|  |  |
| --- | --- |
| **Elicit and Document Requirements** | |
| Problem Statement | Can we design an application that removes the need for physical queues in stores? |
| Problem | Reduce the need for queues at physical locations |
| Target Users | All F&B establishments and retail stores throughout Singapore, android mobile users |

**Functional Requirements**

* 1. **Registration**

  1.1.1.  The end-user and Service-Provider must both have their own account creation process.

        1.1.1.1.  The end-user and Service-Provider must be able to sign up via Google.

  1.1.2.  The system must be able to validate all fields that have been filled up.

      1.1.2.1.  The system must validate that the password and confirm password fields are filled up.

      1.1.2.2.  The system must validate that the user’s username fields are filled up.

      1.1.2.3.  The system must validate that the user’s email fields are filled up.

      1.1.2.1.  If validation fails, the corresponding error must be displayed for the missing field.

  1.1.3.  The system must be able to check the validity of the fields.

      1.1.3.1.  The system must validate that the password and confirm password are the same.

      1.1.3.2.  The system must validate that the user’s username fields do not exist.

      1.1.3.3.  The system must validate that the user’s email fields do not exist.

      1.1.3.1.  The system must validate the verification code matches.

      1.1.3.5.  If validation fails, the corresponding error must be displayed for the missing field.

  1.1.1.  The users should be able to access the registration page from the login page.

**1.2 Login**

  1.2.1. The system must provide a login entry for all users.

     1.2.1.1. The system must provide an email address field for the user.

     1.2.1.2. The system must provide a password field for the user.

  1.2.2. The system must be able to validate the credentials entered by the users.

     1.2.2.1. The system must validate that the email address field is filled.

     1.2.2.2. The system must validate that the password field is filled.

      1.2.2.3. If any of the fields are not filled, a corresponding error message will be displayed for the empty field.

  1.2.5. A successful login shall redirect the different users to their respective homepages.

**1.3 Home/Main Page (End-User)**

  1.3.1. An End-User must be able to view the wait/queue times for shops, eateries, and attractions.

     1.3.1.1.  The system shall show the wait times of establishments listed.

     1.3.1.2.  An End-User should be able to filter/view the wait times by category.

     1.3.1.3.  An End-User should be able to do a quick search for a particular store.

     1.3.1.1.  An End-User should be able to chat with various users in that store

  1.3.2.  An End-User must be able to queue for an establishment using the app.

     1.3.2.1.  An End-User must link their SingPass to join the queue (Theoretical).

     1.3.2.2.  An End-User must be able to obtain a queue number via the app.

  1.3.2.3. Prompts/notifications must be shown when approaching one’s turn/missed queue number (notification system).

    1.3.2.1.  The system must provide an option for the user to rejoin the queue in the event of a missed queue number.

     1.3.2.5.  The End-User should be able to checkout of the store.

     1.3.2.2.  The End-User should be able to leave the queue in the system at any point of time.

  1.3.3.  The system should be able to store the End-Users favourites section.

    1.3.3.1. The End-User should be able to add stores to their favourites list.

    1.3.3.2 The End User should be able to view their favourites list

  1.3.1.  The system should provide a recommended list of stores to the users.

**1.4 Home/Main Page (Service-Provider)**

1.1.1.  A Service-Provider must be able to view and manage its own service queue.

      1.1.1.1.  A service queue should be set up for each provider.

      1.1.1.2.  The service queue does not require any inputs from the provider.

      1.1.1.3.  The Service-Provider must be able to manually insert and remove users in the queue.

      1.1.1.1.  The Service-Provider must be able to notify End-Users in the queue.

   1.1.1.5 Service-Provider home screen should display summary information on the store’s queue

      1.1.1.6 Service-Provider should be able to get detailed information on the queue

**1.5 Feedback**

  1.5.1.  An End-User must be able to report issues and submit feedback.

  1.5.2.  End-User should be able to provide feedback on their experience regarding app usage

**1.6 Rewards**

  1.2.1. End-User can earn points only if they check out of the store.

  1.2.2.  The system must allow users to view the points they have.

  1.2.3.  The system must allow users to view claimable rewards.

     1.2.3.1.  The system should allow users to exchange points for rewards only if they have enough points.

# 2.  Non-Functional Requirements

**Speed**

1. The system must update the crowds every 10 min
2. Display crowd/waiting time information to the user within 10s

**Ease of Use**

1. An end-user shall be able to use all system functions within 10 minutes of first use.
2. An end-user shall be able to join a service queue within 2 screen transitions from the home screen
3. Reaching an endpoint in the system must require a minimised number of clicks (3 - 4)
4. All features should be reachable within 5 actions
5. Once an end-user has joined a service queue, the current queue status of that service queue shall be displayed on the home screen
6. Queue number must be shown on the homepage for existing users checking their queue
7. Scalability - show places previously visited/liked

**Reliability**

1. There shall be 0 incidents of synchronicity of end-user status across different devices
2. There shall be 0 incidents of synchronicity between the service queue status displayed to end-users and Service-Providers
3. Penalty system for false queue
4. Allow users to only queue when they are sufficiently near the area by tracking their location from the store they are trying to join

**Robustness**

1. In the event of an unexpected system failure, the system shall be restarted within 1 minute
2. There shall be at most 1% of unique scenarios that result in a system failure in a month
3. There shall be at most 1% of system failures that result in data corruption in a month

**Portability**

1. An end-user shall be able to join the end of the queue using only the system

**Usability**

1. The application should have Multiple Language support
2. An end-user shall have the option of enabling push notifications to be notified when they are nearing the front of the service queue

### **3. Data Dictionary: General**

|  |  |
| --- | --- |
| API | Application Programming Interface. |
| Authentication | The process of verifying if a user is registered in the system. |
| Check Out | The action of checking out is done by the user and is the act of the user leaving the store/attraction/location |
| End-User | A customer or potential customer at a specific establishment using the system. |
| Feedback | Feedback is what users can provide regarding the APPLICATION via the application feedback page. They may also report any issues with the app through the reports page |
| Google Services | An authentication API |
| Join Queue | The action of getting added/entering a ServiceQueue of a specific Service-Provider |
| Leave Queue | The action of getting removed/exiting a ServiceQueue of a specific Service-Provider |
| MongoDB | MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas |
| Queue Status | Contains the following information about the service queue: Number of end-users in ServiceQueue, Estimated wait time |
| Rating | What users can provide regarding their experience with a particular store/attraction/location. It is given via the app after the user has successfully checked out of the store |
| Service Provider | An establishment utilizing the system to complement or replace their physical queuing system |
| Store/Attraction/Location | A store, location and attraction refer to the same thing in this context, that is the physical setting of a particular store/location/attraction |
| System | The application as a whole, including the user interface and all backend services |
| Terminology | Description |
| User Interface | The boundary used to collect input from and display output to a user |
| venueID | venueID uniquely identifies each serviceProvider in the application |

## 4.  User Stories

|  |  |
| --- | --- |
| Story Description | Acceptance Criteria |
| Tom is a 24-year-old young adult who just graduated and started working. Eager to climb the corporate ladder and find his place in the  company, he spends a great deal of time working. As such, the opportunities that he finds to spend time with his friends and family are precious to him. For Tom, having a nice meal with them after a long day of work is all he really wants sometimes. However, owing to his tight and busy work schedule, he finds it difficult to commit to reserving seats at an establishment, lest he disappoint his friends and family. Instead, he typically queues with them for up to 30 mins, taking the opportunity to catch up with them while waiting. | * View a store near his location with a queue time under 30 minutes * Join the queue successfully |
| Tze Minh is an impatient 21-year-old student who wants to visit fancy stores with her friends. She enters the mall and is unsure what to do but she immediately would like to see something popular yet has a reasonable queue under 15 minutes | * Filter stores by category * View the queue time of those stores * Join a queue of a store under 15 minutes |