Yuri Khechoyan, Steven Olsen, Jessie Wilkins, John McQuaide

FALL 2017

CIS 330 Project Proposal:

Database Management Final Project Proposal – 9/2/17

**Team Members:** Yuri Khechoyan, Steven Olsen, Jessie Wilkins, and John McQuaide

**Project Title:** QUME | qu·eu·ed (working titles)

**Business Requirements:**

**Description**:

Create a Customer Queue platform that helps identify who has not been assisted yet. While also enabling the business to cut their costs, eliminate unnecessary equipment, and create a level of interactivity between the customer and business that has not been achieved yet.

**Why/Importance**:

We chose this topic for numerous reasons. Before we can identify ***what*** we need to do and how we need to do it, we first need to identify the ***WHY***.

**1)** This software is applicable to a variety of different businesses that interact with customers every day.

**2)** The business does not need to buy, maintain, and replace supplementary hardware (example: Bread Co pagers – which are made by a 3rd party company, not Bread Co).

**3)** Since this is software based, it is completely customizable to the heart’s content of the customer (If they are in the customer service realm of a company, the program could ask for the reason of their customers’ visit [Account issues, New Customer that wants to sign up with the company, Billing/Payment, etc.]. If the software would be for Restaurants, it could ask for the size of the party instead of what the reason for their visit was.

**4)** From a business standpoint, the program could also keep a record of each customer and their information [the name and number of the customer, why they came in that day/party size, and a timestamp as to when they came in and left]. That last point would be extremely critical and invaluable to the business because it would be able let management of that business know how long each customer is waiting in the queue so that they could act upon it and figure out how they could cut down the wait time to service more customers per work-day.

**Users:**

Users will consist of the establishment Employees/Management, the customers of that establishment, and Developers (Development purposes)

**Plan of Action:**

Plan of action can be seen below with Use Case Diagrams, User Stories, and High-Level Diagrams

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topics** | **Yuri** | **Steven** | **Jessie** | **John** |
| **Meetings** **(Virtual or In Person)** | Weekly | Weekly | Weekly | Weekly |
| **Main Method of Communication** | Slack | Slack | Slack | Slack |
| **Repository of Choice** | GitHub | GitHub | GitHub | GitHub |
| **Strengths** | Leadership  Organization  Time Mgmt.  UI/UX Design Development | -SQL  -Algorithm Development  -GUI Design | Development  Db. Mgmt.  Communication | Web Dev  Time Mgmt.  Mathematics |
| **Roles** | Project Lead  Project Manager SW Engineer  Systems Admin UI/UX Developer | Worker/Programmer | Db. Manager | Data Analysis |

**Feasibility Analysis:**

**Technical:** From a technical standpoint, this is do-able within the given semester timeframe

**Financial:** Not Applicable. Will not meet with clientele to gather functional/non-functional requirements at this time

**External Entities**: Twilio for sending the SMS messages to update customers, Stripe or Braintree to handle all of the transactions that will be made to Developers

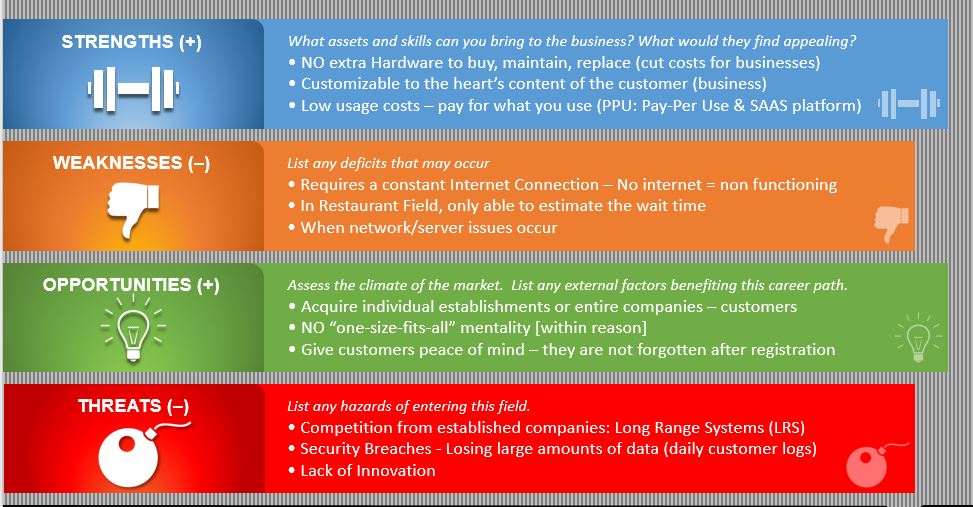
**Legal**: Any and all propriety software will need to be submitted to, filed with and Granted by the US Patent and Trademark Office (USPTO)

**Schedule**: We will only be able to develop this system within the allotted timeframe of one (1) semester.

**Resources**: Resources that will be needed in order to complete this project are: Computers with an IDE, XAMPP, Twilio API with Dependencies, Apache POI API to export information to an Excel sheet (usage may vary), and Photoshop

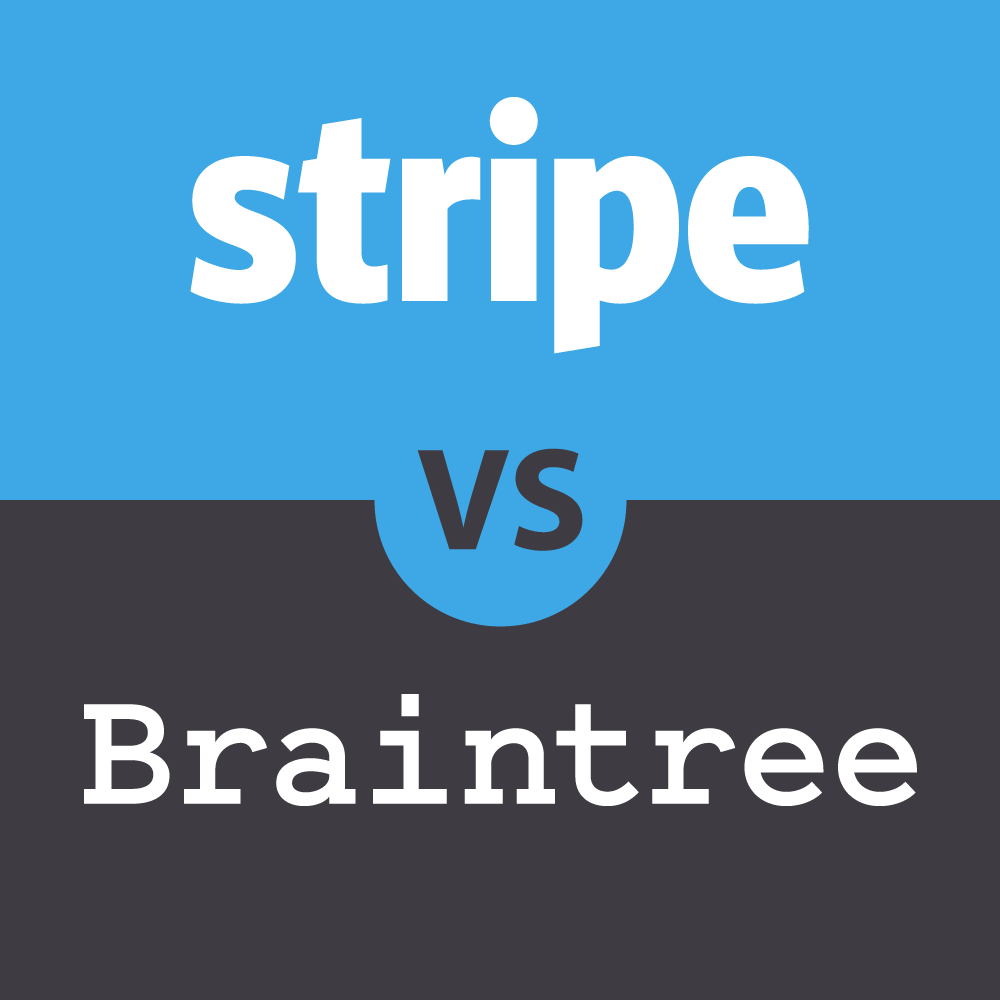
**Market Research**: The largest company that provides queue based systems is Long Range Systems (LRS). Businesses that use this system must buy, maintain, and pay for any repairs done on LRS’s equipment. And depending on the scale of the business, the cost to buy and repair these systems can get expensive.

**SWOT Analysis:**

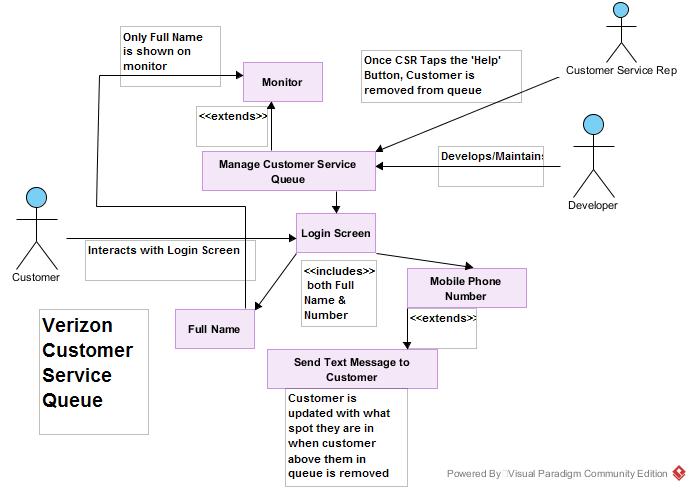


**Payment Model:**

* Implementing Payment System for every customer registered through system
  + Establishment will pay Developers or Company ~ $2-$50 for every customer registered (Price will be set by developer based on average daily customer volume)
    - Ex. 1: 300 visitors/customers @ $2 = $600/day from 1 establishment
    - Ex. 2: 300 visitors/customers @ $50 = $15,000/day from 1 establishment
    - Payments would be made through 3rd Party API



**Use Cases:**



|  |  |
| --- | --- |
| **UC1 Manage Customer Service Queue** | |
| **Scope** | Business |
| **Level** | Sea Level |
| **Goal in Context** | Allows the Developer to access/modify various Customer Service Queue functions. |
| **Primary Actor** | Owner/Developer |
| **Stakeholders** | Developer, Customers & Customer Service Representatives |
| **Preconditions** | Run Customer Service Queue software |
| **Triggers** | Program is launched at the beginning of the work day |
| **Success Guarantee** | The Customer Service Queue waits for user input |
| **Main Success Scenario:** | |
| Customer Service Queue: There are 2 options to choose from:  1. Enter Full Name  2. Enter Mobile Phone Number | |
| **Extensions:** | |
| None | |

|  |  |
| --- | --- |
| **UC2 Enter Full Name** | |
| **Scope** | System |
| **Level** | Sea Level |
| **Goal in Context** | Present a form to the user to enter their Full Name Credential in order to move forward |
| **Primary Actor** | Customer |
| **Stakeholders** | Developer, Customer, Customer Service Representative |
| **Preconditions** | Program must be at the login screen |
| **Triggers** | Customer taps or clicks on the text field to enter their name. Virtual keyboard is initiated or actor uses mechanical keyboard |
| **Success Guarantee** | Customer taps or clicks on the "Done" button to enter their name and move on to the next part – On computer: actor moves onto mobile number |
| **Main Success Scenario:** | |
| 1. Customer: taps on the Full Name text field.  2. Customer Queue: Initiates a virtual on-screen keyboard or uses mechanical for the customer to enter their Full Name  3. Customer taps on the "Done" button when they are finished typing in the name and want to move on to the next part. | |
| **Extensions:** | |
| **3a**: If there is no name in the field or if the Customer provided the first name only, the program will throw a message alert box asking them to include all required information. Same thing happens if they only enter their last name. | |

|  |  |
| --- | --- |
| **UC3 Enter Mobile Phone Number** | |
| **Scope** | System |
| **Level** | Sea Level |
| **Goal in Context** | Present a form for the customer to enter their phone number |
| **Primary Actor** | Customer |
| **Stakeholders** | Developer, Customer & Employee |
| **Preconditions** | Program must be at the login screen |
| **Triggers** | Customer taps on the text field to enter their number. Virtual keypad is initiated |
| **Success Guarantee** | Customer taps on the "Done" button to enter their name and move on to the next part |
| **Main Success Scenario:** | |
| 1. Customer: taps on the Mobile Number text field.  2. Customer Service Queue: Initiates a virtual on-screen keypad for the customer to enter their Mobile Number  3. Customer taps on the "Done" button when they are finished typing in their number and want to move on to the next task. | |
| **Extensions:** | |
| **3a.** If there is no number in the field, if the Customer did not provide the full 10-digit number, or Customer has entered a fake '555' number, the program will throw a message alert box asking them to include all required information. | |

|  |  |
| --- | --- |
| **UC4 Submit Button** | |
| **Scope** | System |
| **Level** | Sea Level |
| **Goal in Context** | Once all required credentials are met, customer can submit this information into the queue. |
| **Primary Actor** | Customer |
| **Stakeholders** | Developer, Customer & Employees |
| **Preconditions** | The Customer must enter all of the required credentials in order for the submission to be processed |
| **Triggers** | Customer must tap the "Submit" button after all field are filled in |
| **Success Guarantee** | A message alert pops up to say "Thank You for Registering" and then the program goes back to the login screen. Customer that just registered receives a text message **"You are now X in line. Thank You for being patient with us!"** |
| **Main Success Scenario:** | |
| 1. Customer: Receives a text confirmation | |
| **Extensions:** | |
| **1a.** In order for this information to be submitted, the Customer must tap on the "Submit" button. | |

|  |  |
| --- | --- |
| **UC5 Removal from Queue** | |
| **Scope** | System |
| **Level** | Sea Level |
| **Goal in Context** | Allows the Employee to remove customers from the list. |
| **Primary Actor** | Employee |
| **Stakeholders** | Developer, Customer & Customer Service Representative |
| **Preconditions** | Customers have already added themselves to the queue and are awaiting assistance |
| **Triggers** | Employee taps on the 'Delete' button |
| **Success Guarantee** | Customer at the top of the list is removed from the queue. Customer that was first in line receives this text: **"You are 1st in line, please wait for your name to be called. Thank You."**  Also, all other customers receive this text message simultaneously **"You are now X-1 in line. Thank You for being patient with us!"** The 'X-1' is substituted with the spot that they hold in the queue. |
| **Main Success Scenario:** | |
| 1. Customer: (if 1st in the queue) is removed from the queue on the monitor in the store. If not 1st, they are automatically moved up 1 spot | |
| **Extensions:** | |
| **1a.** Customer that was removed receives: **"You are 1st in line, please wait for your name to be called. Thank You."**  **2a.** Customer that was not 1st in the queue receives: **"You are now X-1 in line. Thank you for being patient with us!"** | |

**High Level Design:**

