# Project Proposal v1

1. Projected Annual Profit when business is at peak performance

* Laser Mate! is a £2.5 billion yearly profit software business whose primary goal is to enable restaurant customers to order and pay for their meals online using a mobile phone.

1. Product market fit

Why people want to use the product

* Compared to the traditional way of meal ordering, restaurant owners can benefit from lowering waiter cost at £16,800 - £33,600 each year. The restaurant owner will save a staggering around 50% time and effort as they will no longer need to take, record and deliver orders and to give and take payment.
* Restaurant owners will save additional workloads on dealing with staff rota, training, supervisor and salary payment.
* Time taken to order food and drinks using our platform is similar to that using traditional methods

1. Profit Estimation (when we will reach £1 million and the profit at peak performance)

Why this business is worth doing

Revenue = £7,000 per year per restaurant

* We can take 20% of the cost saved by restaurant owners (£3,500)
* The transaction fee in mobile web is 1% lower than that in bank card (0.39% + 2p vs 1.75% per transaction) (£3,500)

Cost = £1,500 per year per restaurant + £2,000 startup cost + £20,000 x 2 base salary

* Senior software developers (site reliability, data security, speed) – seeking partnerships through share option.
* Business start-up cost (business name, legal policies)
* Business maintenance cost (insurance, tax, software deployment fee, Q.R. code generators)
* Post marketing (£1 per post)
* £20,000 Base salary

When we reach £1 million profit disregarding start-up cost and base salary = £5,500 per restaurant per year

* If one person can sign up 1 restaurant per hour
* Assuming no restaurant owner leaves the service, and we have one staff repeatedly sign up restaurant accounts for 8 hours a day
* Number of days needed to reach £1 million annual profit = £1m / (£5,500 x 8 restaurant signups per day) = 23 days.
* Account for profit making from the time at which the restaurant owners adopt our service.
* We don’t need to have one staff once we reach over £1 million profit

Profit when we reach peak business performance

* 1.5 millions restaurants in E.U. and U.S.
* 30% market penetration = 450,000 restaurants
* 450,000 restaurants x £5,500 profit per restaurant = £2.5 billion

1. Unique feature of this business

* Restaurant owners will need minimal staff intervention unless they need to change some menu details or have specific enquiries.
* Powerful platform as high user traffics can help adjacent billion-dollar businesses (e.g., table reservation business, restaurant review site, take-away and delivery and social media)

# Project Overview

This section denotes the consideration points prior to software development. This analysis will help future company strategic refinement. We first define the high-level business requirements before considering the user’s perspectives. Subsequently, we prioritize the software features to implement and how their allocations should be.

## Final Software Product v1

* Pictures of the final product

## Business Analysis v1

### Understanding the Competitive Edge v1

Firstly, the business analysts must understand the unique advantages of the new system. Without having a compelling reason to adopt a new practice, people will not spend the effort and time to adjust their habits. This system suppresses traditional ordering methods by the costs needed to perform the same tasks by £16,800 - £33,600 per year. Restaurant owners will not need to take, record, and deliver the orders and give and take meal’s payment as customers will perform these tasks instead. Since restaurant waiters would only require taking the customers to the table and deliver the meals, they will save at least 50% of the work. By reducing the number of people needed to maintain the restaurant waitering operations by 50%, restaurant owners would save an annual staff cost of 50%. A typical restaurant will have two waitering staff. Therefore, we could help restaurant owners save an annual cost of £8 x 6 hours x 350 days = £16,800. Furthermore, restaurant owners will save additional work on managing staff rota, training, supervision, and accounting.

### Feasibility Study & Literature Review v1

The business should conduct a brief marketing research to assess the scope of similar software. The project should proceed only when the business ideas are not universal. It would be ideal to determine if this business can sustain a sufficient market share. A brief google search suggests that there are less than three companies in the U.K. that conduct businesses with a similar focus. Namely, they are Dines, Yo! Sushi, and Flipdish. Our digital wireframes consider the design options in these applications. Pics

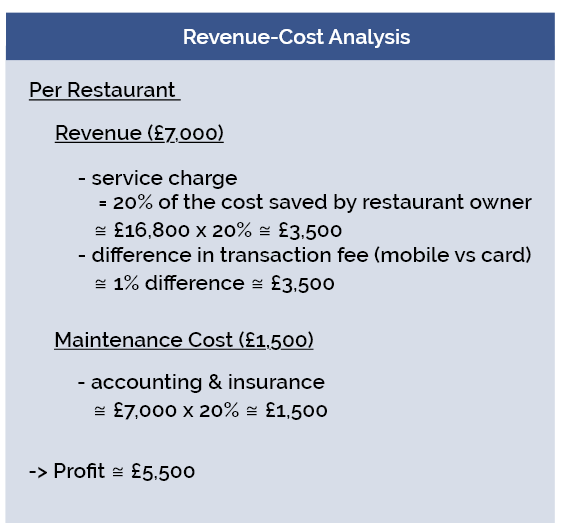
### Revenue-Cost-Profit Analysis v1

Another essential component of the business is revenue-cost analysis. Figure A suggests that each restaurant account signup would attract a profit of £5,500 per year. The method that computes the final value of £16,800 is mentioned in the “Understanding the Competitive Edge” section above. The 1% transaction fee comes from the fact that the transaction cost in mobile web is 1% lower than that in bank card (0.39% + 2p vs. 1.75% per transaction).

In terms of the annual profit analysis, given we have 1.5 million restaurants in the E.U. and U.S., we will hit £2.5 billion at 30% market penetration (30% x 1.5m x £5,500).

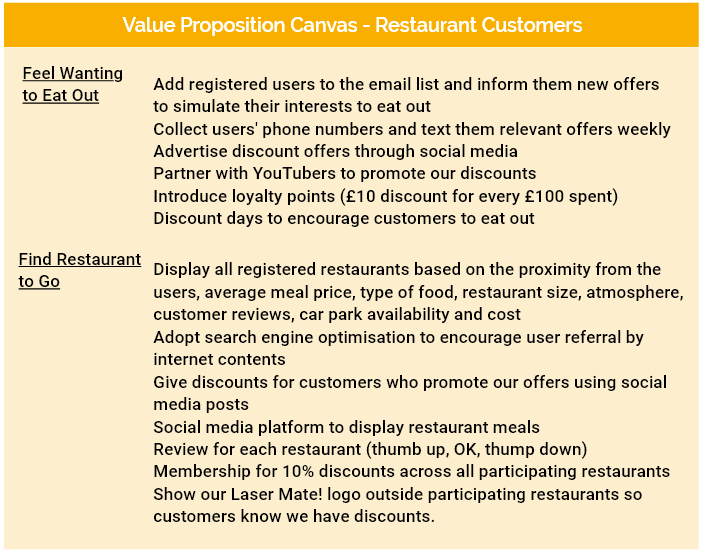
Our utmost system priority is to prevent data breaching and software breakdown (due to high user traffic). To maintain the platform’s security and reliability, we are looking to partner with an experienced software developer with a share option to take care of these software aspects.

Our calculation neglects the start-up cost. These include business registration, legal policies, terms and conditions, cookies, deployment cost, Q.R. code generators, post-marketing, and base salary.



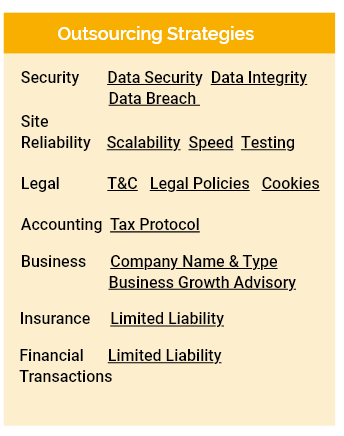
### Value Proposition Canvas

The following work is the analytic outcome of the value proposition canvas model (Figure A). This tool exhaustively defines all the possible tasks for each stakeholder group. We identified all the relevant scenarios each user group (customers, chefs, waiters, owners, company staff, company CEO) would encounter with possible software solutions. The outcome report will help us identify the business activities relevant to our business objectives.



### Business Executive Strategy v1

A reputable software should consider the critical business operations as any design alternations will lead to additional time lost due to redundant software development. The minimal business executive processes include software security and reliability; post-marketing; product review; account registration; platform setup; Q.R. codes postage; and maintenance.



Our software will follow adequate justificatory framework. We will recruit accredited security professionals to ensure that high user traffics will not compromise the platform’s speed and operations. Our outsourcing security team will protect the system from malicious attacks and data breaches.

Following system integrity, we will begin our customer acquisition process. Post-marketing is the best first-wave advertisement channel since stamp, envelop, paper, and printing costs are £1.5 per letter. Compared to all other marketing mediums, it is the method that almost guarantees all the restaurant owners would read the advertisement contents. When the restaurant owner receives the letter (Appendix Letter), they will see a Q.R. code that will redirect them to the company’s YouTube channel, demonstrating the software products. Video presentation should be the most potent persuasion mechanism to attract clients with the software capabilities and benefits.

Restaurant owners who express an interest will scan the Q.R. code and go to the website (picture). The landing page aims to demonstrate our product and the benefit of becoming a member, using minimalistic representation, such as short texts and pictures.

Perspective members could sign up by using our registration page or email. To complete this process, they would only need to fill in 11 pieces of information. We aim to do all the tasks on our client’s behalf to minimize their inputs and efforts. We will deploy a standard automatic email confirmation system to verify the customer’s email address.

The most important aspect of the business process is to set up the restaurant sit-in ordering system with all the business info and menu. Our staff will be fully responsible for this process as the platform must have the correct presentations before deployment. Hopefully, this will prevent clients from feeling frustrated learning the system.

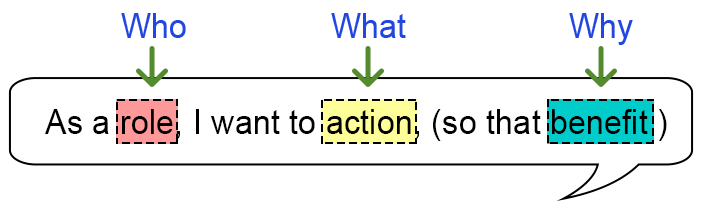
Following the production stage of the restaurant platform, we will print out the table Q.R. codes for later delivery. All the Q.R. codes will have lamination to protect damage due to long-term use. To ensure an appropriate number of replacements Q.R. codes, we will prepare the number of Q.R. codes three times the number of the tables.

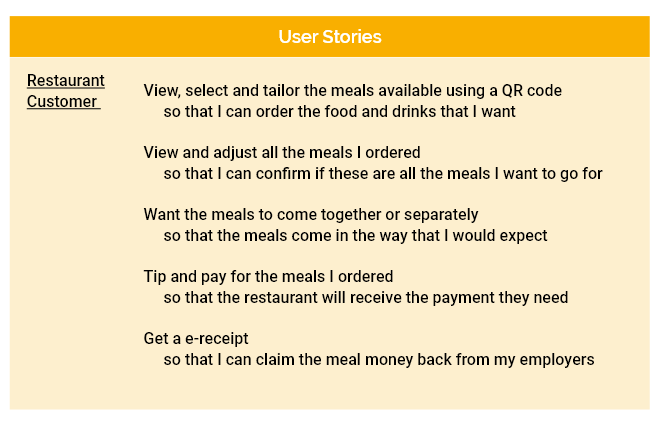
Members with successful signups should now be able to enjoy our software platform. Hopefully, they would need minimal support and intervention as we will upload help and video tutorials to guide them with the platform’s use.

## User Requirement Statements v1

### User Stories v1

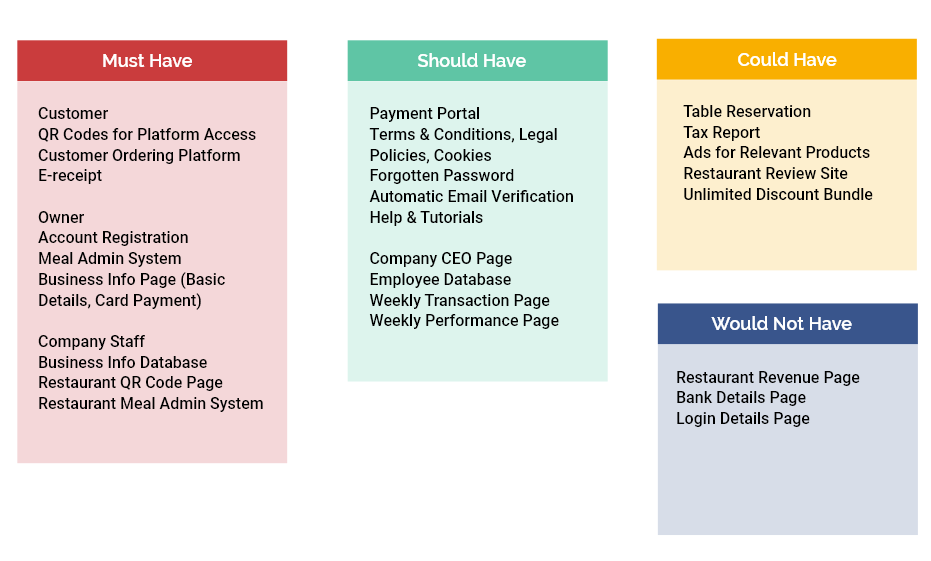
Our user stories will follow the standard pattern. The goal of a user story is to generically define the user requirements of the software. By having a breakdown of the software tasks the users could do, report readers will have an understanding of the possible software features. Role describes the system’s user groups that share a set of predicted tasks. Action(s) are all the generic tasks they will perform when using the software. Benefit justifies why they need to perform theses actions.





### MOSCOW v1

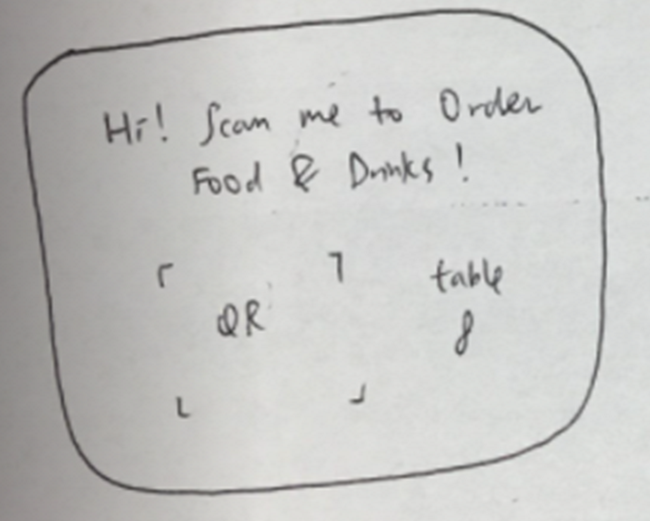
The MOSCOW method prioritizes the software features to implement. We will first develop all the “Must Have” features before the “Should Have” features. Subsequently, we will implement the “Could Have” features. Since the “Would Not Have” features are the ones not to include in the software development planning, they will be in the record, but get ignored.



## Software Design

### Paper Prototyping

Describe what and why paper prototype



Usability Heuristic Evaluation

Semi-Structured Interview & Questionnaire

* Supervisor
* Family member
* Literature Review

### Digital Prototyping

* Paste designs here for definite considerations

Cognitive Walkthrough & Semi-Structured Interview & Questionnaire

* Supervisor
* Family member
* Friends

### Design Principles

<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.

Internationalisation

All the texts in the software, including the button texts and the system messages, can be changed using the database system so that the app can be represented in any languages.

## Software Development Methodologies

### Front-End Design using Adobe XD & Anima

Describe what is Adobe Xd and its unqieu advantage compared to other platform

Convert paper prototype to digital wireframe based on the aforementioned design principles. This is to create a consistent design throughout the application.

Anima is a platform to allow for direct code conversion from Adobe XD to React.JS code. Convert exact design to web platform. Therefore, we should carefully name and merge each component and interactions to ensure meaningful conversion for future code review. It’s a far advanced tool compared to Bootstrap and Material Design.

Watch YouTube videos for inspirations and tutorials of the things that you could do

Naming components – when converting design to code, the SCSS for all the components will have the correct naming conventions. Easy code referral for future adjustment.

The main notes to discuss on Adobe XD are to use the “make component” function to group related items, for examples, you should use this function for all the elements within category. This will enable you to replicate the entire component across all meal interfaces.

Another useful feature of Adobe XD is denoting the interaction between different pages. This will enable you to click through different buttons and swipe through options.

Allow for customerisation for responsive design using XD.

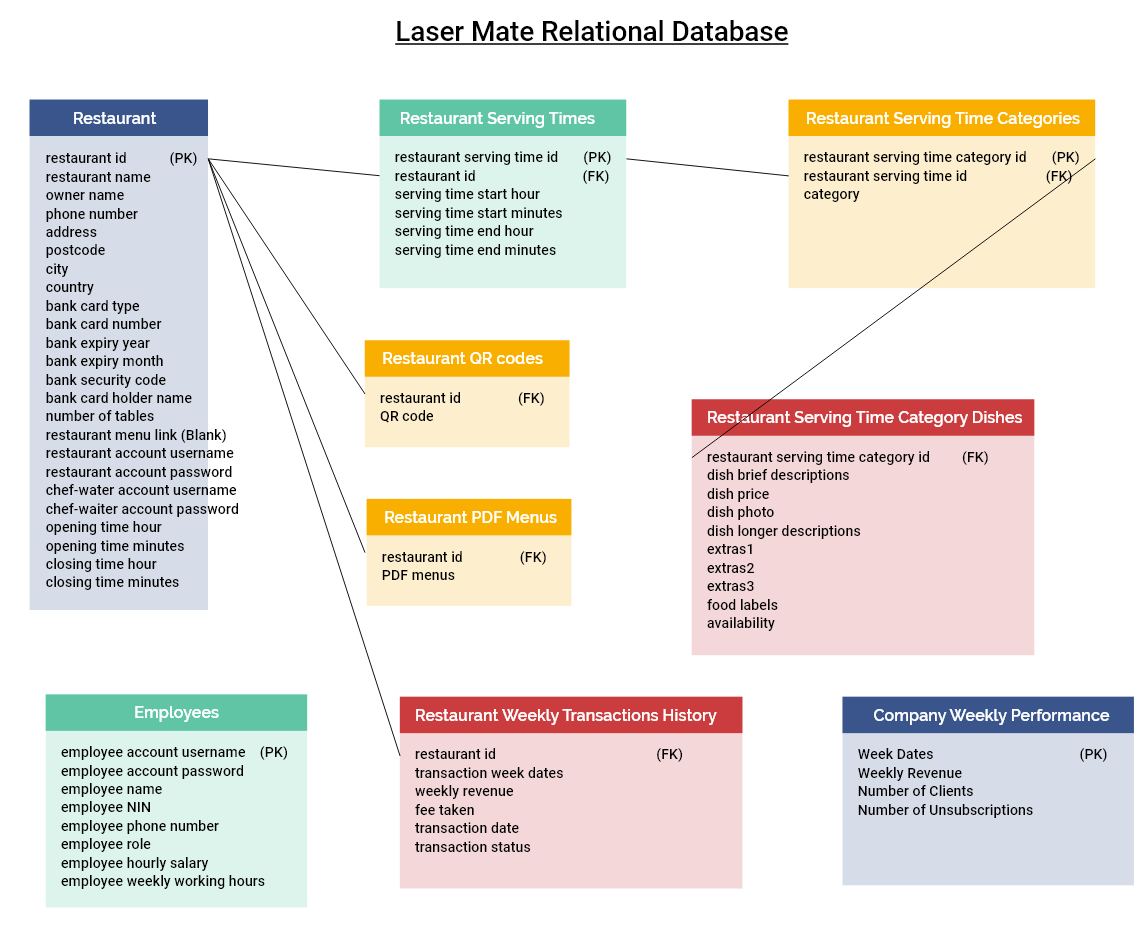
Cross platform between mobile and desktop

### Database Organization

* PostgreSQL
* High user traffics
* High volume of data processing, such as photos, text, and bank transactions.
* High speed
* Instagram

Tutorial Framework

* Select a platform that has video presentation since it’s easier to learn
* No Django official website, not video presentation
* LinkedIn learning is a poor platform since their work is too basic
* YouTube because it attracts a larger pool of resources internationally, therefore having more up-to-date and higher quality resources that could adopt for a wider diversity of problems
* Udemy needs to pay. Although having the table of contents but costs money risks chances. Lower quality than YouTube
* Find coding examples in GitHub for debugging programming issues



The database design for our software implementation follows the standard relational mapping. Each relation will have a primary/foreign key that uniquely identifies a tuple. Data fields that share a one-to-one relationship under the same data category will place in the same relation.

For data sets that share a one-to-many relationship, we will create a new relation which brings the primary key of the “one relation” as a foreign key. This will allow us to have many tuples based on that unique foreign key. If a relation requires a unique identification for further primary-foreign key mapping, the tuple will create a primary key that uniquely represents the full tuple, including the foreign key of the mapping relation.

Our database design does not contain a many-to-many relationship.

### Overall Coding Strategies

1. Develop a toy version using relevant YouTube tutorials
2. Find, filter and prioritize relevant programming tutorials

* First look at the final product to see if it is coded in React.JS and Django.
* This tutorial must cover all the aspects of the coding needed by the real project. Scan through the headings. An critical component Laser Mate! app needs include Django Rest framework, the ability to add, edit, delete, search and update records. Other software guides include responsive design, deployment, dynamically display information from the database, testing, login, registration,
* Look at the codebase and see if it is coded adequately in GitHub. Must have GitHub example code linking from YouTube
* Video demonstrate the step-by-step guide since the code solutions is not the full solutions. You must install the correct packages and use testing devices such as Postman for testing whether a subset of codes work.
* Once gather 5-7 relevant example codes, you will have a wider scope for the potential solutions you could have before diving deep to the videos. Prioritize the yotube project that is closest to the above criteria.
* Start watching the videos and do all the things they tell you so that you are starting to learn what is going on. Don’t skip the videos since the videos are there to explain you the concepts. Without the videos, you wouldn’t be able to understand the meaning behind each of the codebase. At least, skip the videos until you understand over 80% of the code.
* To speed up programming progress, a good practice is not to hand tpye every sippnet of code while watching the videos. Instead watch the videos without much pauses and just copy and paste the required code to the code editor from the GiTlab example project.
* Watch the full video then work on the real project. The reason to watch the full video first is that you can absorb the materials better when you understand how to code the full program instead of just understanding parting of the project which just leave confusion.
* write test cases while doing the project. It allows you to double check your work if you do test cases while you are doing that part and after you have the full project.

## Evaluation

### Evaluation for Paper Prototype

* Paste your question sets using google form

Usability Heuristic Evaluation

Semi-Structured Interview & Questionnaire

* Supervisor
* Family member
* Literature Review

### Evaluation for Digital Prototype

* Paste your google form here

Cognitive Walkthrough & Semi-Structured Interview & Questionnaire

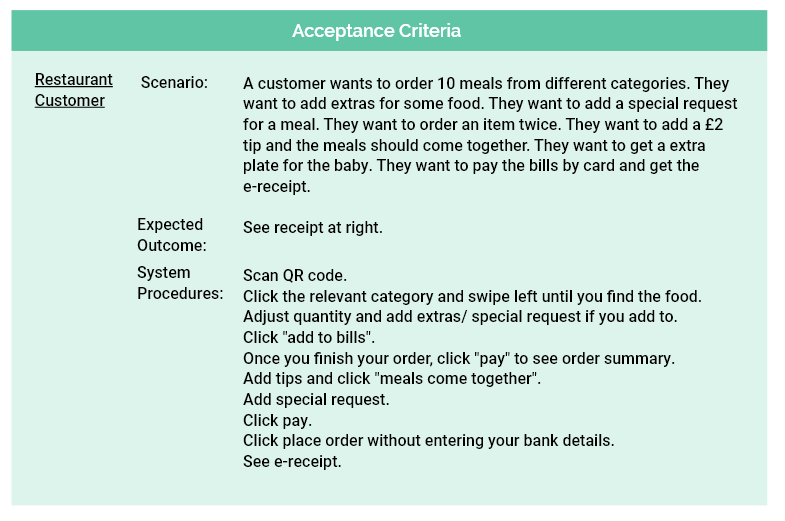
* Supervisor
* Family member
* Friends

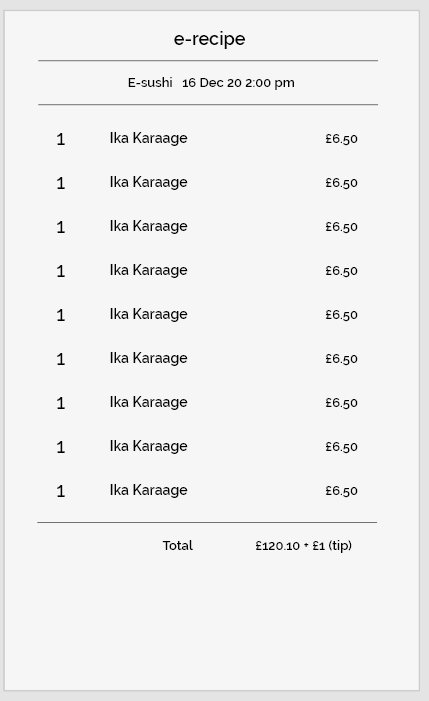
### Testing Documentations

* Paste your expected and actual outcomes here
* Postman – teach the reader how you use it for API testing

#### Acceptance Criteria v1

Acceptance criteria describes an example of practical tasks the system could perform. As this scenario will feed into software testing, it will also include an expected outcome which compares with the system’s actual outcome. System procedures is another aspect of the acceptance criteria that describes the instructions that implements the aforementioned scenario.





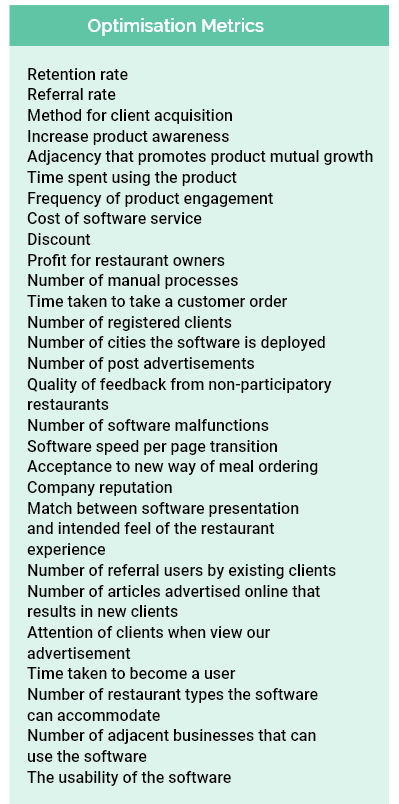
### Evaluation for Software Product

Cognitive Walkthrough & Semi-Structured Interview & Questionnaire

* Supervisor
* Family member
* Friends

### Future Work v1

The company should identify future work to outperform existing practices. By understanding the optimization metrics that determine the product’s success, it would make it harder for other ventures to build a better product. The critical aspects for consideration include keeping customers continue using the system and will not leave; how we can grow the business internationally; and how to maintain system capability. Furthermore, we could consider the points mentioned in the “Value Proposition Canvas” and apply some relevant growth strategies.



# Bibliography

# Appendix

# Video Presentation

1. Demonstrate the final product & the test cases using Q.R. code videos and screenshot photos

== not in dissertation but video=== testing

Customer

* Scan QR code
* workflow to order 10 meals from all different categories; add special requests; add extras; increase meal quantity.
* Read order summary; add quantity; check meal descriptions and prices; swipe through meal over a few pages.
* Check total price; give tips; check new price;
* Change meal together and meal separately;
* Add special requests.
* Order and produce e-receipt

Restaurant Owner

* Account registration; email account confirmation; see whether new account is automatically aggregated in the company staff account
* Forgetting password; email password retrieval;
* Add serving time; add categories; add meal details; edit serving time to check whether other data (categories, meal details) will be changed
* Delete serving time; check whether other data (categories, meal details) will be deleted
* Add restaurant info data; see whether restaurant name and address will be changed in the customer interface; check whether restaurant phone number and owner name will be changed in the company staff interface
* Check whether business info Q.R. code works; whether it shows the customer interface for the restaurant.
* Check whether help page will redirect user to video documentation page.

Company Staff

* do later \*\*\*

=============