the system will provide the possible word options after each character is entered. This is not only to help users find possible results through autosuggestions, this also reduces input error rates. Our search function also allows for multi-selection so that company employees and CEO can compare and further analyse company problems through different database entries. Our database supports infinity scrolling, not pagination, so that, as an example, database users are not limited to see a set number of data entries.

## Design for Hacking

We also design the system in a way that assumes hackers already knew our passwords. We will perform a cross-verification process in which every time a company employee or the CEO logs into their system, they will receive a text alert and only if the company employee types the randomised codes into the account system can the account be logged in. In the case that a hacker logged into our system, they cannot do so unless they also steal the employee phone. Our staff will then have the time to immediately inform the CEO the incident and we can shut down the account immediately.

Our database also assumes that we will have malicious employees who will manipulate sensitive data. After each company employee log into their account, all the confidentiality information, namely, their account password, and their financial data (card number, expire date, security number), will be represented as asterisks. This way, our employees cannot transfer money to their own account through our client bank in a large scale through our system.

## Worst-Case Scenario Documentations

One of our company protocols is to refine our worst-case scenario documentations (in a design for all eventuality manner) in which all the possible worse-case scenarios that can happen to the company will be denoted. This is to ensure that we can foresee all company problems and deal with repetitive incidence as soon as possible before it catches the public eye.

## Collaborative Development

According to the ACM code of ethics, software developers must only work in areas of competence. Consequently, company operations regarding software security, law, and accounting will be crowdsourced to the respective professionals.

# Software Requirements Specification p-7

## User Stories v2 (do it when you start coding)

<https://manifesto.co.uk/how-much-detail-should-a-user-story-have/#:~:text=Conclusion,part%20of%20the%20acceptance%20criteria>.

<https://rubygarage.org/blog/clear-acceptance-criteria-and-why-its-important>

The goal of the user stories [?] is to define the software parts to code and its purpose so that the team can understand the software quickly. The specific format of a user story is - as a [user type], I want to [perform something], so that [benefit].

### As a Restaurant Customer

Must Have: I want to view, select and tailor the meals available in the restaurant to the order summary portal so that I can order the food and drinks that I want. I want to view and adjust all the food and drinks I ordered before payment so that I can confirm if these are the meals I want to go for. I want to tip for the food and drinks that I ordered so that I can reward the restaurant for their impressive services. I want to inform the restaurant owner if they want the order to come together or separately so that the chef can prioritise the meals to cook.

==

I want to pay for my meals through the system so that the restaurant can confirm that they have received the payment.

I want my bank card details to be recorded so that I don’t need to type the same information again. I want to have a printed recipe so that I can claim the money back to my employers as an expense. I want to view the menu online, on the street or outside the restaurant via a QR code so that I can see or order the meals before I go to the place.

I want to reserve a table via the QR code so that I will definitely get a seat when I arrive.

I want to have a centralised platform to see and compare each restaurant based on their food (country, diet type (vegan, vegetarian, gluten free), price, discount, customer service, car park space, atmosphere and location. I want to be advertised with restaurant offers through text or email so that I can think about going to restaurants that I may not normally go for.

Acceptance Criteria

(1)

### As a Restaurant Chef

I want to see digital orders so that it is faster to see what orders need to be cooked; it is less likely for me to cook the wrong food or with the wrong quantity. I want to notify the waiter that the food is ready when they are not nearly so that the waiter knows that they need to serve a meal when they don’t notice it. I want to see all the meals that was ordered so that when the customer says that their meal is wrong, I can see the order history and the meals to cook for them. I want to have a seamless staff rota system so that they know when their shifts are as soon as possible.

### As a Restaurant Owner

I want to develop my restaurant website using the Laser Mate platform so that I can tailor the website and change the contents (menu, contact info, opening hours) quickly. I want to advertise the restaurant through the Laser Mate advertisement platform so that I can save additional costs when I switch the platform (from OpenTable). I want to have a staff rota system so that I can organise and distribute the roles to my staff digitally. I want to have a salary portal so that I can automatically pay my staff their salaries and, oversee, record and adjust the data. I want to have a customer analytics page so that I can see the popular food and drink and those that should be removed from the menu. I want to get advice and inspirations on food and drink recipes and cookery techniques so that my chefs can improve their cooking skills. I want to take online certified business growth and administration courses via Laser Mate’s YouTube channel so that I know how to better manage the restaurant business logistically and systematically. I want my staff (chef, waiter) to see Laser Mate’s induction tutorials so that they understand the operation of the app. I want to use Laser Mate’s low-cost accounting and legal services so that I don’t need to find my own consultant and I can get better advice on restaurant administration. I want to use Laser Mate’s insurance service so that I don’t need to find my own insurance company – it is also easier to provide documentation and get a lower price for the same service. I want to get recommendation and training for restaurant start-up and on dealing with emergency situations, such as breakage in interior design, furniture, repairing kitchen appliances and plumbing. I want to be able to contact the Laser Mate team so that they can change the menu details for me or answer some questions that are not posted online. I want to have a semi-automated emailing function for organising and scheduling food delivery so that I can email the food suppliers for food delivery (the item they need the stock, the time before the food must be delivered). I want to have a printer friendly version of the menu so that so that I can print out paper menus to serve customers who can’t use the Laser Mate platform. I want to have additional menu QR codes so that the customers can still use Laser Mate even when some QR codes are lost. I want to have different menus and costs at different times so that the customers can order different meals based on the time (morning, afternoon, evening). I want to use a mobile ordering system so that the restaurant can abide to the covid-19 rules by enforcing social distancing rules between the customers and the waiters. I want to have a customer complaint page so that I can improve my products and services based on their feedback.

### As a Laser Mate Employee

I want to have a login system to ask me for my email address and my password to get into the company employee platform so that unauthorised people cannot access to my employee account. I want to have a second login system to ask for my phone number and my second password so that the system checks whether I am happy with the email address and the phone number I currently have. It also prevents the situation where other company can log into the staff account if the employee chooses the same password for other companies. I want to then have a phone text message to ask me to input the text code in my phone to the login system so that I can be alerted if another person logs into my account. If I found that an unauthorised person is logging into my account, I can immediately obtain the text message and notify the incident to the company CEO. I want to get access to my account only if the 3-step login is successful at first attempt to prevent others from trying the login details. I want to add new client (restaurant owners) details to the restaurant database so that they can use the ordering system. I want to record the restaurant number so that I will not obtain the wrong information when referring to another restaurant with identical name. I want to record the restaurant name and address so that the customer interface can show this information to the customer to verify that the menu they are looking at is referring to the restaurant they are in. I want to record the name for the restaurant owner so that I can refer to the client by name when I get their phone call and can verify their identity. I want to record the restaurant email so that I can contact them (for advertisement, responding to their emails) by email if they cannot respond by telephone. I want to record the owner phone number so that I can contact them urgently when their system is down. I want to record all the restaurant QR code menus so that I can resend the QR code menus to the restaurant if they have lost it for a particular table number. I want to record the restaurant weblink so that I can generate the QR codes based on these weblinks. I want to record the restaurant login username and password so that I can access and change their restaurant menu (the time it serves, the meal (photo, price, short and longer descriptions, dish category, extras, allergy information and whether they are available for today); restaurant information (restaurant name, restaurant address, restaurant phone number, restaurant owner name, restaurant account password), restaurant staff account details (staff account username, staff account password); financial information (card type, card number, expire date, security number, card holder name). We will now move onto the weekly progress report page. I want to oversee the number of new clients I have added and removed each week so that I can compare and track my progress on client number; I want to understand why clients leaves the company so that we can improve upon the existing service. I want to describe the total projected profit each week so that I can aggregate the profit summary and form a long-term progress report. I want to document new problems and solutions not in the staff manual so that we can develop a central documentation for all the staff and CEO to follow. I want to submit a report to the CEO weekly regarding new innovative suggestions so that the company can improve its products and services. I want to refer to company staff manual every time I do something so that I can strictly follow company executive procedures without missing some important steps and can improve upon the existing documentation.

### As a Laser Mate CEO

We will now refer to the employee database. I want to add a new employee tuple so that I can record company data. I want to allow myself to edit all the information in the database by clicking the data so that I can make changes easily. I want to construct the database so that each company employee can only edit and see the data they registered within the database. This is to prevent malicious employee to reveal all the client data to third parties – they can only reveal the details for the clients they have added. I want to record the names for all the company employees so that I can remember what they call, and I can find each employee data easily. I want to record and oversee the employee login username and password so that if they leave the company without company notice, I can log into their account and change the employee who can edit the restaurant details. I want to be able to change the password for each company employee so that if they leave the company, they cannot log into the system again. I want to ask for a second login details for my CEO account once my first set of username and password are correct so that the hacker will need to enter a second combination of login details before they can access and change the data in the system. I want to have a second login system that again asks for my email address and second password (when actually the system demands for my telephone number under the false email address title) so that I can confuse the hacker to enter the wrong information. In the login system, I want to state that the second password should be 8 characters long, have at least one capital letter and one number when the actual password violates some of these criteria so that I can further confuse the hacker to enter the wrong password. I want to setup the second login system so that it will give me a phone alert that asks for my approval before the hacker can log into the system – therefore, they can only access the database when they get the textual approval from my phone. I want to setup a text messaging system that every time I scroll through another data entry or every time, I edit an entry, I will get a text message alert so that when hackers are viewing or editing my data without my acknowledgement, I will get a security alert straight away. I want to setup the system so that the CEO and the company employee accounts are frozen when I send a specified text code in my phone or via my email so that if either the CEO or the company employee accounts are hacked, the hacker is forced to logout and cannot view or edit the information. I want to have this multiple security measure so that when the hacker changes some parts of my security code, other parts of the security code will hopefully be active. My system will be compromised only when the hacker knows my email address, first password, phone number, second password, steal my phone, know my phone security code, reply to the text message, make sure that I don’t know that the security is compromised every time they access to the new database tuples and don’t allow me to freeze the CEO and employee accounts via phone or email. I want to get an email notification if an employee enters the wrong login details twice so that I can contact them to verify if they entered the incorrect login information twice. I want to store employee NIN number so that I can perform employee tax duty. I want to record employee telephone number so that I can contact them individually if I need to. I want to record the employee role so that I know their job descriptions. I want to record the job descriptions for all the employee roles so that I can organise and allocate tasks effectively. I want to record the employee hourly salary so that I can adjust their salary accordingly and help calculate their weekly working hours. I want to record the employee weekly working hours so that I can calculate their weekly salary and transfer them the money weekly. We will now move onto the restaurant weekly transaction page. I want to record the restaurant number so that I can cross-reference other restaurant data using the restaurant number. I want to record the weekly transaction date so that I can first search a specific restaurant number and sort their weekly transaction date so that I can see the weekly fee taken sequentially by date. I want to record the transaction date so that I can cross-check the full transaction details on the bank app with the transaction date. I want to record the weekly transaction status (success, failed, pending) so that I can quickly repeat the payment transaction and resolve any payment problems. I want to have the aggregated database from all the company employee accounts so that I can take on their job roles if they are sick; I can track the entire company activity for managing personnel. I want to have a manager section for each restaurant tuple so that only the allocated staff and the CEO can see the restaurant data. If the manager entry is changed, the previous allocated staff can no longer see this data entry and the only new manager, and the CEO can see it. I want to have a performance database so that I can track, for each company employee, the number of clients they signed up, the weekly profit that they help the company make, the number of clients who unsub-scripted to our service. I want to set up a telephone verification – the person who accesses the account must enter the randomised code that is sent to my phone to the database platform before they can access to the CEO account - so that I can know when someone log into my account without my acknowledgement. I want to create a new employee tuple so that when a new employee joins the company, I can add their information into the database. I want to allow for data sorting for all data tuples by clicking the database column so that I can easily find the same database entries.

## Acceptance Criteria v1

Acceptance criteria are a set of conditions that the software must satisfy for the user stories to be accepted by the product owner.

### As a Restaurant Customer

Scenario: view, select and tailor the meals available in the restaurant to the order summary portal.

Given a QR code paper on the table. When I scan the QR code with my mobile phone. And I open the meal ordering platform. Then the system shows me the restaurant address that corresponds to the restaurant number, all the user-defined meal categories, the details of the available meal options (brief description, price, add button), at that particular serving time for the meal category for that restaurant number and the “Go to Payment” button. The system will show the horizontal forward/ backward arrow indicator at the top of the screen that shows the possible meal categories for the particular restaurant number if the system can load the next/ previous category entry based on that particular restaurant number and serving time. The system will show the horizontal forward/ backward swiping indicator in the middle of the screen if the system can load the next/ previous meal entry based on that particular restaurant number, serving time and the meal category.

When I click the horizontal forward/ backward arrow indicator at the category section. Then the system will show me the next/ previous meal category at that particular serving time for that restaurant number. And the system will show the horizontal forward/ backward arrow indicator at the top of the screen that shows the possible meal categories for the particular restaurant number if the system can load the next/ previous category entry based on that particular restaurant number and serving time. When I click the horizontal forward/ backward swiping indicator in the middle of the page. Then the system will show me the next/ previous four boxes of meal descriptions for that meal category at that particular serving time for that restaurant number. And the system will show the horizontal forward/ backward swiping indicator at the middle of the screen that shows the next/ previous sets of possible meals for the particular restaurant number if the system can load the next/ previous meals entry based on that particular restaurant number, serving time and category.

When I click one of the four boxes that corresponds to a meal. And I go to the extended view of that particular meal. Then the system shows me the additional details of that meal (photo, brief description, longer description) based on the restaurant number, meal serving time and meal category, as well as the way the customer can tailor it, special requests, the number of orders, the “Add to Bill” button and the “Cancel” button. When I click the radio button (given that there is a meal tailored section), it will turn green/ white if the previous state was white/green. When I click the plus/ minus button for the meal quantity section, the number will increase/ decrease by one.

When I click the “Add to Bill” button. Then the system adds the customer phone number, restaurant number, restaurant table number, meal ordered, meal price and the special requests to the system customer temporary order database. And the system shows me the “Item Added!” message. And after 0.5 second, the system redirects me to the main ordering platform. When I click the “Cancel” button. Then the system redirects me to the main ordering platform.

Scenario: view and adjust all the food and drinks I ordered before payment.

Given that I clicked the “Go to Payment” button. When I open the order summary platform. Then the system shows me the list of food and drinks I have added to the bill, the number of these orders, the prices, the total price, add tips, a click box (whether to serve the meals separately), the special requests comment, “Back to Menu” button and “Go to Payment” button. When I click the vertical up or down “More Items” button. Then the system shows me the previous few or next few food and drinks items I have ordered. When I click the “Back to Menu” button. Then the system will show me the restaurant meal ordering platform with four boxes of meals under a particular category.

Scenario: tip for the food and drinks that I ordered.

Given that I clicked the “Go to Payment” button. When I open the order summary platform. Then the system shows me the add tip function. When I click the plus button next to the add tip function. Then the system shows me another interface that asks me to add the tip amount. When I enter a non-numerical value and press enter. Then the system will show me another interface that writes “Invalid Tip Value!” and a “Back” button. When I click the back button. Then the system takes me back to the add tip amount interface. When I enter a numerical value. Then the system shows me the order summary page with the updated tip value next to the tip label.

Scenario: inform the restaurant owner if they want the order to come together or separately.

Given that I clicked the “Go to Payment” button. When I open the order summary platform. Then the system shows me the “Meals Come Together?” label with a tick box. When I click the tick box (when the tick box is empty). Then the system inserts this data to the temporary order database. “meals come the system shows a tick inside the tick box. And the system will When I click the tick when there is already a tick inside the tick box. Then the system shows no tick inside the tick box.

### As a Restaurant Chef

### As a Restaurant Owner

### As a Laser Mate Employee

### As a Laser Mate CEO

## Business Requirements

## MOSCOW p-7

Security is should have because we need to hire a security team to do that

Payment is should have because we need to ask for the payment company for it

# Software Development Process – ongoing

https://en.wikipedia.org/wiki/Software\_quality\_assurance

<https://standards.ieee.org/standard/24748-3-2020.html>

Summarise all methods

Business Models

https://www.investopedia.com/terms/b/businessmodel.asp

Value Proposition Canvas

<https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/>

User Stories

<https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/>

MOSCOW method

We want to showcase the recommended approach to develop the software as a whole. The point of this section is to document the algorithmic thinking that everyone can apply to develop the perfect software at first attempt with the shortest time possible. A world-class blueprint that produces the mastermind of designers, strategists, readers and 10x programmers – that those without this can never rival.

## Phase 1 – Product Conceptualisation, Prototyping, Evaluations and Testing

We want to showcase the approach I use develop this software. The point of this section is to document the algorithmic thinking so that other software developers can build upon my existing programming strategies. First, you should draft an overall development plan (go to Ctrl + F -> Heading) with prioritisation from start to end. You may want to refer to the internet resource and find the industrial approach to develop each section. The first stage of software development is software requirement specification, in which I try to think, for each stakeholder (customers, chefs, owners, company employees, CEO), what they want to achieve with the software.

Then, for each step, list the key tasks you need to work on and give any supplementary details. The next step is to go through each key task and add relevant research contents. With all the possible information you have written for each section, draft the dissertation into an essay from start to finish. You may want to prioritise which section to work on first and you may want to cross check and refine other sections as you work through the project.

You should now be ready for coding.

First, we write out the full systematic requirement analytics processes (Figure?) how the full restaurant business operation occurs, for the different stakeholders. After having a list of goals, we think about and write how software can better improve the experience. Once a table that denotes the goals and the software features for different stakeholders is written, we develop the paper prototype (see appendix) using the software components set out in the requirement statement. While writing the paper prototype, we exclude unimportant features so that we only write the minimal design. We now forget and set aside all these design inspirations and draft out another requirement statement, without referring to another notes. We want to use a different method, namely, the Value Proposition Canvas model (VPC) [?] (Figure X). This is to develop a second, better plan based on an established understanding. The VPC brainstorms the software ideas by identifying the customer profile (their gains, pains, jobs), and subsequently the value proposition (gain creators, pain relievers, product and services).

Restaurant Customer

|  |  |
| --- | --- |
| Goals (things that they will do step-by-step) | Software Features (how software can achieve it) |
|  |  |

Figure X

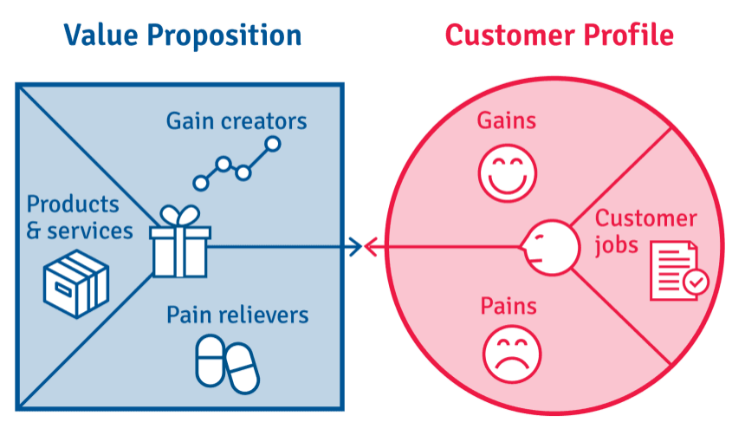
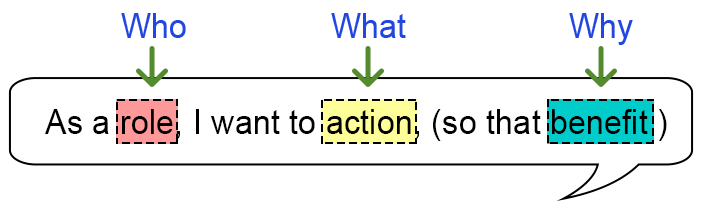


Figure X

Based on the two requirement statements and the paper prototype that are derived under different models and understandings,

we then create the minimal digital prototype (see appendix) using Adobe XD. The process is to gather and graphically place all software components logically

rewrite the requirements using the user stories [?] (Figure). The point to repeat the same process with different methods is to collect as many requirements as possible for future work. The user stories can now act as a communicator through which any other software developers can checklist the software components within the scope of the project, with the underlying reasons of importance. Based on the user stories, we develop



Write the bullet points then the passage

===

After that, using the (4) MOSCOW method [?], we categorise the software components based on the level of priority each feature should be deployed. Lower priority will be our future work. The objective of this prioritisation approach is to ensure that we first develop a minimum working product that our clients can use. Now, having a list of high-priority features, we then write out the corresponding (4) test cases with acceptance criteria under the test-driven development approach so that we develop a contract which states all the test cases for which their acceptance tests must pass for a minimum viable product.

After having all the software specifications set out in a contract, we then write out the overall (5) paper designs on paper for all the software components that are determined to be the highest priority. We will then draft out some (6) survey questions using Google Form, which will then be used to conduct the (7) semi-structured interviews. These interviews will be performed with my project supervisor and my family. We will also take some design guidance from an (8) online heuristic evaluation [?]. Once all the requirements and suggestions are collected, we develop a (8) digital wireframe using Adobe XD. We will perform additional evaluations, by first designing another version of (9) Google Form survey, such as (10) several semi-structure interviews, again with my project supervisor, my friends and family. Finally, we will (9) rewrite the project requirement specification and finalise the (10) digital wireframe and (11) the acceptance test criteria.

## Phase 2 – Software Coding, Programming Documentations and Testing

1. Platform Considerations

* version control (GitLab)
* web front-end (bootstrap, Django)
* database (PostgreSQL – scalability)
* test suites
* Security
* software deployment

## Phase 3 – Software Deployment, Evaluation and Testing

# Software Coding Manual p-7

Web link

## Phase 1 – Pre-Programming Strategy

Justifications – above is the tricks – why use that platform

### Platform Considerations

Tried react native. The reason we use react native is so that I can get experience in coding a Android, iPhone and web at the same time. However, it turns out that there are more additional things to learn. This will drag the development speed for the project. So react was used at the end.

https://www.linkedin.com/learning/choosing-a-cross-platform-development-tool/cross-platform-vs-native-development?u=26205482

* version control (GitLab)
* web front-end (bootstrap, Django)
* database (PostgreSQL – scalability)
* test suites
* Security
* software deployment

Database Requirements (PostgreSQL)

* High Traffics (50 million users per day)
* High volume of photo and text update and retrieval, bank transaction
* High speed

Framework Documentation and Support

* LinkedIn Learning (Intensive documentation and consider Ease of Development)
* Used by Similar Multi-Billion Apps such as Instagram

Ease of Development

- Bootstrap Studio – design webpages without coding then copy and paste the auto-generated code to Django

Security

* Django in-built security

Cost & Reliability of Coding Platform

* Development and deployment
* AWS

Cross-Platform – responsive designs

* Between mobile, tablet and desktop web
* Tablet (chef/waiters & business owner & admin interface)
* Mobile (customer)

Testing Suite

* Spring Boot

Evaluation

- Google Form

### Overall Coding Workflow

See Django documentation

### Coding Tutorials & Documentation Searching

<https://www.valentinog.com/blog/drf/>

**Responsive Design (2015)**

– ensure that the web app is displayed accordingly in the mobile phone, tablet and desktop [https://www.linkedin.com/learning/creating-a-responsive-web-design/introduction-to-this-course?u=26205482#](https://www.linkedin.com/learning/creating-a-responsive-web-design/introduction-to-this-course?u=26205482)

1. Relevant Sections for the material – responsive web design

* header, main section, atmosphere section, content, navigation, footer content
* font, text styles, heading and page container, logo, button, table, graphics
* nav bar, list items, links, device computability options for nav
* adjustable layout for large and medium screen
* moving navigation for smaller screens, rearrange logo, main section, atmosphere, main text, spacing, footer
* making adjustment for the smallest screens

## Phase 2 – Coding Executions and Documentations

### p8 (actual coding and testing)

### Programming Principles

### Coding Templates

Reduce image size

# Evaluation Techniques v3 p-6

<https://software.ac.uk/sites/default/files/SSI-SoftwareEvaluationCriteria.pdf>

<https://software.ac.uk/sites/default/files/SSI-SoftwareEvaluationTutorial.pdf>

more web link

## Phase 1 – Paper Prototype Evaluations

### Literature Review

Once the paper prototype is drafted, we performed a literature review. It is a process whereby we find and compare related software products to see potential design flaws and improvements. Since it is a fairly new business idea, our literature review indicates that there are limited existing designs for evaluation.

The software designs we find are the Dines and Starbucks (see Appendix -> Literature Review). We have transferred the good features in their software designs and analyses their software weaknesses. Overall, I think that their order interface is unpleasant to use because it is jus a list of food and drinks. The user should have a better app experience to see the meal photos.

### Semi-Structured Interviews

The first evaluation with my supervisor was conducted over Zoom. We looked at each paper prototype and discussed how the designs can be improved based on a set of pre-scripted general questions. I planned to ask the supervisor what each webpage do (to see their understanding of the app) and how would you design the webpage differently (do spot any design weakness). However, we ended up not following any of these questions and we just talked about what can be added to the design because it was quite defective. Instead of writing minutes, I typed out the design suggestions into the PowerPoint (see GitLab -> Wki -> Evaluation) straight after the comment was made, so that I don’t need to spend 2x the effort to listen back to the whole video. All evaluation comments are reflected to the digital wireframe designs.

Although this interview evaluation didn’t follow the procedures as planned, I had a first-time experience of a design evaluation and build some understanding of the process.

I also conducted another design interview evaluation using a similar approach with my family.

## Phase 2 – Digital Wireframe Evaluations

### Subjective Evaluations

After having evaluated my paper prototype with my supervisor and my family, I developed a digital wireframe with considerations of their design suggestions. After that, I read through the previous sections of the dissertation again, namely, the business executive blueprint, design principles, and the user stories so that I can logically walk through each digital designs and ensure that they make sense and adhere to the software requirements. The reason I did this evaluation before inviting other participants is that I want to make the designs as perfect as possible before the next evaluations. Afterall, I would care about the design more than other people. It is easy to make mistakes at first attempt and it is important to not to evaluate your products when there are still many design defects.

### Online Heuristic Evaluations

<https://uruit.com/ux-quiz>

Once the digital wireframe is repeatedly checked against in terms of the logistics and designs, I then check the software design against the formal criteria, using online heuristic evaluation [?] (see Appendix -> Online Heuristic Evaluations).

### Ethical Form

<http://dcs.gla.ac.uk/ethics/assessment-form.pdf>

The first section of the interview process, after briefing the participants the overall evaluation format, is the ethical form. This goal of an ethical consent form is to protect the wellbeing and the right of the participants. The development of the ethical form in our evaluation considers all the ethical checklists set out by the School of Computing Science at the University of Glasgow [?]. It has been signed by myself, and cross-checked by the supervisor (see Appendix > Evaluations > Ethical Form).

### Semi-Structured Interviews (write after evaluation)

<https://www.quirkos.com/blog/post/semi-structured-interview-guide-qualitative-interviews>

how to draft questions (one at a time) principles

using the cognitive walkthrough [?]

<https://en.wikipedia.org/wiki/Cognitive_walkthrough>

think-aloud [?] techniques – talk the evaluator through each web pages in terms of what they are doing and what they are trying to do with the app.

<https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>

get supervisor to check evaluation sheets, integrate the ethical form into my evaluation.

The second attempt at evaluation is more constructive, mainly because I went through the dissertation again to make sure that each software parts are logical and adhered to the business executive blueprint, design principles, software requirement specifications stated above. I have conducted an interview with my family and friends first to make sure that the designs makes sense to them. Refine my evaluation goals and questioning approach. Check some internet resources. Draft the evaluation procedures with google form and rehearse in my mind the full interview process.

<https://docs.google.com/forms/d/e/1FAIpQLSfj5qns9JwFUL-IT7pLzIqUyGwFzyLAmGMIO4gDp4eEEElOfg/viewform?usp=sf_link>

1. Supervisor
2. Family
3. Friends

## Phase 3 – Final Product Evaluations

### Software Testing

Evaluate software against user and system requirements

### Business Execution Testing

Develop User Supports & Employee Manual

Check whether the web designs fits for purpose

### Subjective Evaluations

Responsive design

All sections

### Semi-Structured Interviews

1. Supervisor
2. Family
3. Friends

# Software Testing p-7 (document each testing examples) p10 (run all tests)

<https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>

some one example how testing is done for each section

## Business Requirement Testing

– Functional Testing

* Business requirement met

## User Requirement Testing

* End-to-end testing
* Replicate user behaviours

## Error Testing

* Methods, functions and class testing
* Unit test

## Software Feature Testing

* Smoke testing
* Software features and functionalities work

## Interface and Service Integration Testing

* Integration
* Modules or services used by the application work well together

## Performance Testing

* System response time and loading speed under different data volumes

Optimise and evaluate your system performance, not lagging with mass data entry and retrieval – stress test

Response time, loading time

# Software Deployment p-7

Web link

What is the choice of platform?

How to do it – the weblink and your brief descriptions

# User Supports & Employee Manual

Document potential issues and solutions so that employees can follow

Reduce enquiry to CEO

## User Support Page

via YouTube Tutorial Channel p11

Headings and Contents

## Company Employee Execution Manuel

Device

Headings and contents

All possible problems

How to answer enquiries from customer and restaurant owners

How to do things properly

Document process and improvement – trouble-shooters

As part of staff training

Document why good and bad

# Conclusion

# Bibliography

Business Models

https://www.investopedia.com/terms/b/businessmodel.asp

Value Proposition Canvas

<https://www.b2binternational.com/research/methods/faq/what-is-the-value-proposition-canvas/>

User Stories

<https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/>

MOSCOW method

IEEE International Standard – System and Software Engineering Life-Cycle Management https://standards.ieee.org/standard/24748-3-2020.html

# Appendix

Business Executive Blueprint

* Software Post Marketing
* Member Registration Confirmation Email

Digital Prototype

Software Coding Templates Documentations

Software Testing Templates Documentations

Evaluations

* Literature Review
* Online Heuristic Evaluations
* Ethical Form
* Phase 2 Semi-Structured Interview Questionnaire Analytics
* Phase 3 Semi-Structured Interview Questionnaire Analytics

Systematic Requirements Analysis

Software Requirements Specification

* Value Proposition Canvas
* Paper Prototype

User Support Manual

Company Employee Execution Manel