**Restaurant Reservation System**

**Summary:**

Restaurant Reservation System will be used by restaurants to register and handle their customers easily. The aim of the system is to help restaurants in allotting tables to customers automatically.

**Business Flow:**

The customers details which includes From, name, and number of people are taken as input. This input is passed to the NodeJS Api via frontend iOS app.

The customer is allocated to table based on number of people and number of seats on that table. If no table is available for a customer based on his requirement then he is given a token number which represents his position in the waiting list queue.

Once a table becomes available then the queue count is decremented for every customer in the waiting list and one with the least queue count is allocated a table.

Each customer is given a check-in time depending on status of reservation. The customer can query the api to know his status via Twilio API. As of now the trial version of API is being used to add the customers number to the Twilio API.

There is an addon functionality by which a customer can ask for an extension of 10 mins once in his registration.

Every customer who has completed his dining is marked complete by the admin and the table is freed for next customer in queue.

The admin can clear all tables and reinitialize then by calling the api on start of the business day.

The admin can view the status of all customers anytime using the API.

**Programming Logic**

Core Technologies used:

* Front end mobile app
* Nodejs backend API
* Mongo dB cloud storage
* Twilio SMS api to communicate with server
* Moment library to calculate wait time
* AWS/ngrok for cloud hosting

**Table Schema:**

{entity: String,

tableNo: String,

numberOfSeat: String,

availability: String,

phoneNumber: String }

**Customer Schema:**

{phoneNumber: String,

phoneNumberTo: String,

customerName: String,

peopleCount: String,

checkinTime: String,

waitingTime: String,

status: String,

tableNo: String,

extension: String,

queueCount: String}

**Functionalities:**

post(/customerRegistration): Registers customers to queue or places them in waiting queue.

post(/smsInteraction): Used by the customer to send and receive SMS from the NodeJS API via Twilio regarding status and extension time.

post(/initializeTables): Used by admin to clear all tables.

post(/initializeSpecificTable): Used by admin to checkout a customer and assign the freed table to next customer in queue.

post(/displayCustomers): Used by admin to display status of customers on mobile app