**Eat With Us**

**Created by Tyler Pearson, Mansi Chaubey, Jasmine Gutierrez**

**CS 440**

**at the**

**University of Illinois Chicago**

**Spring Semester 2021**

**Table of Contents**

[List of Figures](#_heading=h.3znysh7) 5

[Project Description](#_heading=h.1t3h5sf) **6**

[Project Overview](#_heading=h.4d34og8) 6

[The Purpose of the Project](#_heading=h.2s8eyo1) 6

[The User Business or Background of the Project Effort](#_heading=h.17dp8vu) 6

[Goals of the Project](#_heading=h.3rdcrjn) 7

[Measurement](#_heading=h.26in1rg) 7

[The Scope of the Work.](#_heading=h.lnxbz9) 7

[The Current Situation](#_heading=h.35nkun2) 7

[The Context of the Work](#_heading=h.1ksv4uv) 8

[Work Partitioning](#_heading=h.44sinio) 8

[Competing Products](#_heading=h.2jxsxqh) 10

[The Scope of the Product](#_heading=h.z337ya) 10

[Scenario Diagram(s)](#_heading=h.3j2qqm3) 10

[Product Scenario List](#_heading=h.1y810tw) 11

[Individual Product Scenarios](#_heading=h.4i7ojhp) 11

[Stakeholders](#_heading=h.2xcytpi) 12

[The Client](#_heading=h.1ci93xb) 12

[The Customer](#_heading=h.3whwml4) 12

[Hands-On Users of the Product](#_heading=h.2bn6wsx) 12

[Maintenance Users and Service Technicians](#_heading=h.qsh70q) 13

[Other Stakeholders](#_heading=h.3as4poj) 13

[User Participation](#_heading=h.1pxezwc) 14

[Priorities Assigned to Users](#_heading=h.49x2ik5) 14

[Mandated Constraints](#_heading=h.2p2csry) 15

[Solution Constraints](#_heading=h.147n2zr) 15

[Implementation Environment of the Current System](#_heading=h.3o7alnk) 15

[Partner or Collaborative Applications](#_heading=h.23ckvvd) 16

[Off-the-Shelf Software](#_heading=h.32hioqz) 16

[Anticipated Workplace Environment.](#_heading=h.1hmsyys) 16

[Schedule Constraints](#_heading=h.41mghml) 16

[Budget Constraints](#_heading=h.2grqrue) 17

[Naming Conventions and Definitions](#_heading=h.vx1227) 17

[Definitions of Key Terms](#_heading=h.3fwokq0) 17

[UML and Other Notation Used in This Document](#_heading=h.1v1yuxt) 17

[Data Dictionary for Any Included Models](#_heading=h.4f1mdlm) 18

[Relevant Facts and Assumptions](#_heading=h.2u6wntf) 18

[Facts](#_heading=h.19c6y18) 18

[Assumptions](#_heading=h.3tbugp1) 18

[Requirements](#_heading=h.28h4qwu) **19**

[Product Use Cases](#_heading=h.nmf14n) 19

[Use Case Diagrams](#_heading=h.37m2jsg) 19

[Product Use Case List](#_heading=h.1mrcu09) 19

[Individual Product Use Cases](#_heading=h.46r0co2) 20

[Functional Requirements](#_heading=h.2lwamvv) 22

[Data Requirements](#_heading=h.111kx3o) 23

[Performance Requirements](#_heading=h.3l18frh) 23

[Speed and Latency Requirements](#_heading=h.206ipza) 23

[Precision or Accuracy Requirements](#_heading=h.4k668n3) 24

[Capacity Requirements](#_heading=h.2zbgiuw) 24

[Dependability Requirements](#_heading=h.1egqt2p) 24

[Reliability Requirements](#_heading=h.3ygebqi) 24

[Availability Requirements](#_heading=h.2dlolyb) 24

[Robustness or Fault-Tolerance Requirements](#_heading=h.sqyw64) 24

[Safety-Critical Requirements](#_heading=h.3cqmetx) 25

[Maintainability and Supportability Requirements](#_heading=h.1rvwp1q) 25

[Maintenance Requirements](#_heading=h.4bvk7pj) 25

[Supportability Requirements](#_heading=h.2r0uhxc) 25

[Adaptability Requirements](#_heading=h.1664s55) 26

[Scalability or Extensibility Requirements](#_heading=h.3q5sasy) 26

[Longevity Requirements](#_heading=h.25b2l0r) 26

[Security Requirements](#_heading=h.kgcv8k) 26

[Access Requirements](#_heading=h.34g0dwd) 26

[Integrity Requirements](#_heading=h.1jlao46) 27

[Privacy Requirements](#_heading=h.43ky6rz) 27

[Audit Requirements](#_heading=h.2iq8gzs) 28

[Immunity Requirements](#_heading=h.xvir7l) 28

[Usability and Humanity Requirements](#_heading=h.3hv69ve) 28

[Ease of Use Requirements](#_heading=h.1x0gk37) 28

[Personalization and Internationalization Requirements](#_heading=h.4h042r0) 29

[Learning Requirements](#_heading=h.2w5ecyt) 29

[Understandability and Politeness Requirements](#_heading=h.1baon6m) 29

[Accessibility Requirements](#_heading=h.3vac5uf) 29

[User Documentation Requirements](#_heading=h.2afmg28) 30

[Training Requirements](#_heading=h.pkwqa1) 30

[Look and Feel Requirements](#_heading=h.39kk8xu) 30

[Appearance Requirements](#_heading=h.1opuj5n) 30

[Style Requirements](#_heading=h.48pi1tg) 30

[Operational and Environmental Requirements](#_heading=h.2nusc19) 30

[Expected Physical Environment](#_heading=h.1302m92) 30

[Requirements for Interfacing with Adjacent Systems](#_heading=h.3mzq4wv) 30

[Productization Requirements](#_heading=h.2250f4o) 31

[Release Requirements](#_heading=h.haapch) 31

[Cultural and Political Requirements](#_heading=h.319y80a) 31

[Cultural Requirements](#_heading=h.1gf8i83) 31

[Political Requirements](#_heading=h.40ew0vw) 31

[Legal Requirements](#_heading=h.2fk6b3p) 32

[Compliance Requirements](#_heading=h.upglbi) 32

[Standards Requirements](#_heading=h.3ep43zb) 32

[Design](#_heading=h.1tuee74) **32**

[System Design](#_heading=h.4du1wux) 32

[Design goals](#_heading=h.2szc72q) 32

[Current Software Architecture](#_heading=h.184mhaj) 32

[Proposed Software Architecture](#_heading=h.3s49zyc) 33

[Overview](#_heading=h.279ka65) 33

[Class Diagrams](#_heading=h.meukdy) 33

[Dynamic Model](#_heading=h.36ei31r) 34

[Subsystem Decomposition](#_heading=h.1ljsd9k) 34

[Hardware / Software Mapping](#_heading=h.45jfvxd) 35

[Data Dictionary](#_heading=h.2koq656) 36

[Persistent Data Management](#_heading=h.zu0gcz) 36

[The Database will be constantly updated if changes were made. The company's finance team will receive reports from the client either weekly or monthly depending on what the client’s package is.](#_heading=h.hg9pl6yku2qm) 36

[Access Control and Security](#_heading=h.3jtnz0s) 36

[Global Software Control](#_heading=h.1yyy98l) 36

[Boundary Conditions](#_heading=h.4iylrwe) 36

[Subsystem Services](#_heading=h.2y3w247) 36

[User Interface](#_heading=h.1d96cc0) 36

[Object Design](#_heading=h.3x8tuzt) 38

[Object Design trade-offs](#_heading=h.2ce457m) 38

[Interface Documentation guidelines](#_heading=h.rjefff) 39

[Packages](#_heading=h.3bj1y38) 39

[Class Interfaces](#_heading=h.1qoc8b1) 39

[Test Plans](#_heading=h.4anzqyu) **39**

[Features to be tested / not to be tested.](#_heading=h.2pta16n) 39

[Pass/Fail Criteria](#_heading=h.14ykbeg) 40

[Approach](#_heading=h.3oy7u29) 41

[Suspension and resumption](#_heading=h.243i4a2) 41

[Testing materials ( hardware / software requirements )](#_heading=h.j8sehv) 43

[Test cases](#_heading=h.338fx5o) 43

[Testing schedule](#_heading=h.1idq7dh) 44

[Project Issues](#_heading=h.42ddq1a) **44**

[Open Issues](#_heading=h.2hio093) 44

[Off-the-Shelf Solutions](#_heading=h.wnyagw) 45

[Ready-Made Products](#_heading=h.3gnlt4p) 45

[Reusable Components](#_heading=h.1vsw3ci) 45

[Products That Can Be Copied](#_heading=h.4fsjm0b) 45

[New Problems](#_heading=h.2uxtw84) 46

[Effects on the Current Environment](#_heading=h.1a346fx) 46

[Effects on the Installed Systems](#_heading=h.3u2rp3q) 46

[Potential User Problems](#_heading=h.2981zbj) 46

[Limitations in the Anticipated Implementation Environment That May Inhibit the New Product](#_heading=h.odc9jc) 46

[Follow-Up Problems](#_heading=h.38czs75) 46

[Migration to the New Product](#_heading=h.1nia2ey) 47

[Risks](#_heading=h.11si5id) 47

[Costs](#_heading=h.3ls5o66) 47

[Waiting Room](#_heading=h.20xfydz) 47

[Ideas for Solutions](#_heading=h.4kx3h1s) 48

[Project Retrospective](#_heading=h.302dr9l) 48

[Glossary](#_heading=h.3z7bk57) **48**

[References / Bibliography](#_heading=h.2eclud0) **49**

[Index](#_heading=h.thw4kt) **50**

### **List of Figures**

[Figure 1 - Context of the Work](#_heading=h.3dy6vkm) 8

Figure 2 - Scenario Diagram [11](#_heading=h.3dy6vkm)

Figure 3 - Use Case Diagram 20

Figure 4 - Data Structure Diagram 24

Figures 5, 6, 7 - UI Design Prototypes 36, 37

Figure 8 - Hardware/Software Mapping Deployment Diagram 35

Figure 9 - Client Class UML Diagram 33

Figure 10 - Dynamic Model Diagram 34

**List of Tables**

[Table 1- Work Partitioning for the Product](#_heading=h.tyjcwt) 8

Table 2 - Product Scenario List [11](#_heading=h.tyjcwt)

Table 3 - Product Use Cases 20

Table 4 - Individual Use Cases 21, 22

# Project Description

## Project Overview

The website is for local restaurant owners to be able to better communicate with the public with better access to digital marketing support. This will allow for restaurants, who have been suffering the most during the current pandemic, to be able to communicate with marketing companies, to better serve their needs.

## The Purpose of the Project

### The User Business or Background of the Project Effort

Local restaurants: access to digital marketing and relevant metrics, allow for better, more localized marketing during the current pandemic related recession, provide quotes and general information for the type of marketing being asked for. In other words, help restaurants continue to get customers during an economic downturn.

### Goals of the Project

Provide businesses a platform to be able to promote and keep their business afloat due to the pandemic.

Provide those who do not know how to promote their business online a platform and guide them how to promote it and keep engagement with their customers and to get new ones within the process.

Allow businesses to avoid huge delivery fees that they get charged while working with apps like uberEats, grubhub etc.

### Measurement

Collect data on what services are being purchased and at what quantities/frequencies by the restaurant owners. Ideal goal would be for each restaurant to have at least one marketing strategy/company selected, with a return on investment. Can determine the return via surveying the restaurants and marketing firms for satisfaction and previously mentioned data collection

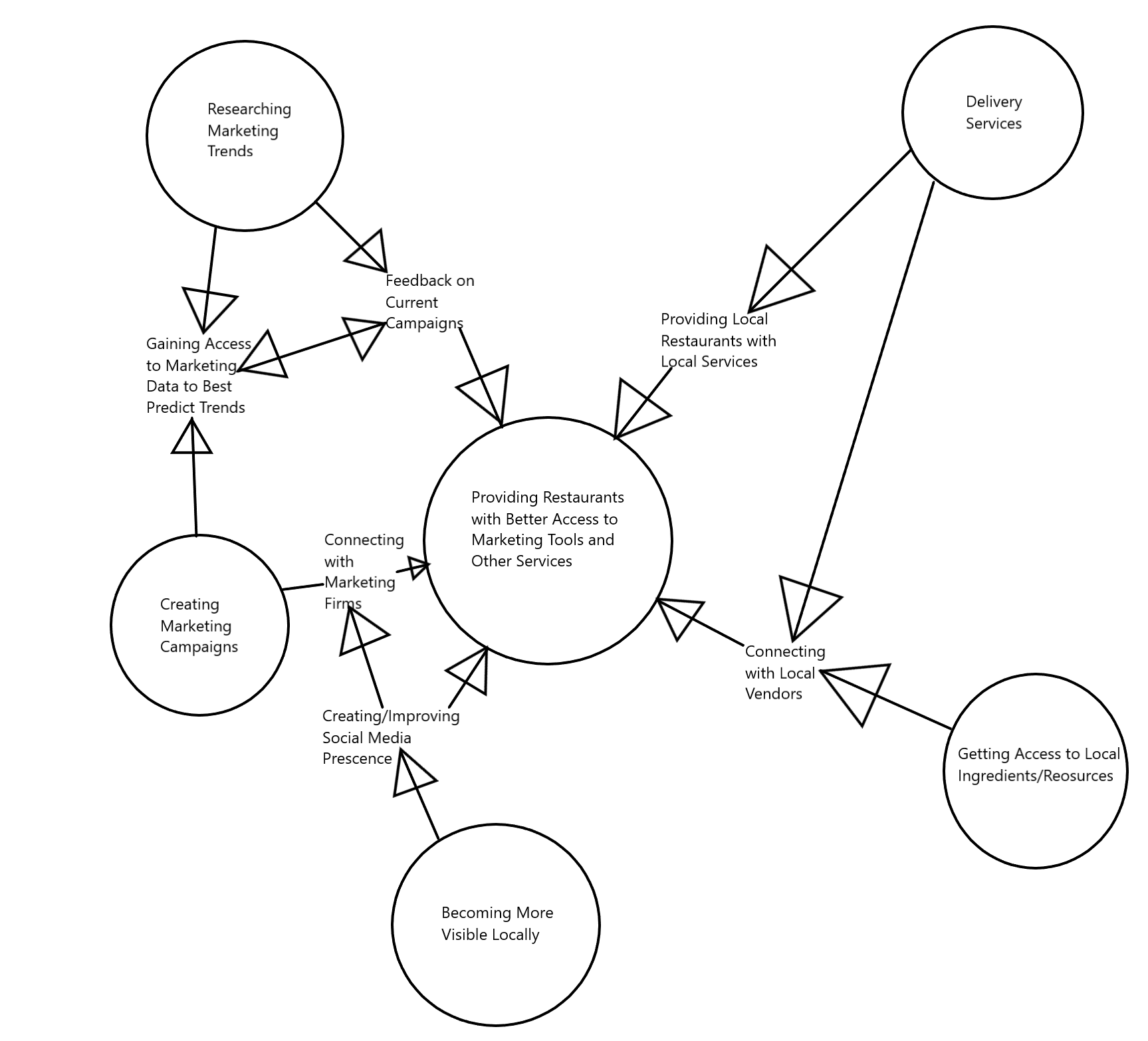
## The Scope of the Work*.*

Our product will address local restaurant’s needs for better digital marketing solutions.

### The Current Situation

During the current pandemic many restaurants are going out of business because they are unable to get enough customers for dine-in, and they do not have delivery available. A lot of ‘old-school’ restaurants do not have an online presence which means that potential customers do not know about the existence of such restaurants. With this website they will be able to combat such problems. More or less, the idea is to make it easier to get access to marketing strategies and services, without having to resort to a middleman.

### The Context of the Work



### Work Partitioning

|  |  |  |
| --- | --- | --- |
| **Event Name** | **Input & Output** | **Summary** |
| 1. Gathering client needs for marketing | Form (In) | Create a form that will allow user to select needed services and search for services not listed on the form in the beginning |
| 1. Website gives quotes for requested services | Service List (out) | For each item selected in above form, display page with companies providing services and price quotes |
| 1. Website displays breakdowns on costs and expenses | Monthly Stats Page (out) | Display the costs for all marketing services purchased monthly. Pie chart to graph where the money is being spent. Create a button to pay monthly bill |
| 1. Restaurant owner/manager requests to pay bill | Billing and Accounts Page (in/out) | Ask for the user to log in again. When authenticated, show the amount owed and owed to whom. Feature the payment option after the owed amount is shown. Also same screen for actually buying services from firms/contractors. |
| 1. Work Order is Placed | Service List (out) | When a service is selected by the user and the service selected for purchase, page loads in a in-page confirmation and delivers user to specifications screen |
| 1. Work Order Specifics Recorded | Work Order Page (in) | Form is generated for the work order, displaying the firm/contractor service is being purchased from, type of service, and cost per time period. User can request specific changes from the standard service through a text box that sends request to firm/contractor after order submission and payment |

### Competing Products

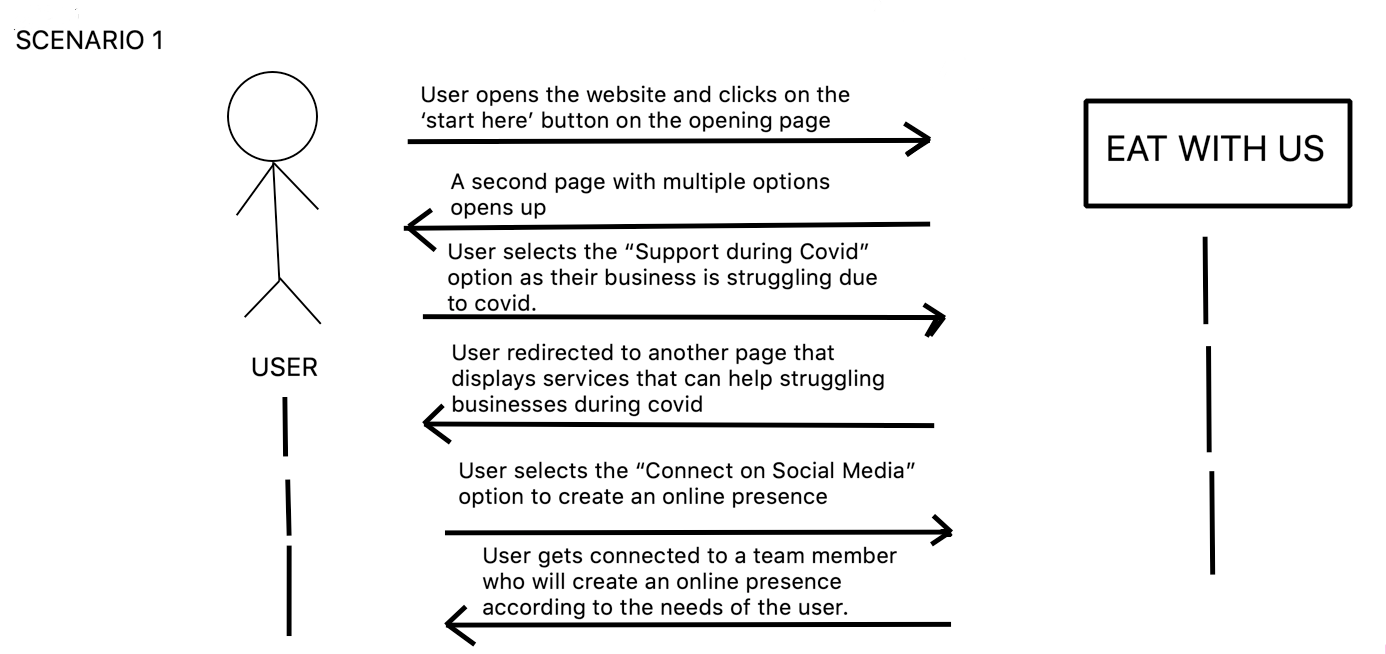
There are many delivery services available like: ubereats, postmates, grubhub which help restaurants set up delivery services and give them an option to be found online when a customer is ordering through those apps, but these platforms charge a 30-40% service/delivery fee per order from these restaurants, which often leads to them not making enough money to stay in business. And while there are websites that offer marketing solutions, they are not combined with other restaurant related services, driving the total cost up for the restaurant owner.

With this website, such businesses will now have a chance to have full control over their online presence and will be able to avoid paying huge amounts of money to third party websites/apps.

## The Scope of the Product

The product is designed to handle the majority of the work related to getting a marketing campaign underway, from finding the correct strategy to employ to setting up the meeting between the restaurant and the marketing firm that can best serve their needs, to handling the financial side of the marketing agreements (payments, subscriptions).

### Scenario Diagram(s)



### Product Scenario List

|  |  |
| --- | --- |
| Scenarios | External actors involved |
| 1. User opens the website and clicks on the ‘start here’ button on the opening page | Internet Connection and Connected Device |
| 2. A second page with multiple options opens up | Internet Connection and Connected Device |
| 3. User selects the “Support during Covid” option as their business is struggling during covid | Services provided by local vendors |
| 4. User redirected to another part that displays services that can help struggling businesses during covid | Companies that provide those services |
| 5. Users select the “Connect on Social Media” option to create an online presence. | Information has been filled out |
| 6. Users gets connected to a team member who will create an online presence according to the needs of the user. | Availability of the team and team’s connection |

### Individual Product Scenarios

A restaurant owner finds out about our service and is interested in avoiding huge fees associated with third party apps and delivery services such as ubereats. His business is already struggling due to the pandemic and on top of that his business is charged 30-40% in delivery fees. This has led to the business struggling to keep its doors open. He opens our website and reads the description of what the website offers. He then clicks on the “Start Here” button, which redirects him to a second page where he can see all the services that are offered. After this, he clicks on the “Current Issues” button and is prompted to fill out a form about all the issues his business has been facing due to the pandemic. After this he is connected to a team member who helps set up his profile. After this, he adds menu items and pictures to finish the profile set up. Now, he can also put in work orders and get in contact with local vendors for a small monthly fee. He can also see his monthly expenses breakdown with detailed graphs and charts.

## Stakeholders

### The Client

Currently, there is no client attached to the product, and as such, the development team (Group 13) is acting as the client until one can be found.

### The Customer

There will be two major types of customer that will be paying a subscription when using the product: restaurant owners/managers and marketing firms/contractors. The restaurant owner/manager will be subscribing to the product as a way to help grow their business during a period of economic recession and help to bring in customers. The marketing firm would not be paying a subscription per se, but more of a percentage of the money used by the restaurants to purchase their services. For the restaurants, the subscription will be a small amount of money, no more than $100/month, while the marketing firms would be paying at maximum 40% of the sale per purchase of their services.

### Hands-On Users of the Product

Generally, the main users of the website will be restaurant owners and managers, who will be responsible for running the restaurant and promoting it to attract business. In terms of restaurant experience, the user should be a journeyman or master at their jobs, so more technical language related to the restaurant industry would be applicable and appropriate to use. There will be another type of user on this website as well, the marketing firms showcasing their services on the website. Naturally, they are going to be masters at their craft, and should be capable of using the product with minimal effort on their part.

The website is meant to make the job easier for the user and not want to tear their hair out from an overly complicated program. In other words, the program should be simple enough that a novice with very little experience with technology can use it easily and effectively with minimal effort.

The website itself should be neutral towards concepts like age or gender identity, and should not make any distinction from person to person unless requested. There should be an option to change the language used for it, however, as it is not only English speakers that run restaurants/marketing firms and would be using the product.

Just because someone may have a disability does not mean that they are incapable of running a restaurant, so it is imperative that accessibility is key for the product. When dealing with physical disabilities, there needs to be the option for tools like text-to-speech and vocal dictation present in the website, as well as an option to change the size of icons, text, and other items in the UI that may be difficult to see for someone with poor eyesight.

When dealing with intellectual disabilities, simplicity is key. Any colors on the screen should NOT be blazingly bright, as in neon colors, and should be calmer, more neutral tones. Sounds made by the website should not be as loud as to be disorienting for someone with auditory sensitivity. The language on the site should be as simple as possible without the loss of information, as in complex enough for the message to get through completely, but not so complicated that people would get lost reading it.

### Maintenance Users and Service Technicians

Installation, maintenance, and updating will be done by the main users, restaurant owners/managers and marketing firms. The main idea is that installation should be as simple as logging onto the website, and that updates can be done automatically when new updates are pushed to production. If there are needs outside of basic operations (updating and such), there will be assistance available either through emailing the developers for help or trying to do self help through a general help wiki of sorts.

### Other Stakeholders

As it currently stands, the project currently does not have stakeholders, but will hopefully be able to attract the attention of a major firm involved in either the technology or marketing firms. As a matter of thought however, there are a few other types of stakeholders that would be interested in the product not yet listed:

Business Experts: As stated earlier, the product is designed to help restaurants get better connections to marketing firms. A product like this would be very beneficial to Business Experts and Business Assessment firms, as it means that they have to do less research and decision making on marketing and can focus more on other, more pertinent areas. In the event that there may be a business expert and a marketing expert/firm are both stakeholders and wishing to have access to the same information, this can be remedied easily

Testers: Building any major program requires quite a bit of testing, and this one will be no different to say the least. Considering that the website will be handling payment for marketing services, it will need extensive testing to ensure that there will be no issues with payment processing, either on our end or through a service like PayPal. The Product and Quality Assurance Testers will be absolutely instrumental in the development of the product, and will work hand in hand with the Security Experts in the wings.

Security Experts: Security is going to be the top priority for the website outside of its main purpose. The restaurants and the marketing firms will have accounts created on the site, and many will be paying for services through it as well. With this in mind, we would want to have security as airtight as possible, to minimize any attempts to break into a customer’s account or gain access to financial information. There is also the highly unlikely possibility of a malicious person gaining access to a restaurant’s account and ordering a marketing campaign that goes directly against the business. By having a Security Expert as a stakeholder for the product, the product will be prodded and poked constantly through development, to ensure that the worst that can happen does not and that any security concerns are minimized.

### User Participation

The users will be massively important to the development of the product, as they are going to be the main factor for shaping the website properly. The user will be providing input on usability requirements, as in what services need to be included besides the basic functionality, ratings systems, as in feedback on the quality of the provided services, and in terms of interface design, as in how they wish for the website to look and how they can customize their profile and company page(s).

In simple terms, there will be allocation needed for user input on the development as well as feedback on design, bugs, etc.

### Priorities Assigned to Users

Key User(s): The restaurant owners/managers are our biggest concern for feedback, as they are the reason the product is being developed in the first place. Any issues generated by them are to be of the highest priority, and any suggestions should be at least slightly considered. Suggestions made by them should only be rejected if they would impact the whole of the product and/or its other users.

Secondary Users:The marketing firms advertising their services on the platform will be second behind the key users for priority, but only just slightly, as they are still what the key users will be spending money on. They should be treated respectfully when they have issues or suggestions, but if those suggestions were to cause problems for the key users, they are to be ignored and/or politely rejected if possible.

Unimportant Users: The unimportant users are pretty much the ones browsing the site without having spent much time on it or without engaging with the service providers on the site. In other words, if they are just browsing without spending, anything they say, unless it is informing the team of a product-breaking issue, should be ignored. If needed, we can deliver these types of users advertisements and disable them if they create an account on the site for a trial period, and then disable permanently if they spend purchase services through the product. Any users that misuse the product should be dealt with directly. For the first offence, a warning, and the next offence will result in the deactivation of their account and a product wide ban if the offence is severe enough

## Mandated Constraints

### Solution Constraints

Description: The product will be capable of operating on multiple different web browser clients (Google Chrome, Microsoft Edge, Mozilla Firefox, Opera, Apple Safari).

Rationale: The clients will almost definitely not be delegating themselves to just one web browser, and will be accessing the product from multiple sources.

Fit Criterion: The product will be tested to be compatible with the listed browsers.

Description: The product will operate as lightly as possible, and try to minimize the amount of data and memory needed to operate it.

Rationale: Clients will not all have the most up to date and powerful PCs within their workplace and will have varying broadband speeds and possible data caps.

Fit Criterion: Each web page in the product will be no more than 1MB in size and operate using React methodology for loading elements for the user. Videos will not be hosted by the website directly, and the team will not use any video components in the product anywhere.

Description: The product will stay up to date with all security concerns during time of operation and consistently update security to minimize any hacking attempts.

Rationale: Customers will be purchasing work orders through the site, as well as paying their monthly bill. Firms/contractors will be receiving payments for work orders and also paying their bill through the site.

Fit Criterion: Development team will employ the services of security experts to continually improve on the security of the product, and will stay up to date with any major news regarding possible security/data breaches attempted.

### Implementation Environment of the Current System

Hardware needed for this system is any Graphical Operating System that has a web browser installed, capable of downloading files if needed, and a network connection. Development communications will happen in person, online, or through a phone. For customer and firm/contractor support, there will be a self-help page and a direct line to a public relations agent connected to the team if there are any problems that are not covered by the self help page.

### Partner or Collaborative Applications

The product will be compatible with reading PDFs and processing them as needed, as well as giving information about the customer’s costs and expenses in the form of a Microsoft Excel spreadsheet and/or CSV file.

### Off-the-Shelf Software

The product must be packaged with a server and database to hold not only the website itself but also to hold the customer’s and firm’s/contractor’s information and accounts, and must also come with a payment processing service as well, to handle the transactions that will happen through the site.

### Anticipated Workplace Environment*.*

Ideally, the work environment would be happening in a back office, away from the noise and distractions inherent in a kitchen environment. However, this is a very ideal environment for the product to exist in, and is in no way guaranteed in every case. In a non-ideal environment, the system being put in place may be in a more hectic area of the customer’s workplace, such as the front desk of the establishment or, worst case, the actual kitchen environment. As such, if any sound is to be made by the product, it must be volume controllable, either silent in more quiet environments or capable of being heard in a loud environment. Visual alerts will not be implemented with the product, as the possible environments that the product will be operating in can become health and safety hazards if visual distractions are introduced (kitchen environments).

As the product is meant for business applications, allowing the download of forms from the website itself is needed as well, as many businesses will wish to have access to paper copies of these documents for their own records.

Returning back to the physical environments the product will be operating in, touch screen capability will be implemented as well, as having to use a keyboard and mouse in a kitchen is anathema to cleanliness, which is king in the kitchen. An easy to clean screen with minimized areas for dirt and germs to gather is key, and will hopefully be how the product is to be implemented in a kitchen environment.

### Schedule Constraints

As it has been stated by the World Health Organization, as the effects of climate change really begin to ramp up in terms of environmental and economic impact, the likelihood of another new disease spreading globally and creating a new pandemic will increase as well. The current pandemic has already claimed the livelihoods of numerous restaurants across the United States, and that number will only continue to increase until some form of herd immunity is achieved. Another pandemic will most likely follow the same pattern as the current pandemic is. Therefore, it is imperative that the product release version 1.0 before the beginning of a new pandemic, whenever that may be. Pandemics are not something that can be predicted easily or planned for, like an election or a holiday, so it would be best to get a working, complete version of the product out the door as soon as possible to mitigate the chance that the team is “caught with its pants down”.

### Budget Constraints

The max that should be spent on development should revolve around whatever the developers, in this case college students, can afford. Considering that college students are stereotypically lacking in pretty much any type of disposable income, this should be incredibly close to the floor, as in no more than $2500, which is a stretch in of itself.

## Naming Conventions and Definitions

### Definitions of Key Terms

Customer: The representative of the restaurant, either the owner of the establishment or its manager, buying the work order from the firm/contractor.

Work Order: Order from a customer to a firm/contractor specifying what service they are requesting, what special requests the customer wishes to have fulfilled, the duration that the service will take place in, and the specific type of service requested (Digital Marketing Campaign, Newspaper/Local Radio Ads, Local Delivery Service Contracts, etc.)

Local: Referring to businesses, resources, and services within the operating area of the customer.

Firm/Contractor: referring to the marketing firms, delivery service companies, and other relevant service companies that offer their services through the site.

Product: in reference to the project itself, which will operate as a website. Can be used interchangeably with “website” when referring to the project.

Service List: The directory of services provided by the firms/contractors operating on the site itself. The customer will be shown this after they input a search request to find the work order that best fits their needs.

Client: Refers to both the customers and the firms/contractors operating the product.

### UML and Other Notation Used in This Document

There are no relevant diagrams or notation used in this document so far.

### Data Dictionary for Any Included Models

Work\_Order = Invoice\_Number + Service\_Type + Specific\_Requests\_Sent\_By + Service\_Timeline + Service\_Duration + Cost\_Of\_Service + Service\_Provider + Is\_Repeating\_Request

Is\_Repeating\_Request: Boolean value that tells if this is a monthly service or a one time only service being requested in the work order

User\_Acct = User\_Name + User\_Address + User\_Workorder\_History + User\_Financial\_Info

## Relevant Facts and Assumptions

### Facts

The average traditional restaurant has a 3% to 5% profit margin. [7]

10,000, or 17%, of restaurants in the United States have permanently closed their doors because of CoVID-19. [6]

17% of restaurants close within the first year of operation (not the same as those closed because of the CoVID-19 pandemic). [8]

Customers will pay up to 40% of the cost of the food being delivered to them compared to what the restaurant is charging. [9]

### Assumptions

There will be no competing product released in the time it will take to develop and release the first working version of the product to market.

There will still be support for the product’s third party components, as in payment processing and server support.

The development team will assume that there will still continue to be enough restaurants open to warrant the continued development of the product.

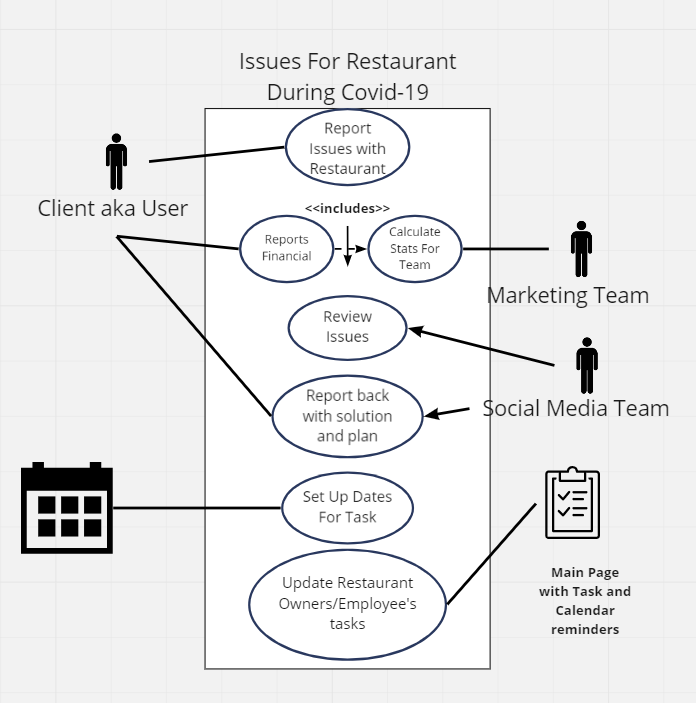
The development team will assume that the user will have access to the internet and all necessary software/hardware required of them, and that the user is one of the targeted users for this product.

# Requirements

## Product Use Cases

### Use Case Diagrams

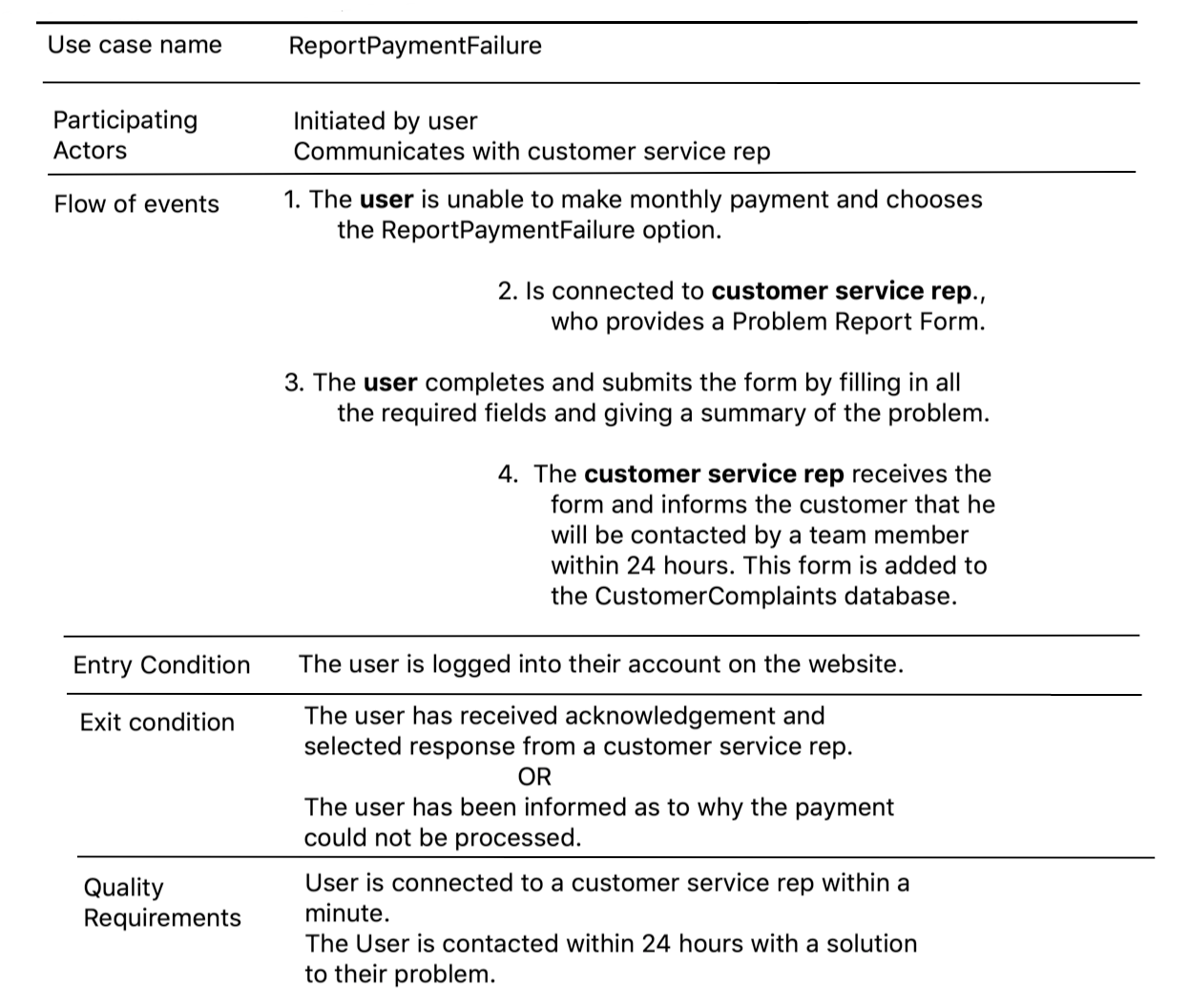
Use CaseDiagrams:

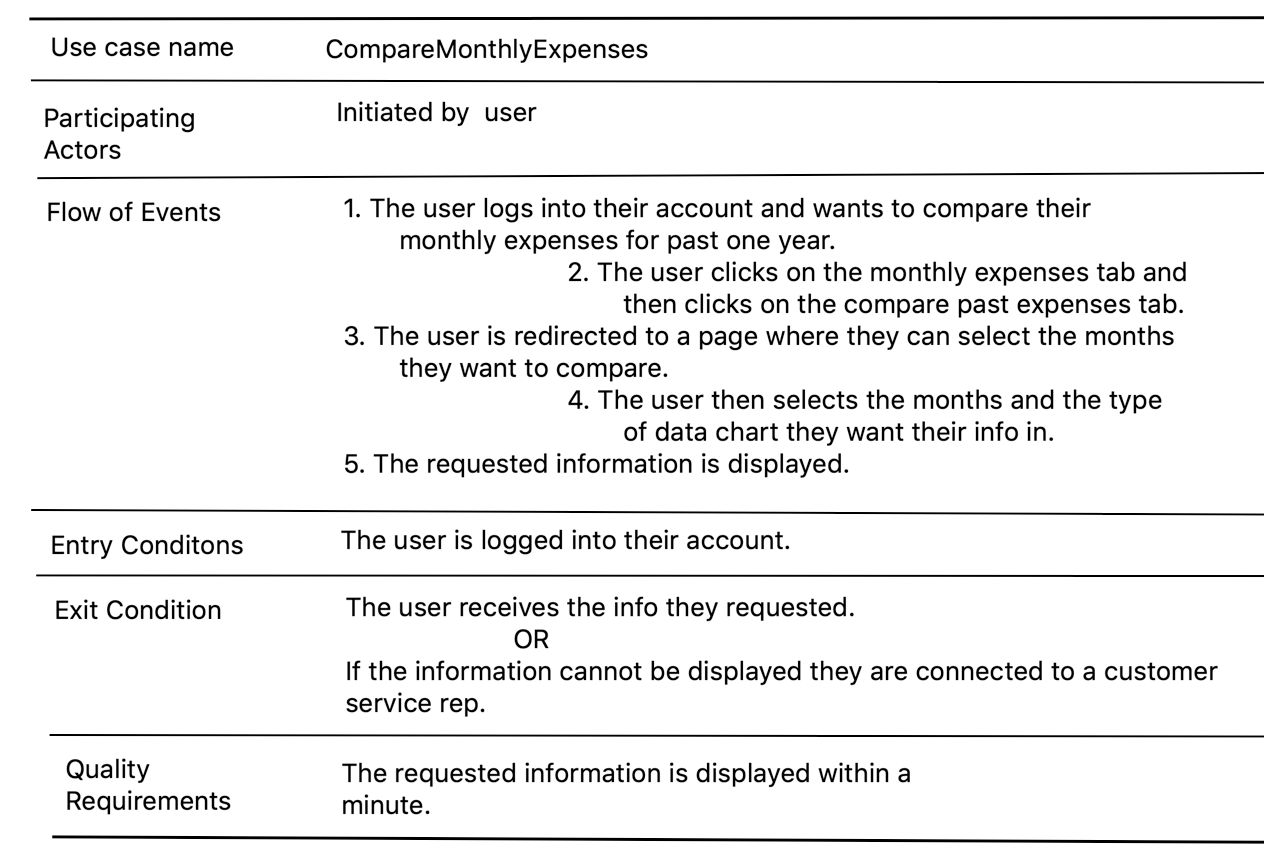


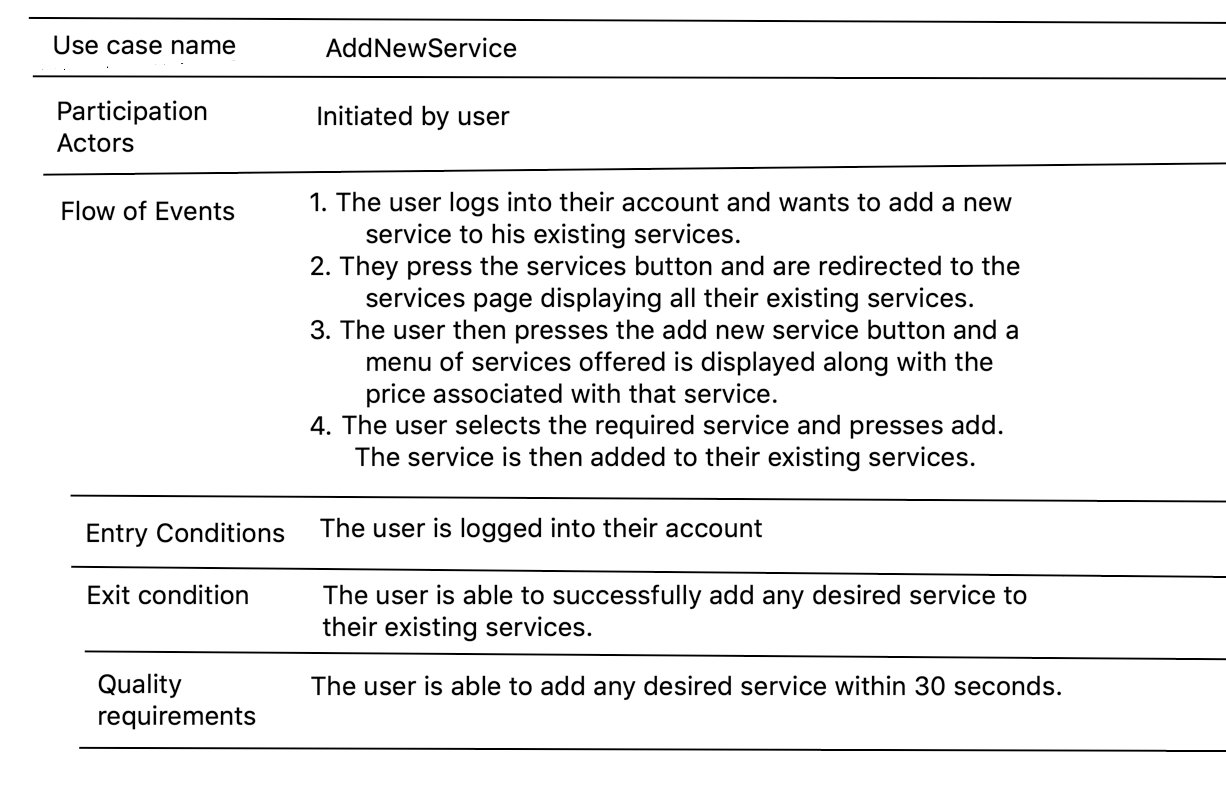
### Product Use Case List

|  |  |
| --- | --- |
| Use Cases | Actors |
| ReportPaymentFailure | User, Customer service rep. |
| CompareMonthlyExpenses | User |
| AddNewService | User |
| GetHelp | User, Customer service rep. |
| Begin | User, Customer service rep. |
| CreateSocialMediaPresence | User, Customer service rep. |

### Individual Product Use Cases







## Functional Requirements

* **Begin**

Description: When the user clicks on the begin button, they should be prompted to

fill out a form and after that the user should be connected to a customer service sales rep.

Rationale: The user should be able to submit their requirements and become familiar with how the website works.

Fit Criterion: Check if the user can use the website effortlessly and check if the information added by the user is correct.

Acceptance Criteria: The user should begin using the website efficiently and the website should have general information about what the user’s needs are.

* **GetHelp**

Description: User is able to ask for help if stuck on a feature on the website.

Rationale: It is important for users to be able to ask for help in order for them to

actually benefit from using the website.

Fit Criterion: After getting help the user is able to overcome the problem.

Acceptance Criteria: User feedback should be positive for customer service and

overall usability for the website.

* **CreateSocialMediaPresence**

Description: User is able to create a social media profile for their restaurants and

control their online presence through the website.

Rationale: It is important for restaurants to have a good online presence because

these days people always google the place they are interested in going to.

Fit Criterion: The user is able to create a good online profile with the help of the

social media team.

Acceptance Criteria: The profile created is fun and interesting.

* **AddNewService**

Description: The user should be able to add new services for their restaurant.

Rationale: It is important for the user to be able to add or remove services

according to their changing needs.

Fit Criterion: New service added without any problems.

Acceptance Criteria: The user should be able to successfully add a new service

and should be able to provide feedback if they encounter any problems.

* **CompareMonthlyExpenses**

Description: The user checks and compares monthly expenses.

Rationale: The user should be able to check and compare monthly expenses in

order to keep their spending in check and manage the restaurant efficiently.

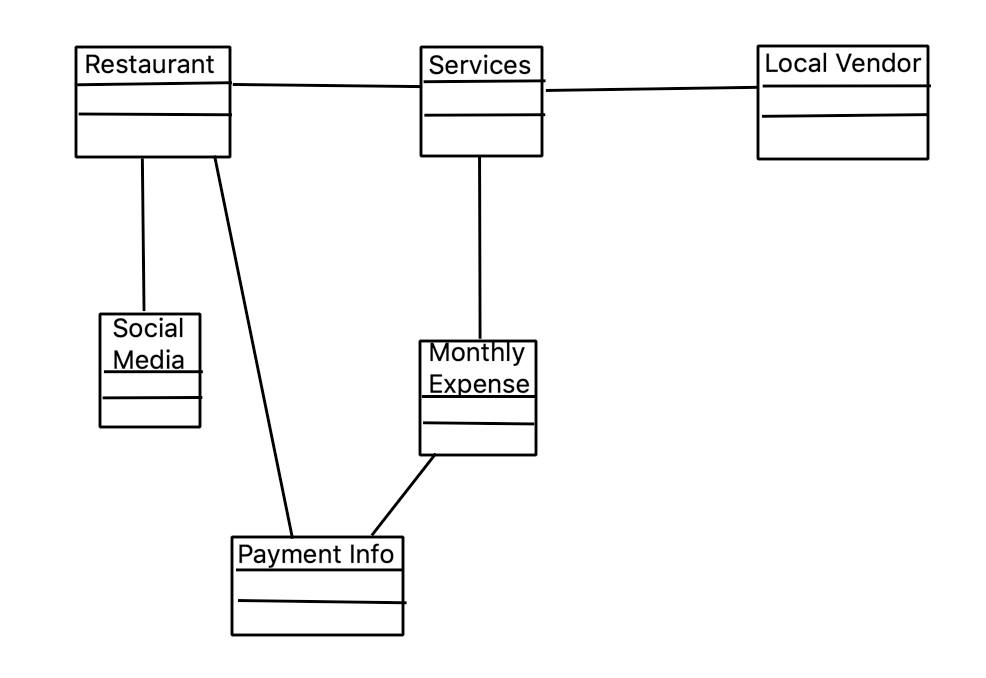
Fit Criterion: The user views correct and easy to read monthly reports.

Acceptance Criteria: When the users selects the monthly expense option, they

should be able to correct reports without any problems.

## Data Requirements

* Database containing monthly expense history for each restaurant: The database should be able to store all information regarding past and present monthly expenses for each restaurant. The user should be able to access this information. Failure to do either of these tasks will result in test failure.
* Database containing all services being used by every restaurant: The database should be able to store all the information regarding past and present services being used by every restaurant. Users should be able to view their past and current services. Failure to do either of these tasks will result in test failure.
* Customer Complaints Database: This database should be able to store all customer complaints sorted by months. Users should not be able to view this. Failure to do either of these tasks will result in test failure.



## Performance Requirements

### Speed and Latency Requirements

The product will operate fast enough to not interfere with the user’s day to day operations. Any and all responses from the product to the user will take no longer than 3 seconds, unless there is an issue on the user’s end. The same limit shall be imposed on all tasks that the user requests from the product.

Any changes made to the product during operation will be loaded in when the user next refreshes their page, where it will be automatically delivered to them.

Any reports sent out by the product to the user will take no more than one day from the time of generation to be delivered electronically.

### Precision or Accuracy Requirements

All monetary amounts will be rounded when appropriate and possible, with the value rounding up to the nearest value, and running to two decimal places. All times within the product will be synchronized to their respective time zones. Any measurements not otherwise mentioned will use the imperial system when inside the United States, and the metric system otherwise.

### Capacity Requirements

During standard business hours in all timezones the product is operating in, the product will cater to up to 200 users simultaneously during this period. During non-business hours, the maximum load the product will be expected to handle without issue will be 130 users. In the event of there being more users than the maximum load for the time period, the development team will attach a backup server to handle the excess load on the product.

## Dependability Requirements

### Reliability Requirements

No data will be lost or damaged because of a failure in the product’s systems, and the product’s systems will fail no more than one half-hour period a week. Any failures that occur will not last longer than that one half-hour period total. In the need of system maintenance, the product will warn the user two hours in advance, then one hour in advance, cutting this time in half until maintenance begins, as to avoid any issues with the users being blindsided with an unexpected maintenance period. These maintenance periods will be on a monthly basis at maximum, bimonthly preferably.

### Availability Requirements

The product will be available 24 hours a day, 365 days a year without any interruptions lasting longer than half a day. Ideally, the product will achieve at least 99% uptime while in operation. This will not include maintenance periods, as they will be expected downtimes and these downtimes will be broadcasted to the users before they happen.

### Robustness or Fault-Tolerance Requirements

In the event that the user loses connection to the internet for any reason, the product will still display all information on the current webpage being displayed to the user. If the user tries to load a new page without an internet connection, the product will not be expected to load that at all.

If the user is to lose connection to the product when entering in any information, the product will save any fields or information that has been provided beforehand, but not expected to save any information that was being loaded into the product. This will include strings being input by the user when using a chat function in the product.

### Safety-Critical Requirements

This product will be accessed by the user through a web browser, so the only physical component(s) needed for operating the product will be third party and not supplied by us. This does not mean that there are no safety measures in place, however.

The product will not be party to or cause any issues with the health and safety of the user and their device, as such, there will be absolutely zero cryptocurrency miners attached to the product under any circumstances. If it was there, it would impact the product’s performance and the lifespan of the user’s device. The product will also not be capable of causing any issues with the user’s health at all, such as causing an epileptic seizure or revealing the user’s location to other users and individuals outside of the product without the user’s consent.

## Maintainability and Supportability Requirements

### Maintenance Requirements

Data verification and validation will happen on a weekly basis, at a minimum once a month, to check and correct any errors that may have occurred with the data. All general maintenance, server maintenance, database maintenance, and otherwise, will be done on a regular basis, at minimum once a month. These systems are critical to the success and operation of the product, so their health is top priority.

The product will be easy to maintain, so that if a different team than the one that made the product is in charge of the product’s operation, they can do so easily. The product’s code will be heavily documented, with a general summary of each major file and its function being attached to the top as a comment. The users will not be expected to maintain the product, that duty will fall to the development team alone.

### Supportability Requirements

Most support will be accomplished with a self help page, to the extent that the product should be entirely self supporting if possible. This page will be designed the same way stylistically as the rest of the product, and will host information on how to resolve any possibly common problems that users may have when using the product. The page will be simple to navigate and easy to look and search through, where it will take no more than 3 minutes for a user to find what they were looking for on the self help page, assuming that their issue is one that is commonly reported.

However, there may be a time that the issue that a user has with the product cannot be resolved with a self help page. Therefore, the product will have a small help desk of no more than 5 employees. There will be no more than that as the self help page will be the main problem solving resource for the users, and that only the most uncommon or unsolved problems will be brought to the help desk’s attention.

### Adaptability Requirements

This product will be designed to operate not only in an office environment, but also in a front desk and restaurant kitchen environment as well. The product will be able to run on all major web browsers on desktop machines and mobile devices, which will allow it to run on any operating system that can run those browsers. In the future, the product will be able to be run in a mobile environment as a standalone application rather than using a web browser, as well as adding translations for the product for less common languages in the United States, such as Swedish or Chinese.

### Scalability or Extensibility Requirements

The product will be able to handle up to 10,000 transactions per hour within a year of its launch, and will increase this capacity by 10,000 transactions minimum per year. As more users begin to use the product, more data will be collected from them. Therefore, the product will have a database and server storage capacity of at least forty terabytes on launch, and will multiply this capacity by two every time the current filled capacity reaches four fifths of the total capacity. In terms of bandwidth, the product will be capable of handling up to 15,000 users within the first year, bringing up this number whenever possible without any major updates to the product.

### Longevity Requirements

The product shall be expected to operate within its operational budget without any major overhauls or upgrades for a period of three years, with updates during this time consisting almost exclusively of security updates and data redundancy operations.

## Security Requirements

### Access Requirements

As the product will be handling financial information for processes such as billing, the team will hire a security consultant to best prevent any security issues from happening, rather than waiting for an event to occur and handling it then and there. Physical access to the hardware components of the product, such as the main server or the databases, will be limited to those with the minimum security clearance as dictated by the security expert. At minimum, assuming that the team will be working in an office environment and not working remotely, all team members will have physical access to the main part of the office, but anywhere that would house critical components of the product or its data will be limited to the individuals listed above.

Data access for individual users will be limited to the user themselves and to a systems administrator or security expert if an issue arises with the user’s data that they cannot fix themselves or are having trouble accessing. The user can also limit what data the team can access as well, assuming that the data is not critical to the team’s support for the product or financial information. If the user loses access to their account, there will be two factor authentication processes and account recovery processes installed to allow the user to recover their account.

### Integrity Requirements

All data created by the users and stored by the product’s servers and databases will be checked and looked over with a program designed specifically for that purpose or by the hands and eyes of the development team when and if necessary. All data will be stored in a redundancy system, where there will be more than one copy of the collected data at any point in time, at minimum two copies of the data collection. Periodically, at minimum once every month, the data will be checked over for any corruptions or incorrectness and be fixed by hand if necessary or by restoring the data to a previous state from a redundant storage system.

The product will protect itself from intentional abuse by the user and from any attempts to intentionally destroy data without the express go ahead from the development team. Any attempts by a user to abuse the product or its data will result in a monetary fine if minor enough, and being completely removed and banned from the platform if severe enough.

### Privacy Requirements

The product will collect the minimum amount of data necessary to protect the development team from issues related to financial records and evidence of communications, and none of it will be sold to third parties or used by the team to gather advertising revenue. All information will be stored within a system of heavily encrypted databases, where new data will be held in a database connected to the system network for a period of one month, after which they will be moved to a database not connected to the development network, and will only be connected to it for the transfer of data. This will help to mitigate the chance that large amounts of data are lost to a malicious attack. This system will be accessible to the customer should they request that they wish to correct any non-financial information that has been collected.

As for smaller bits of information, such as cookies, the product will include the bare minimum amount of data necessary to have these services be operational.

The users will be made aware of the information policy and be notified of any changes of this policy that may happen at any time. They will be notified when their stored/collected data is being scheduled for a transfer to the offline database and when this transfer occurs.

### Audit Requirements

The product will record who visits the site itself and stores that as an encrypted log file. This file will be stored the same way as the data mentioned in the above section, inside a system of encrypted databases. All financial transactions and related information will be encrypted and stored in the same way, in case of an event like a warrant or federal inquiry.

### Immunity Requirements

The product will be tested thoroughly by security experts and will be equipped with enough security measures to give the users as much protection and confidence as possible when using the product. The product will scrub all inputs as standard and will employ a CAPTCHA service to limit the possibility of bot activity on the product.

## Usability and Humanity Requirements

### Ease of Use Requirements

* The product should be tested to check if it can be easily used by people of different ages and experiences, it should also be checked for ease of use with users who do not understand english.
* The product/website can be used easily irrespective of age. The user does not need to remember much about using the product as the website is extremely straightforward and users are usually redirected from one page to another without any confusion. The chances of error committed by the user are very low as every link and heading is pretty self explanatory. The user will also have the option to chat with a team member in case they are confused about a feature or want to know if they are using the website correctly.
* Motivation: The product is easy enough for anyone to use irrespective of age or the language they speak. It is almost impossible to make mistakes as the product is straightforward and if a user clicks on a wrong link they can always go back to the previous page by pressing the back button. The website will also have an option to translate all the info into different languages, so it will be easy to use for someone who doesn’t know english.

### Personalization and Internationalization Requirements

* Initially the product will only be available in the US, but as its use extends to other countries, users will be able to select the currency according to the country it is being used in, and the user will also have an option to select a language of their choice. The website and the info on it will be available in most languages used worldwide.
* In the monthly expenses page, users will be able to decide the type of graph (pie chart, histogram etc) they want to view their info in. They will also be able to change colors on the monthly expenses page.

### Learning Requirements

* The product should be as easy to use for a non-tech savvy person as it is for a tech savvy person.
* The user will be able to learn how to use the product properly after just one use.
* Any person, old or young, irrespective of the language they speak and the country they are from should be able to learn how to use the website easily after just one use.

### Understandability and Politeness Requirements

The language used in the product should be complex enough to be understood by people in the industry, but simple enough for anyone else to understand. In other words, the language should include the user’s industry jargon when necessary but be simple, short, and concise everywhere else. Any language used should be as neutral as possible and free of any controversial or divisive terminology, words, phrases, or other expressions.

In terms of the product’s construction, the product will hide all of that information from the users, as they will not need to know or understand how it works to be able to use the product.

### Accessibility Requirements

The product shall be compatible with standard text to speech software, with a special UI generated if needed. There will also be options for changing the font to one that is easier to read for people with dyslexia, as well as color change options for different types of color blindness as well. The product will be compliant with the Americans with Disabilities Act for all other accessibility options and disabilities not listed here.

### User Documentation Requirements

The only documentation that will be supplied with the product will be the EULA, the product will be simple enough that there will be no need to include any major documentation with it.

### Training Requirements

Training will be minimal, as the product will be designed to be as simple as possible. The users are expected to already be busy with their day to day operations, and will not be expected or required to be trained to use the product. At most, the development team will create a basic guide to use the software, though the design should be made as simple as possible to avoid any issues on how to use the product.

## Look and Feel Requirements

### Appearance Requirements

The product will use neutral tones for the color scheme, and will not use anything that would be incredibly bright/obtrusive to the user’s experience. The product will comply with corporate branding standards, and should allow for the user’s corporate logo to be displayed on the user’s personalized homepage.

### Style Requirements

The UI will be as minimalistic and clutter free as possible. The appearance should cause a positive reaction in most of its users. The product should look and feel as simplistic as possible, where most users will feel that the product is easy to use and satisfactory for their needs. However, the minimalist design will not impact any user functionality or usability at all.

## Operational and Environmental Requirements

### Expected Physical Environment

The product will be used in two environments: a kitchen environment and a back office environment. In the kitchen, the product should be loud enough to overcome the noises of a busy kitchen, such as the dishwasher and cooking sounds. In a back office environment, noise generated by the product should be no more than half the noise that should be expected in the environment. All physical requirements will be handled by the device the user operates to access the product, as the product as seen by the user is a website.

### Requirements for Interfacing with Adjacent Systems

The product will work on the last five releases of the ten most popular web browsers. The data content will be a standard web page, with general features and languages attached (JavaScript, PHP, etc.) and the product will be able to handle frequent access from the web browsers, at minimum once every hour. The product will also be capable of operating and communicating with its server(s) and databases up to the previous two versions of their operating systems. Their contents will be the actual website and its features and the data collected from the users respectively.

### Productization Requirements

The product should be easily accessible through any device with a search engine on it. Since the product is a website, all the user needs is a smartphone/tablet/pc, a search engine installed on the device and an account on the website.The user will just have to look up the name of the website on a search engine to be able to open it and use it. Once the user subscribes to the website, they will have to pay a monthly subscription fee in order to continue using it.

### Release Requirements

New functionalities and features will be added to the website regularly.

For example: In the beginning the website will only be functional in the US, and with time it will be made available in more and more countries with the options of more and more languages and currencies. These changes will come with different releases of the website.

## Cultural and Political Requirements

### Cultural Requirements

The product will be culturally neutral if possible, and will attempt to conform to cultural standards as much as possible as the product begins to move outside of the United States and into the global market. As such, the product will keep a record of all public holidays in all areas that the product is available and operating in. Also, the product will not be offensive to any religious or ethnic group.

### Political Requirements

The product will be politically neutral, and will not comment on any political events happening or act as a statement for or against any political statement, unless said statement is to deny someone their human rights, in which case it can be used only to be against said statement. The product is expected to be developed and hosted in the United States. That is not a guarantee, however, and it is expected that the tools and resources used when developing the product will come from local sources.

In terms of corporate politics, the idea is the same, that the product will be politically neutral as much as it can be. That does not mean that the product and/or the development team should not be responding to any potential competition.

## Legal Requirements

### Compliance Requirements

The product will handle and implement personal information in compliance with the Data Protection Act, and handle all accessibility requirements in compliance with the American with Disabilities Act. In regards to all laws on financial information and other unrelated laws, it will be handled as to be compliant with any and all applicable laws. The product will be compliant with all applicable laws within the nations that the product will be operating in as well.

### Standards Requirements

The product will be compliant with all industry standards in regards to server implementation; database maintenance and construction; website implementation, development, maintenance and design; and all standards for all other parts of the product not mentioned above.

# Design

## System Design

### Design goals

The assessment that the user gets must be made to be as straightforward for the user to be able to figure out and reflect what is going on with their business and what is not working for them. Also, the UI for the website should be simple and easy to use for all whether or not they use the internet often to use it all the time. The goal is for the user's request to be met fast and as soon as possible for the marketing team, and the social media team. We want users to be able as soon as they request what they want from the company. The teams immediately are able to begin to plan how to create goals and tasks for users to help with their restaurant. We want the users to feel like a client who is taken care of and also will see results once they implement the task/goals assigned to them.

## Current Software Architecture

Environment: Standard web designer client (specific one left up to development team) for the page, Linux for server environment.

Programming Language: HTML5/CSS bare minimum, Javascript most likely for actual logical programming. SQL/MongoDB distribution for database management, specific distribution left to discretion of the development team.

APIs: Left to the discretion of the development team

## Proposed Software Architecture

### Overview

The code will implement all diagrams and ideas presented in this section with a good degree of faithfulness to them.

### Class Diagrams

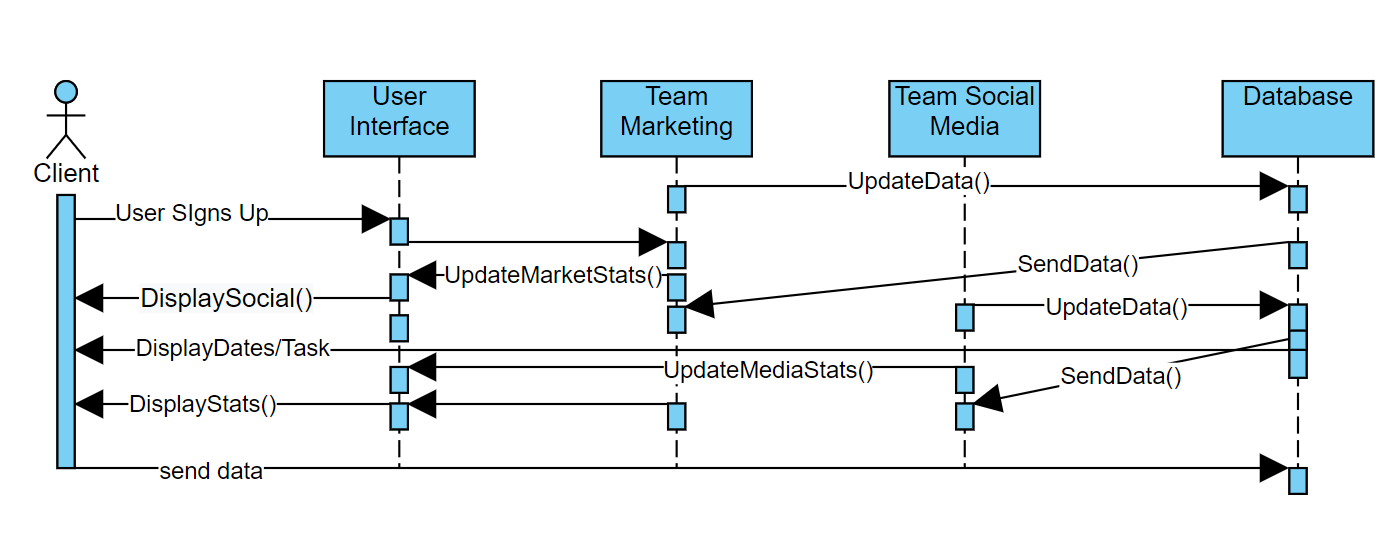
### 

### 

Client will contain a singleton instance. As each client gets declared, they will fill out a form that will describe and narrow down their issues they have with their establishment. Once the company receives the establishment's information, they then come up with a package deal for the user to select. The issues of the client are come up and are assigned to each team depending on what they need to look at. Once the client finishes setting up their account and agrees to the terms and conditions, then each team that is assigned with the client will work together with the client and update the progression on the clients issues. The marketing team will create statistics and graphs from the client's financial information that will be sent to them while the social media team will come up with promotion strategies for them to use. Both will include dates of when to meet up and talk about issues and also to assign tasks for the client to do for their company. The dates will consist of what is due for the client, to meet up if there are concerns or for progressions to see if there is improvement or not.

### Dynamic Model

First, the client will sign for an account and take an assessment to determine what they need help with. Then, the user will select a package they would like. The Teams will both get access to the clients information, and then they will determine ideas and suggestions/tasks for the clients to do and provide guidance for how to take on their tasks/suggestions. The marketing team and social media team will post the statistics they found based on the user’s information. The users profile will contain dates to meet with each team and will update them as needed.

**

### Subsystem Decomposition

Marketing Subsystem:

The Marketing Teams System will use the clients information and data. They will then create/set up a database that will update constantly with the data information. This Subsystem will update the users interface with graphs and charts constantly every so often.

Social Media Subsystem:

The Social Media Team will help create profiles and keep track of the social media engagement the clients establishment gets. This data will be updated on the UI and will show how many followers/likes and how many times the clients establishment gets searched up for they get each week or month depending on the package.

### Hardware / Software Mapping

### 

Database that will contain Client Information and Issues From Assessment. The Logic System prepares the info for each Team/Creates Issues For the Client that needs to be addressed. The Client Info System Takes Clients Info and creates visuals such as statistics and charts/graphs for the client to see. The User Interface Displays the clients basic information, contains dates to be able to assign from the team or for the client to request to meet with the client. Each Team Puts their task and suggestions for the client to see. The graphs shows the clients information and updates their statistics on a daily basis and allows the user to compare numbers from the past as well.

### Data Dictionary

Work\_Order = Invoice\_Number + Service\_Type + Specific\_Requests\_Sent\_By + Service\_Timeline + Service\_Duration + Cost\_Of\_Service + Service\_Provider + Is\_Repeating\_Request

Is\_Repeating\_Request: Boolean value that tells if this is a monthly service or a one time only service being requested in the work order

User\_Acct = User\_Name + User\_Address + User\_Workorder\_History + User\_Financial\_Info

### Persistent Data Management

### The Database will be constantly updated if changes were made. The company's finance team will receive reports from the client either weekly or monthly depending on what the client’s package is.

### Access Control and Security

The host of the servers need to make sure that the clients information is protected and make sure we give access to certain information for different teams. For instance, the Marketing Team can see all finances but the social media team should not. They need to make sure that the client’s info is private.

### Global Software Control

The host of these servers need to make sure that the private information gets received to the right type of people who access it and to make sure that the statistics need to be private as well.

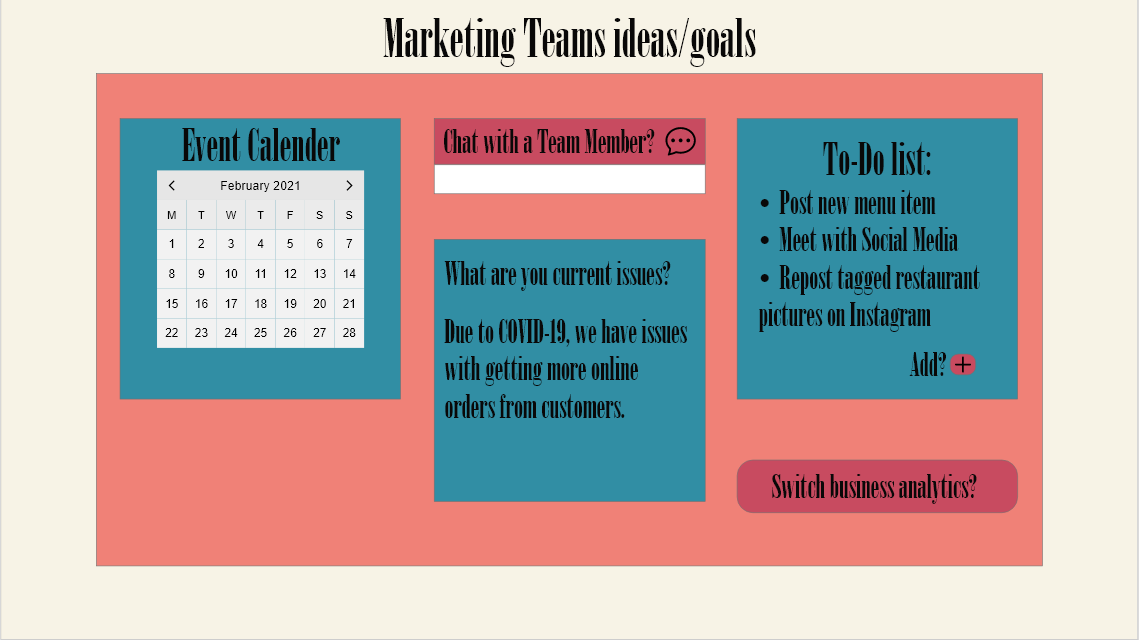
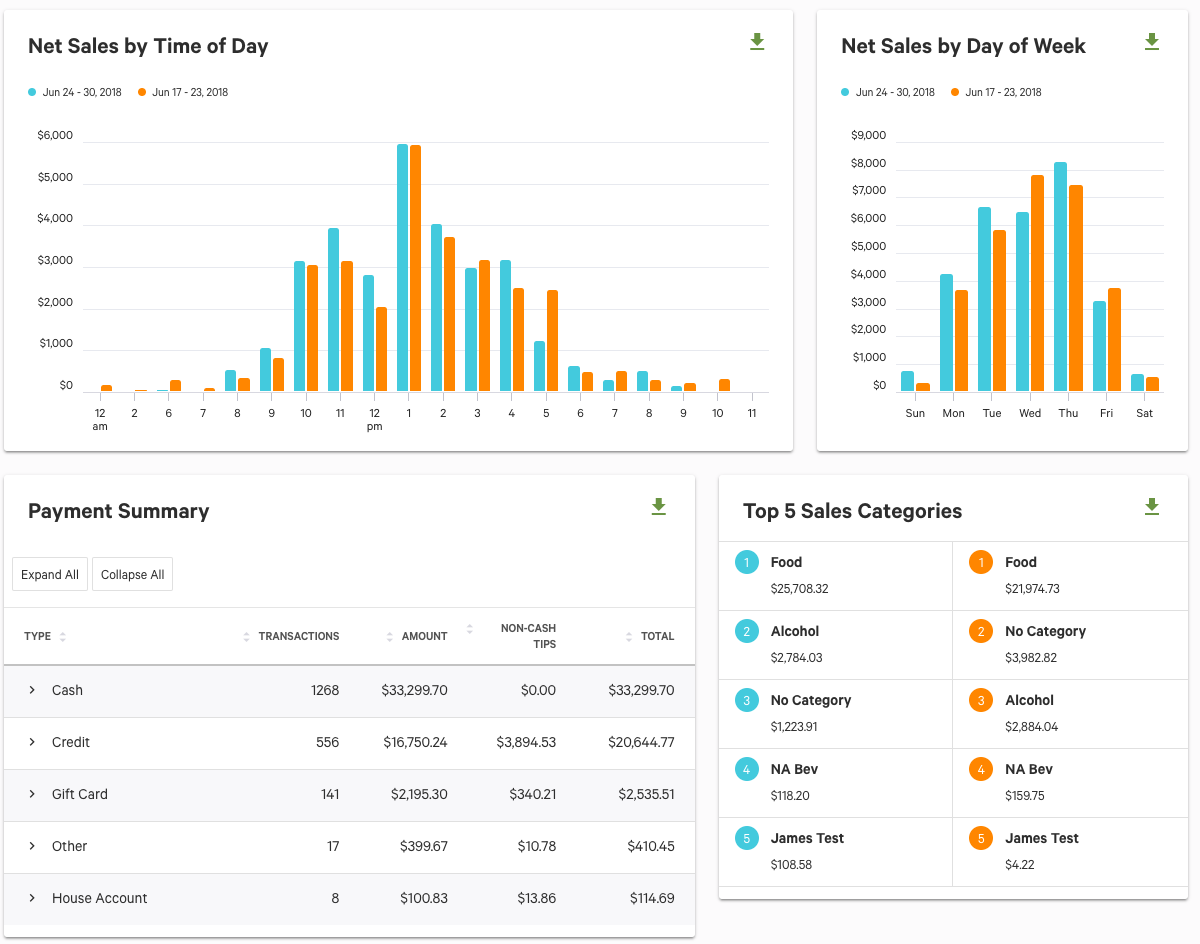
### Boundary Conditions

Each Client Gets a Certain Package, so one needs to make sure that the specified package is what the user is allowed to see. If it is a basic package, they cannot access all the information and data statistics the client can see. .

## Subsystem Services

Data from the Client and what each Team has created for the company gets saved constantly every so often, so that data does not get erased.

## User Interface

Below are the design team’s mockups of the UI. The development team is free to change the designs shown here as they see fit if they are not happy with these. 

## Object Design

### Object Design trade-offs

The trade-offs would be centered around how much content can you put on the UI and how showing the statistics work. It can tend to be tricky to how to constantly show statistics and update them constantly depending on what package the client decides. Also, the servers can be a lot to handle if there is a lot of clients since it would be teams working with clients and a server to host the client sides and the teams who work with the client's side. It can be tricky to control and protect what information gets shown or not.

### Interface Documentation guidelines

All interfaces should be documented in such a way that they can be easily understood by anyone looking at the interface’s code directly and be simple enough where the documentation will take no more than two pages to be written out fully.

### Packages

It is assumed that the development team will choose the packages necessary to complete the product as they see fit, and that they will choose the packages based on the requirements listed in this document. At minimum, the packages will follow all design and functional requirements listed above and below, preferably without having to do major modifications to the packages.

### Class Interfaces

The inferences should reference the clients information and what entails about the issues and concerns they have. The way it should be set up is that each instance of the interface will have the clients info and what issues they need to address per company section that will help them with that. For instance, the marketing team has its own class and ways to address it, so that the interface can receive the customers issues.

# Test Plans

## Features to be tested / not to be tested.

* Account Functions
  + Account Creation
  + Account Recovery
* Creating a Work Order
  + Work Orders for One Service Once
  + Work Orders for One Service Multiple Times
  + Work Orders for Multiple Services Once Each
  + Work Orders for Multiple Services Multiple Times
* Customizing a User’s Home Page
* Account Security
* Financial Security
* Processing/Confirming a Work Order
  + Same Sub-Tests as Creating Work Orders
* Sending/Receiving Messages Through Product
* Handling/Saving/Modifying User Information
* UI Stability
* Processing Speed

## Pass/Fail Criteria

* + 1. **Account Functions**
       1. **Account Creation**: User email id and password should be authenticated and once a user registers successfully, their information should be added to the user database. Inability to do so results in failure of the test.
       2. **Account Recovery**: The test passes if the user is able to recover their account after successfully registering with the service. Otherwise, the test fails.
    2. **Creating a Work Order**
       1. **Work Orders for One Service Once**: The test passes if the user is able to create a work order for any service of their choice, but only for one instance. Otherwise, the test fails.
       2. **Work Orders for One Service Multiple Times**: The test passes if the user is able to create a work order for any service of their choice for multiple instances. Otherwise, the test fails.
       3. **Work Orders for Multiple Services Once Each**: The test passes if the user is able to create multiple work orders for any services of their choice, but only for one instance. Otherwise, the test fails.
       4. **Work Orders for Multiple Services Multiple Times**: The test passes if the user is able to create multiple work orders for any services of their choice, for multiple instances. Otherwise, the test fails.
    3. **Customizing a User’s Home Page**: The test passes if the user is able to customize their home page with respect to currency, language and type of graphs for monthly expense information. Otherwise, the test fails.
    4. **Account Security**: The test passes if only the developers have access to user information like user’s name, address, email id, password. Otherwise, the test fails.
    5. **Financial Security**: The test passes if only the developers have access to a customer’s financial information. Otherwise, the test fails.
    6. **Processing/Confirming a Work Order**
       1. Same Pass/Fail criteria as Creating Work Orders
    7. **Sending/Receiving Messages Through Product**: The test passes if the users and customer service reps are able to send and receive messages. Otherwise, the test fails.
    8. **Handling/Saving/Modifying User Information:** The test passes if user information present in the database can be edited/modified. Otherwise, the test fails.
    9. **UI Stability:** The test passes if the UI is stable and does not crash. Otherwise, it fails.
    10. **Processing Speed:** The test passes if the website sends and receives data, and performs other functions in a desired timeframe. Otherwise, it fails.

## Approach

The testing approach will be conducted as such:

Whenever a function or feature of the product is finished up or an update to one has been pushed, the development team will begin the testing on that function. When all testing is finished up and all tests have passed, the team will finalize the changes and begin work on adding in/updating a function. If any test fails during the testing phase of the new/updated function, the team will handle the problem as detailed in section D.

At the end of the working day, when all team members have finished up with their work on the product and have pushed their code to the repository, the head of the team will begin an end-of-day automated testing cycle, where all new features are to be run with a predefined testing schedule to see if they may have caused any functions or code to break by being included in the product. Once this is done, the head will report to the rest of the team if the tests passed or failed, and then relay what needs to be done the next day.

## Suspension and resumption

* + 1. Account Functions
       1. Account Creation:

**Suspension**: Further testing should be suspended if the user id and password cannot be authenticated during account creation, or if the account details are not being added to the database.

**Resumption**: Testing can be resumed if userid/password can be authenticated and account details are being successfully added to the database.

* + - 1. Account Recovery

**Suspension**: Further testing should be suspended if an already registered email cannot be extracted from the database.

**Resumption**: Testing can be resumed if the information of an already registered user can be extracted from the database successfully.

* + 1. Creating a Work Order

**Suspension**: Further testing should be suspended if a work order cannot be successfully added to a particular account, for the requested number of times and services.

**Resumption**: Testing can be resumed if a work order can be successfully added to a particular account, for the requested number of times and services.

* + 1. Customizing a User’s Home Page

**Suspension**: Further testing should be suspended if a homepage cannot be customized to change the language, currency, or the type of graph to show monthly expenses.

**Resumption**: Testing can be resumed if a homepage can be customized to change the language, currency, or the type of graph to show monthly expenses according to the user’s choice.

* + 1. Account Security

**Suspension**: Further testing should be suspended if account information can be viewed by anyone other than the developers.

**Resumption**: Testing can be resumed if only the developers are able to view account information.

* + 1. Financial Security

**Suspension**: Further testing should be suspended if financial information can be viewed by anyone other than the developers.

**Resumption**: Testing can be resumed if only the developers are able to view financial information.

* + 1. Processing/Confirming a Work Order
       1. Same Suspension and Resumption criteria as Creating Work Orders
    2. Sending/Receiving Messages Through Product

**Suspension**: Further testing should be suspended if messages cannot be sent and received between customers and customer service reps in a timely manner.

**Resumption**: Testing can be resumed if messages can be successfully sent and received between customers and customer service reps in a timely manner.

* + 1. Handling/Saving/Modifying User Information

**Suspension**: Further testing should be suspended if user information cannot be successfully modified and if the changes do not reflect in the database in real time.

**Resumption**: Testing can be resumed if user information can be successfully modified and if the changes reflect in the database in real time.

* + 1. UI Stability

**Suspension**: Further testing should be suspended if the UI leads to a lot of crashes while using the product.

**Resumption**: Testing can be resumed if the UI works flawlessly while using the product.

* + 1. Processing Speed

**Suspension**: Further testing should be suspended if the processing speed for the product is over the desired timeframe.

**Resumption**: Testing can be resumed if the processing speed for different instructions goes below the desired timeframe.

## Testing materials ( hardware / software requirements )

1. A device with a search engine and a working internet connection is required. Can be a smartphone, tablet or a PC.
2. The product must be able to interact with databases in order to store customer information, service information etc.
3. The product should be able to interface with adjacent systems used in the overall development of the product.

## Test cases

* + 1. Account Functions
       1. Account Creation

**Test Case**: Results should be checked and added to the information database when valid user id, password and user information are entered.

* + - 1. Account Recovery

**Test Case**: Check if valid user id is entered, if yes, the user should be able to recover their account by setting a new password.

* + 1. Creating a Work Order
       1. Work Orders for One Service Once

**Test Case**: Check if a service/multiple services have been requested and if it is for a single instance or a multiple instances.

* + - 1. Work Orders for One Service Multiple Times

**Test Case**:Check if a service/multiple services have been requested and if it is for a single instance or a multiple instances.

* + - 1. Work Orders for Multiple Services Once Each

**Test Case**:Check if a service/multiple services have been requested and if it is for a single instance or a multiple instances.

* + - 1. Work Orders for Multiple Services Multiple Times

**Test Case**:Check if a service/multiple services have been requested and if it is for a single instance or a multiple instances.

* + 1. Customizing a User’s Home Page

**Test Case**: Check if the homepage can be successfully modified by selecting different languages, currencies, graphs etc.

* + 1. Account Security

**Test Case**: Check if the account information saved in the database can be accessed through the website, or by anyone other than the developers.

* + 1. Financial Security

**Test Case**: Check if the financial information saved in the database can be accessed through the website, or by anyone other than the developers.

* + 1. Processing/Confirming a Work Order
       1. Same Sub-Tests as Creating Work Orders
    2. Sending/Receiving Messages Through Product

**Test Case**: Check if messages can be sent and received in a desired timeframe by a user account to service reps and vice-versa.

* + 1. Handling/Saving/Modifying User Information

**Test Case**: Check if user information can be changed on the website and if the changes reflect in the database in real time.

* + 1. UI Stability

**Test Case**: Check if the UI for the website is functioning and stable, and does not lead to a lot of freezes or crashes.

* + 1. Processing Speed

**Test Case**: Check if the website sends and receives data in a desired timeframe.

## Testing schedule

* Testing should be repeated several times throughout the development process of the website. Some tests need to be repeated in a sequence after other tests are passed to check the functionality of the product to a proper extent.
* User authentication and account recovery should be the first features to be tested as the product can only be used if the user has an account. Account security and financial security should also be tested in the beginning.
* Customization and modification features should be tested next.
* Finally, all features related to work orders and services provided should be tested all together.

# Project Issues

## Open Issues

As the American public becomes increasingly conscious of their privacy online and their data, and especially with the European Union passing its own internet privacy regulations and laws, it will be expected that the United States government will follow suit in the near future. Whether these rules and regulations will be as intensive as the EU’s regulations, less stringent, or even more intensive and comprehensive remains to be seen.

There has also been a development in California in regards to the employment status of “gig workers”, specifically with Uber and Lyft drivers [4]. If food delivery is to be implemented into the product, then this will need to be acknowledged and accounted for. It is unknown whether or not other states will follow California’s example and rule the same way, but the team and product will be able to adapt to any one of these possibilities happening.

## Off-the-Shelf Solutions

As the product will be innovating in the restaurant industry and less so in the tech industry, the development team should not have to worry as much about creating and implementing new technologies, and will instead be able to use technologies already available to the public.

### Ready-Made Products

In regards to the database portion of the product, there are already multiple different designs and distributions of databases, such as MongoDB and the various SQL forks, that there will be no need to “reinvent the wheel” when it comes to managing data. When it comes to payment processing, if handling payment is too difficult of a task to do/ too time consuming, then implementing a service like VISA or PayPal would suffice easily.

Also, all hardware components will not need to be developed in-house, as there will not need to be any “innovation” with that. The server and hosting can be bought from separate vendors instead.

The product will not use a service like WordPress for its design and implementation, as that would make the site less unique than its competitors and make the team dependent on help and support from WordPress itself.

### Reusable Components

In terms of libraries and toolkits, standard web development tools and libraries like the various JavaScript libraries will be used to develop the product if needed. It will be expected that there will not be any issue with using these for implementing the product fully.

It will be expected that the server will run a Linux-based operating system, either one that can be acquired without cost, such as a Debian distribution, or one that must be paid for, such as a Redhat or OpenSUSE. Support for the server, at least software side, will be coming from the toolkits provided by the server operating system.

### Products That Can Be Copied

The team should be able to use a service like Google Firebase to handle authentication and account recovery for user accounts as well as for more generalized database applications. This would save at least a week or so of development time, from not having to set up an account system to standard authentication being easy to handle. If there are any other authentication and/or account systems that exist, they can be just as viable as the one listed in this paragraph.

## New Problems

### Effects on the Current Environment

There is the possibility, however slight, that the product will affect the employment of individuals who would otherwise be doing the work that the product will be doing itself, but this is expected to be an incredibly small number of people at maximum. The product will create a higher availability of marketing services to restaurants, which in turn is expected to increase the demand for said services. If anything, the product may end up creating jobs rather than remove them.

### Effects on the Installed Systems

The product is expected to have minimal effects on the systems it will run on, but the development team will take care to make sure that there will not be any issue with users running the system, such as some users still using Internet Explorer.

There is a chance of the general public becoming more accepting of cryptocurrency and may request to pay their bill with it. In the event that this becomes a major request from all users of the product, the development team should begin drafting plans for implementing payment this way.

### Potential User Problems

The product is expected to not conflict with the user at all, but there may be a possibility the product may generate an adverse reaction either from the politically-neutral stance of the product or its color scheme. In the event that the reaction is politics related, the development team is expected to ignore it and not do anything. If it’s the color scheme, the team will log the complaint. If there are enough complaints, as to be determined by the development team, then a change in color scheme will be planned for and implemented in the future.

### Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Currently, there are no known physical limitations to implementing the product.

### Follow-Up Problems

The major situation that could spell disastrous for the development team and the product is as if the product grows too big too quickly. If there is not enough bandwidth and server space to accommodate large influx of users onto the product, this will lead to slowdowns, decreased performance, and complete denial of service in the worst case scenario.

Otherwise, the only other major problem that could hit the product will most likely be a natural disaster, such as an earthquake or tornado. The team should have at least one backup of the product available in case there is a disaster that takes out the original product, but as for how they will be able to handle the destruction of the physical components is currently unknown.

## Migration to the New Product

Not Applicable

## Risks

The most likely risk that will impact the project is creeping user requirements. As the product seeks to not only encompass marketing for the restaurant industry but also for services like food delivery, the users will be asking more and more for new requirements and features to be added to the product. If this backlog grows large enough, the product will experience feature creep, which is a massive problem for any software development team. This, in turn, will lead to the team experiencing more problems, like burnout and low productivity. Getting out of that feeling is near impossible for a team once they start to feel it.

The team should also take care to ensure their metrics and measurements to measure success and growth with the product are adequate and accurate. If these items are not both, then getting a good reading on what is working versus what is not will be impossible, let alone how well they are working.

## Costs

Assuming that each input/output flow would cost $100 to implement, business events would not cost anything (as it is expected that the team will not go to these), each use case is roughly $75, each requirement, both functional and nonfunctional, costs $50, and that each constraint would cost $50 to meet, it is expected that this product will cost around $1,850. There would be an estimated four input/output flows, six use cases, ten functional and nonfunctional requirements, and ten constraints to handle. This is an estimated cost, however, and the development team may find themselves having to spend more or less depending on the circumstances involved and products/tools available.

As in the case of time, it is expected that each team member will spend at least one and a half hours on developing this product daily, if not every other day.

## Waiting Room

General ideas to consider in the next release of the product and not the current release, ordered from most likely/possible to least:

**Delivery Service:** The ability for the product to arrange delivery of foods from client restaurants to their customers either through an in-house service or through a separate company that the product put into contact with the client.

**Non-Online Advertising:** Advertising via traditional media and methods, such as television/radio commercials, billboards, and/or newspaper advertisements.

**In House Payment Processing:** Handling payments in house similar to how PayPal would handle payments. Consider this goal a lofty one that should be planned out first at a much later date than the first official release.

## Ideas for Solutions

In regards to bill payment and processing, until the product actually has payment processing natively made for it, the product should handle payment via third parties like VISA and PayPal.

In terms of server design and implementation, the team should be able to use a Linux Server distribution rather than a paid server operating system like Windows. This will help keep costs down and usability high.

Security is incredibly important for the product, so if there is a product/service that can handle the heavy lifting in regards to security and maintaining the security requirements, the development team will adopt it as soon as possible, assuming that it can fit into the budget itself.

## Project Retrospective

Doing much of the work during project meetings and in short bursts did not work out as well as anticipated, as it only led to stress and worry on the part of everyone in the group. Keeping in contact via messaging clients like Discord really did help however, especially considering the CoViD-19 pandemic currently still happening as of the writing of this document. It helped to make up for not being able to meet together in person once this semester.

The major problems that we, the design team, had was simply getting the time to get together and handle major developments on this document. It would have been a good idea to schedule out the meetings with the group altogether rather than having the meetings be as spontaneous as they were. That way, there would have been less stress over when the meetings were going to happen and be able to plan around the meetings as necessary

# Glossary

**Customer**: The representative of the restaurant, either the owner of the establishment or its manager, buying the work order from the firm/contractor.

**Work Order**: Order from a customer to a firm/contractor specifying what service they are requesting, what special requests the customer wishes to have fulfilled, the duration that the service will take place in, and the specific type of service requested (Digital Marketing Campaign, Newspaper/Local Radio Ads, Local Delivery Service Contracts, etc.)

**Local**: Referring to businesses, resources, and services within the operating area of the customer.

**Firm/Contractor**: referring to the marketing firms, delivery service companies, and other relevant service companies that offer their services through the site.

**Product**: in reference to the project itself, which will operate as a website. Can be used interchangeably with “website” when referring to the project.

**Service List**: The directory of services provided by the firms/contractors operating on the site itself. The customer will be shown this after they input a search request to find the work order that best fits their needs.

**Client**: Refers to both the customers and the firms/contractors operating the product.

**Gig Worker**: Person that works in an informal, non-contract setting/job environment, such as a taxi service like Uber/Lyft or on a website like Fiverr.

**MongoDB**: Document oriented database system designed for modern systems[5].

# References / Bibliography

|  |  |
| --- | --- |
| [1] | Robertson and Robertson, Mastering the Requirements Process. |
| [2] | A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013. |
|  |  |
| [3] | M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.  [4] Allyn, Bobby. “California Judge Orders Uber And Lyft To Consider All Drivers Employees.” *NPR*, NPR, 10 Aug. 2020, [www.npr.org/2020/08/10/901099643/california-judge-orders-uber-and-lyft-to-consider-all-drivers-employees](http://www.npr.org/2020/08/10/901099643/california-judge-orders-uber-and-lyft-to-consider-all-drivers-employees).  [5] “The Most Popular Database for Modern Apps.” *MongoDB*, [www.mongodb.com/](http://www.mongodb.com/).  [6] Valinsky, Jordan. “10,000 Of America's Restaurants Have Closed in the Past Three Months.” *CNN*, Cable News Network, 9 Dec. 2020, [www.cnn.com/2020/12/08/business/restaurant-closures-coronavirus/index.html](http://www.cnn.com/2020/12/08/business/restaurant-closures-coronavirus/index.html).  [7] Resendes, Stephanie. “The Average Restaurant Profit Margin and How to Increase Yours.” *Restaurant Insider*, Restaurant Insider, 6 Apr. 2021, upserve.com/restaurant-insider/profit-margins/.  [8] Ozimek, Adam. “No, Most Restaurants Don't Fail In The First Year.” *Forbes*, Forbes Magazine, 3 Feb. 2017, [www.forbes.com/sites/modeledbehavior/2017/01/29/no-most-restaurants-dont-fail-in-the-first-year/?sh=537e978c4fcc](http://www.forbes.com/sites/modeledbehavior/2017/01/29/no-most-restaurants-dont-fail-in-the-first-year/?sh=537e978c4fcc).  [9] Lichtenstein, Noah. “The Hidden Cost of Food Delivery.” *TechCrunch*, TechCrunch, 16 Mar. 2020, techcrunch.com/2020/03/16/the-hidden-cost-of-food-delivery/. |

# Index

Design 61, 63

Requirements 35, 51, 58

Test 64, 65