CMPE 272 - Class Project

Have A Seat

Team #5

Team Members

Amit Pandey (011424231)
Jayam Malviya (011435567)
Sparsh Sidana (011413402)
Vimanyu Aggarwal (011006125)
Github Repo : https://github.com/SJSU272Lab/Fall16-Team5

Problem Statement:

How many times has it occurred that you went to a diner and couldn't find a seat? You probably ended up spending time in long queues, ruining your dining experience and getting frustrated – not knowing that there is a diner with available seats on the next street or there was a seat available just a few minutes ago when you stopped just for a small chat. Moreover, restaurant owners lose revenue (as well as food!) when a customer enters the diner right after the owner closed the cash counter machine. So, there is a need to develop a platform on which the consumers and the restaurant owners can communicate in real-time.

Target Audience:

@Consumers:

Customers who wish to eat at a diner in a specific location without making prior reservations.

@Owners:

Restaurant owner who are looking for improving opportunity costs and have seats available in their restaurant.

Proposed Solution:
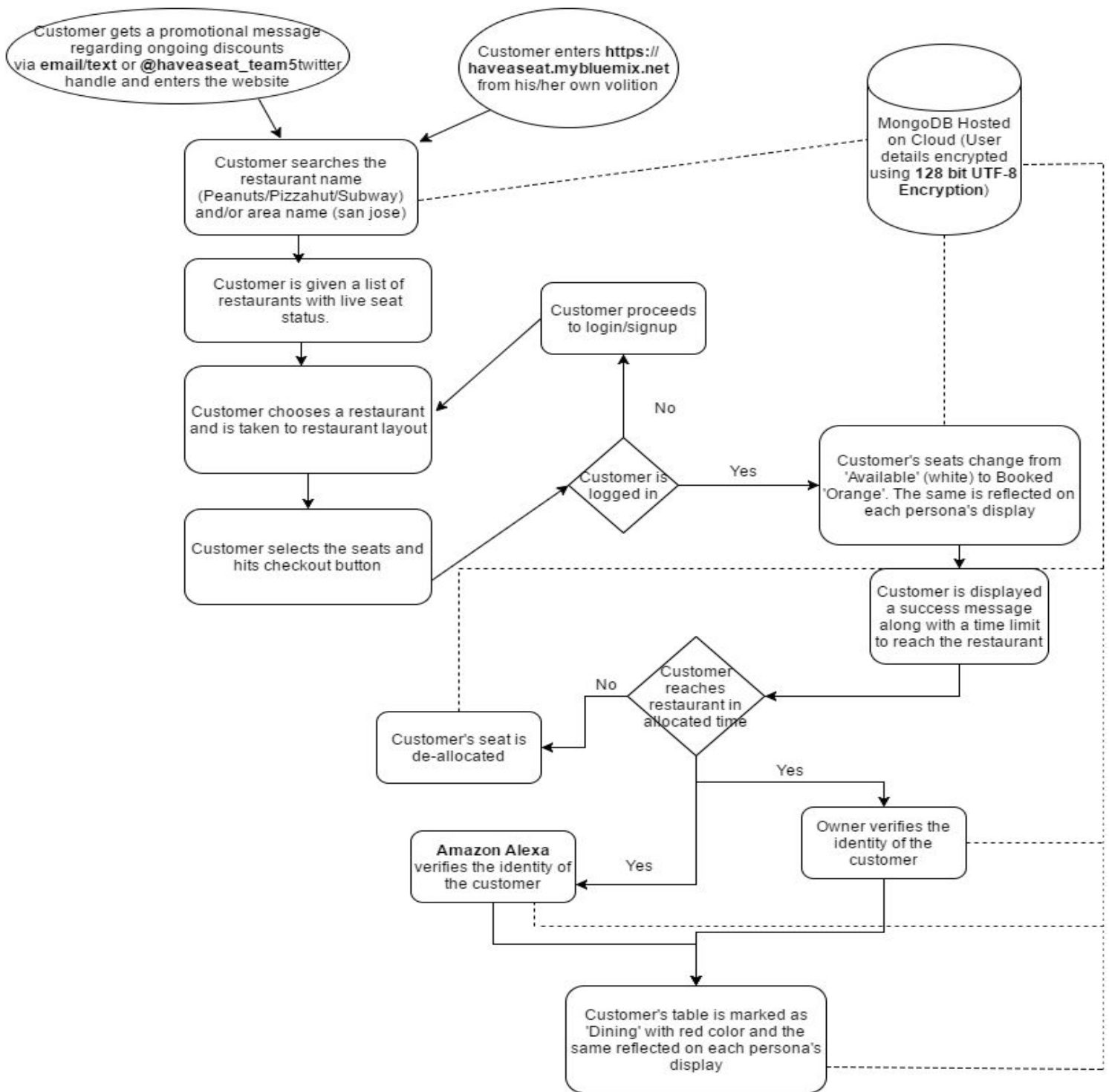
@Consumers:

Customers can now look up in the app and see if there is a seat available at the location where they want to eat. They can submit the request to book the seat, and the app will give them a time-limit to reach to the restaurant. If they reach the restaurant in the specified time, the owner can check-in them and mark their seat as booked. As soon as the owner marks the seat as booked, it will be reflected across all the apps in real time.
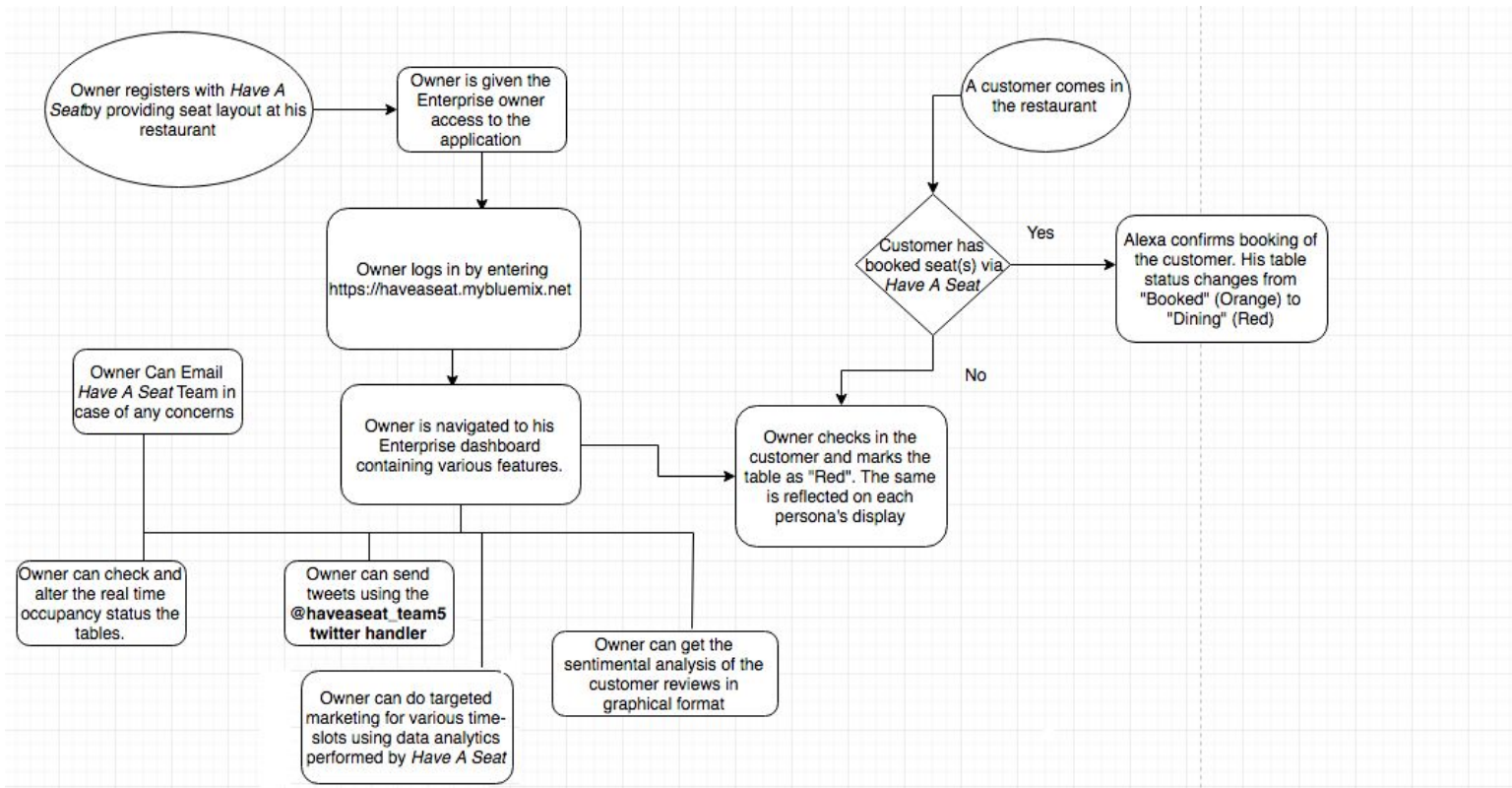
@Owners:

Owners will get a valuable information on his dashboard via ML algorithm based on time slots of his restaurants, in order to give offers to targeted customers. These customers are crunched out by Epsilon- Greedy algorithm, their emails are fetched and sent to owner's dashboard.
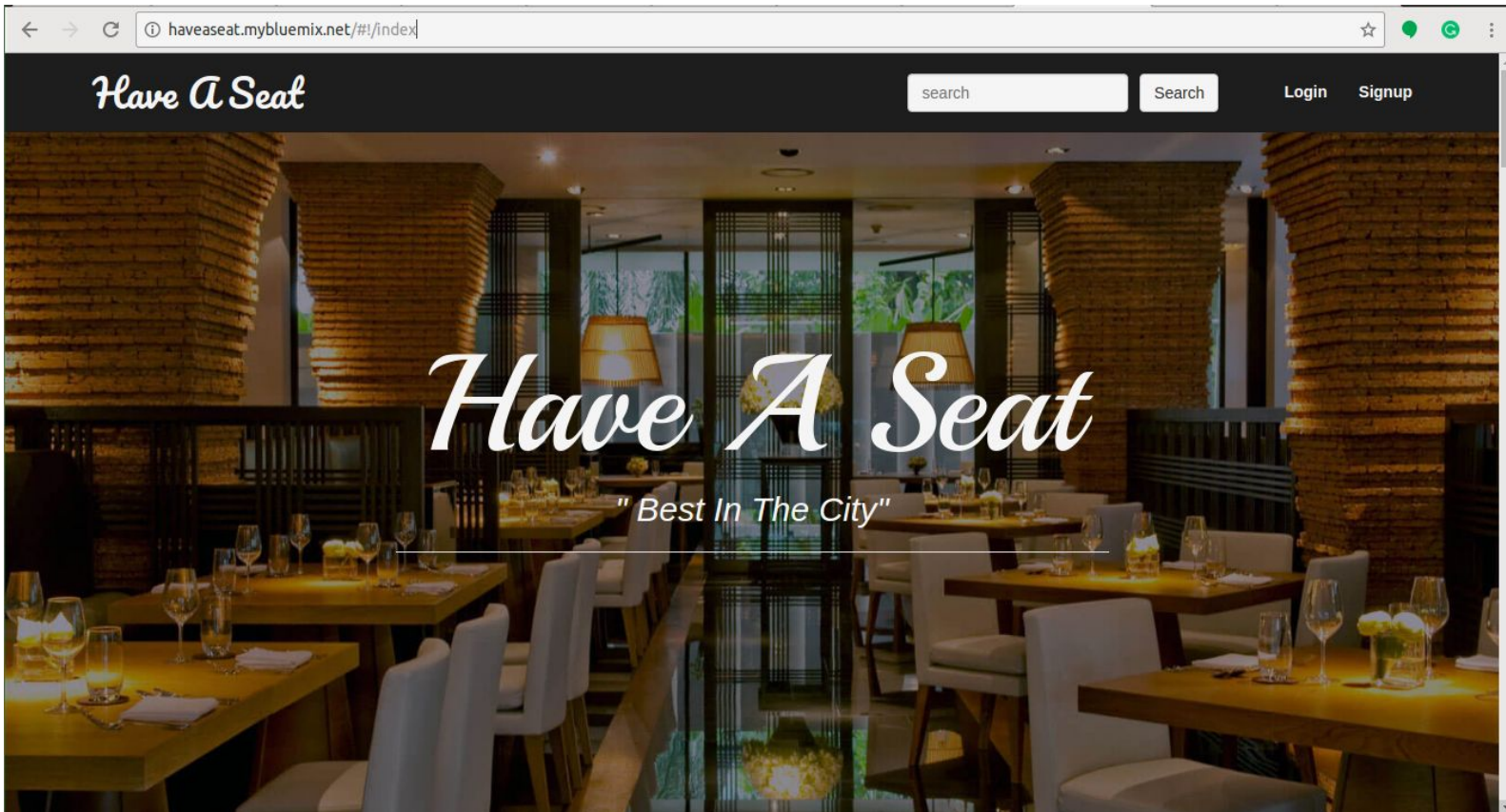
Flow diagram for Customer:

Customer gets a promotional message regarding ongoing discounts via **email/text** or **@haveaseat_team5**twitter handle and enters the website

Customer enters **https://haveaseat.mybluemix.net** from his/her own volition

MongoDB Hosted on Cloud (User details encrypted using **128 bit UTF-8 Encryption**)

Customer searches the restaurant name (Peanuts/Pizzahut/Subway) and/or area name (san jose)

Customer is given a list of restaurants with live seat status.

Customer proceeds to login/signup

Customer chooses a restaurant and is taken to restaurant layout

Customer selects the seats and hits checkout button

Customer is logged in

No

Yes

Customer's seats change from 'Available' (white) to Booked 'Orange'. The same is reflected on each persona's display

Customer is displayed a success message along with a time limit to reach the restaurant

Customer reaches restaurant in allocated time

No

Customer's seat is de-allocated

Yes

Owner verifies the identity of the customer

Amazon Alexa verifies the identity of the customer

Yes

Customer's table is marked as 'Dining' with red color and the same reflected on each persona's display

Architectural Flow Diagram for Restaurant Owners

- **Owner registers with *Have A Seat* by providing seat layout at his restaurant** → **Owner is given the Enterprise owner access to the application**
- **Owner logs in by entering https://haveaseat.mybluemix.net**
- **Owner is navigated to his Enterprise dashboard containing various features.**
- **Owner Can Email *Have A Seat* Team in case of any concerns**
- **Owner can check and alter the real time occupancy status the tables.**
- **Owner can send tweets using the @haveaseat_team5 twitter handler**
- **Owner can do targeted marketing for various time-slots using data analytics performed by *Have A Seat***
- **Owner can get the sentimental analysis of the customer reviews in graphical format**
- **A customer comes in the restaurant**
- **Customer has booked seat(s) via *Have A Seat*** — Yes → **Alexa confirms booking of the customer. His table status changes from "Booked" (Orange) to "Dining" (Red)**
- No → **Owner checks in the customer and marks the table as "Red". The same is reflected on each persona's display**

Screenshots & Explanation Of Application

Welcome Page for Customer and Owner

This is the default landing page when a user (either customer/owner) enters the website : https://haveaseat.mybluemix.net . This page contains the basic introduction to the concept of "Have A Seat" as well as provides the links to facebook page of Have A Seat, contact information form at the end. User input **fields are protected from XSS attacks and SQL Injection.**

Search page with search results and the "available seats" on the restaurants
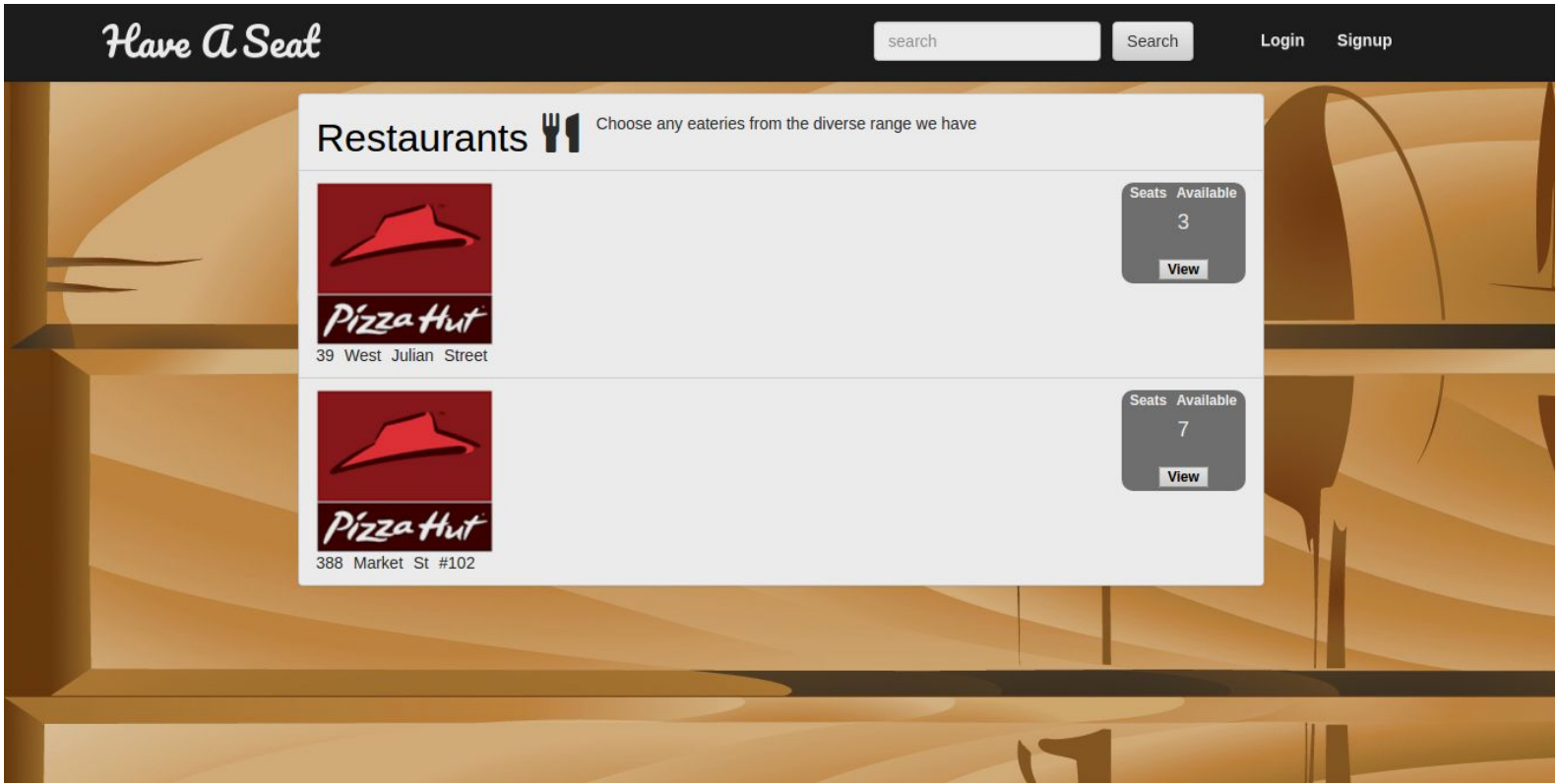
This page gives the live status of seats availability when user searches based on location and/or restaurant name. User input is filtered in the search box so that the search works even if the user enters keywords without spaces, in different cases (upper/lower) etc. **Customer need not login to view the live seat availability.**

Seats view of restaurant for customer persona :
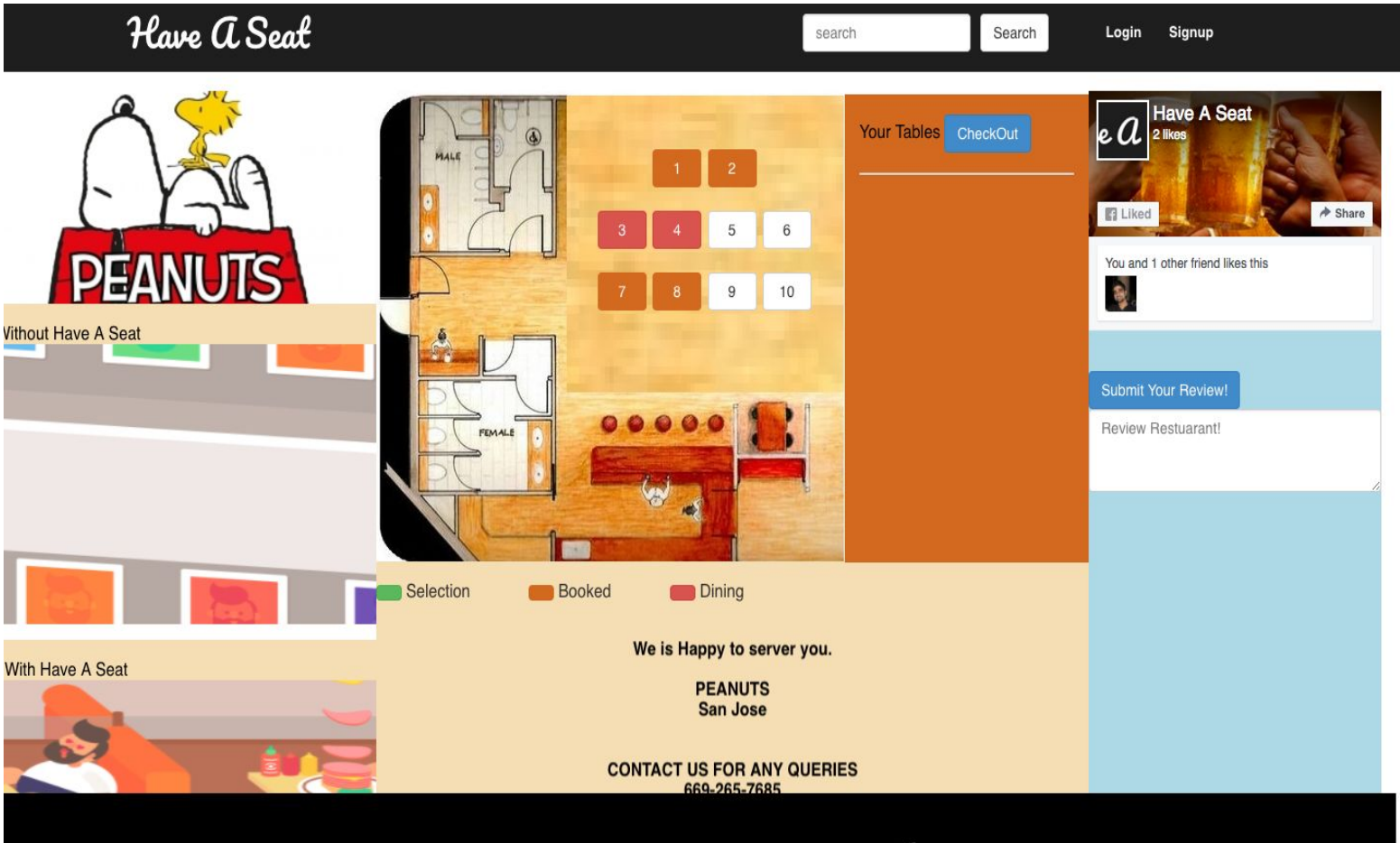
Customer can select the available/white seat(s) from the seat layout. User is given a ticker to contact the Have A Seat development team in case he/she has any concerns. Customer can also send reviews regarding the restaurant on this page. The **user reviews are fed to a sentimental analysis using NLTK (Natural Language ToolKit) library available in Python-Flask framework.** This analysis gives the insights in graphical format to the restaurant owner on his/her dashboard. Customer can only book seats after logging in but can view available seats without login.

User booking seats layout at the restaurant:
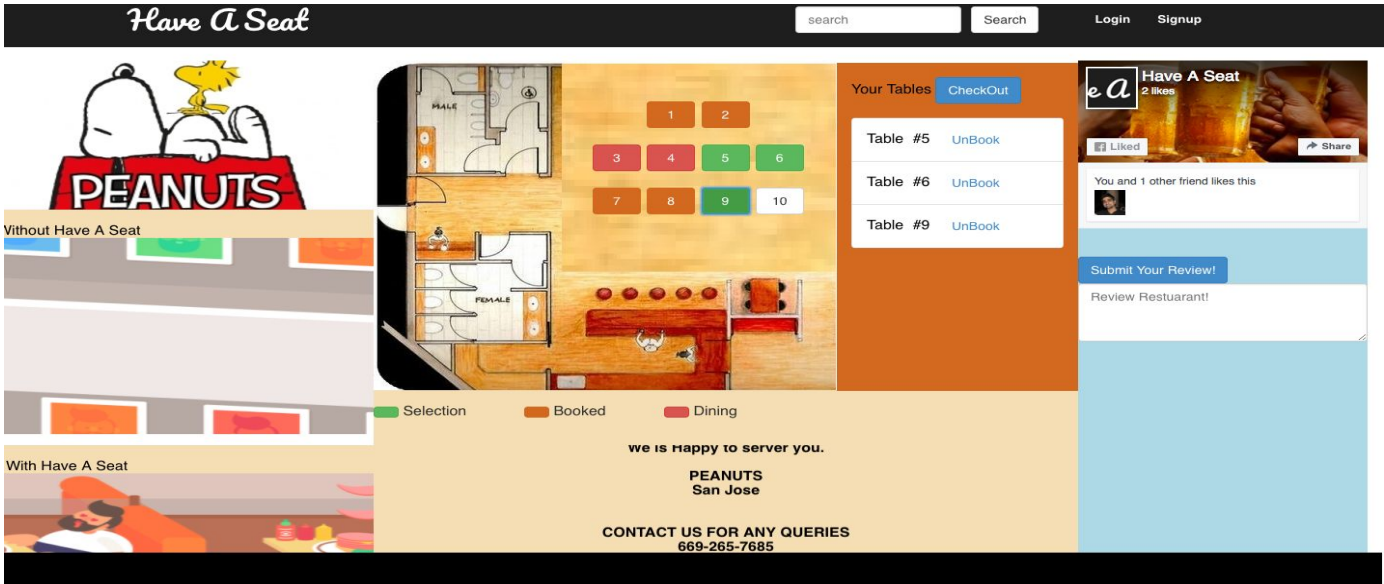
The customer can book seats on this page by clicking on the seat he/she likes based on their positions as per the layout. The customer can book multiple seats for dining. The color of seat changes to Green when the customer selects the seat. The seats selected are displayed below the checkout button whenever the customer selects one and the list of the seats is updated dynamically with the customer's selection.

Customer un-booking from the selected seats
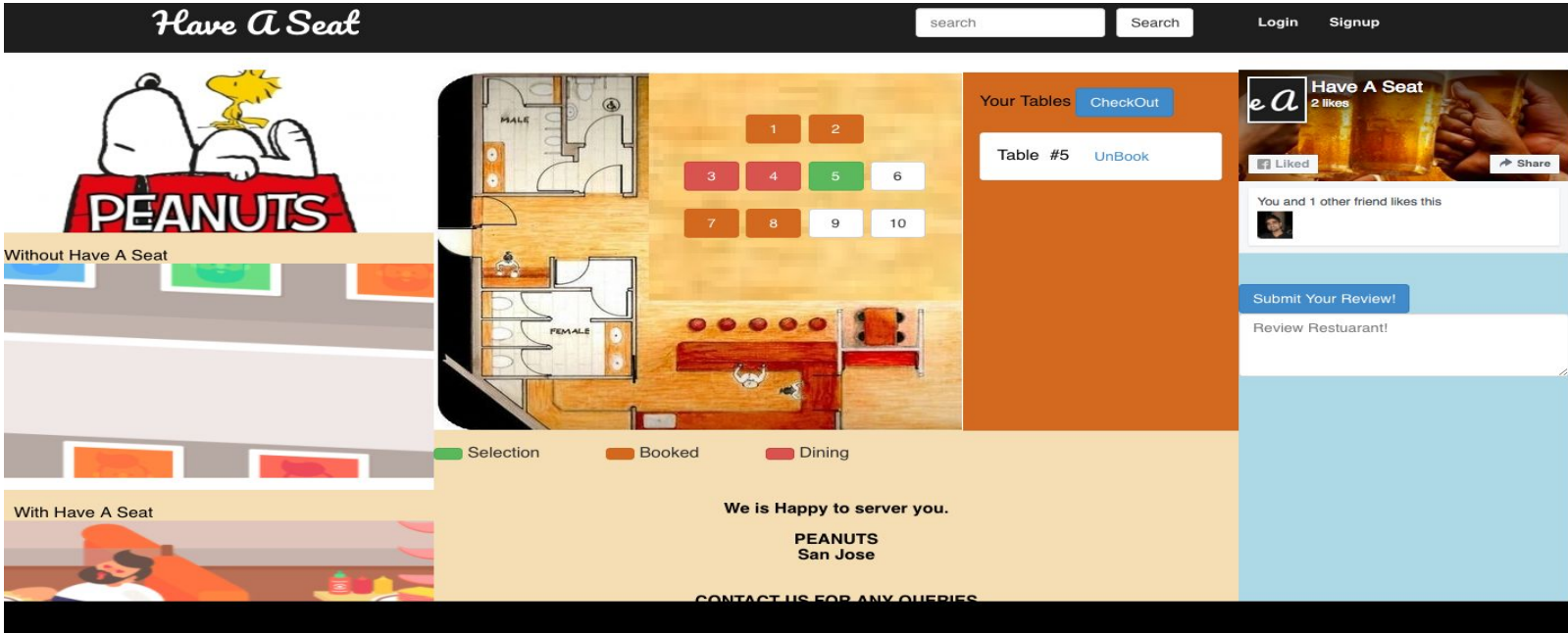
The customer can unselect the seats he/she had selected earlier by clicking on the unbook book button corresponding to the table number which is displayed below the the checkout button on the page.

Customer trying checkout without logging in
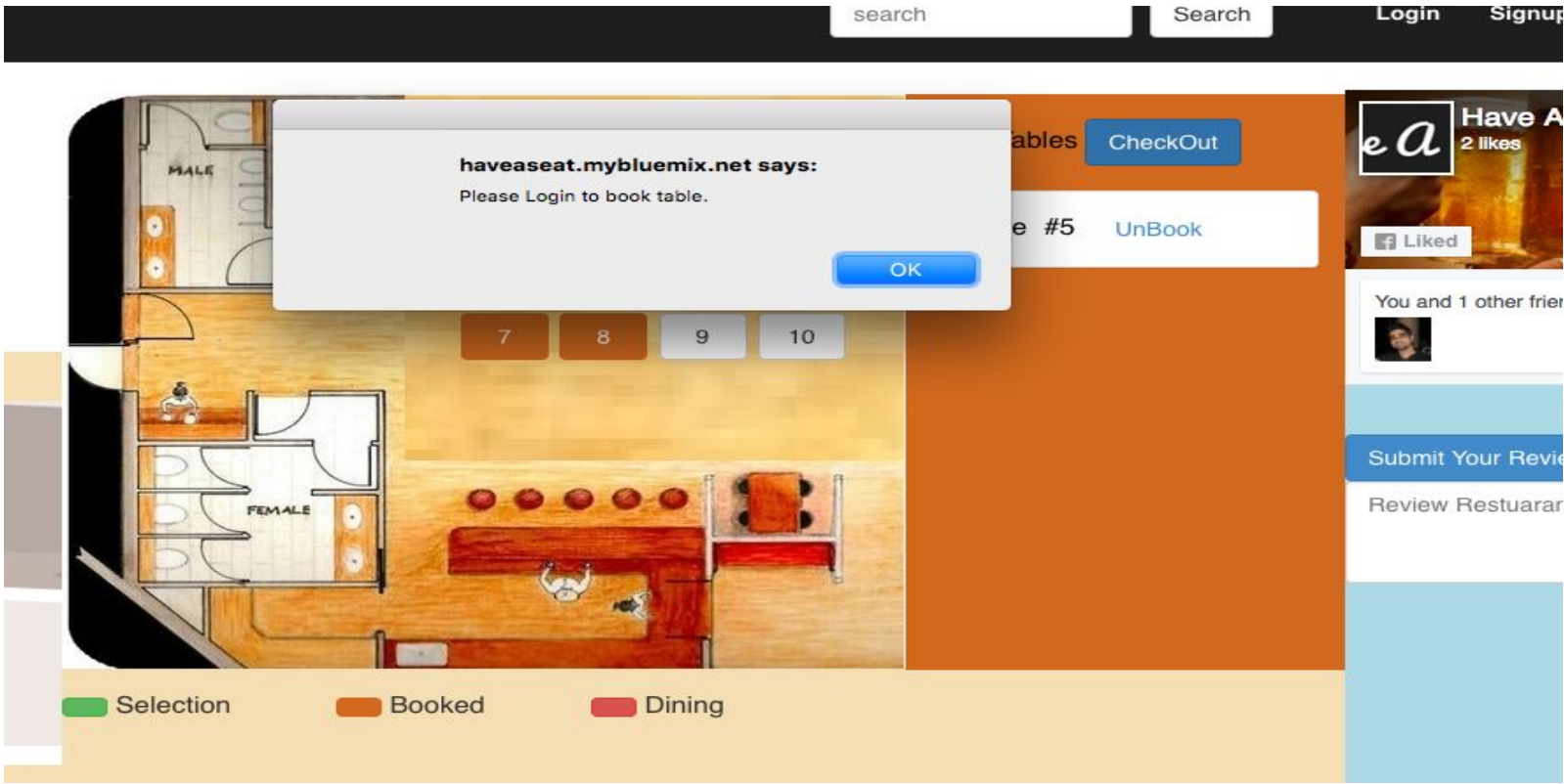
As we can observe the application allows the user to browse through the application, and view the currently available tables at the restaurants, but in order to check out his table he must be logged in. This screen depicts Login message that the user is given when tries to book a table without being logged in.

Login Page

User must use this page in order to login to the application, the same page is also used by the owner to login. Upon successful authentication the user is redirected to the home page. And if it's an owner he is redirected towards his dashboard. **All the users' passwords are encrypted in the database as it is hosted on a third-party.**

```
"_id": {
    "$oid": "5843df1af2fae95ffa2b0387"
},
"Password": "$2b$12$21klXHIYSVMr.w2SXRVYNu792VkqeSN57xAHhYza1ZXN1XhJAKqPi",
"customerName": "Emmy Stone",
"Email": "emmystone@gmail.com"

{
"_id": {
    "$oid": "5843df6bf2fae95ffa2b0388"
},
"Password": "$2b$12$BuJEia2z881/mWtrewN/seX9QGm.YlfDK0H2QJHaZibd82IWFT4qO",
"customerName": "Lolita Gonzalez",
"Email": "lolita@norsul.com"
```

As can be seen the passwords of the users are **encrypted using UTF-8 128 bit encryption** before storing in the database hosted on cloud using **Python's bcrypt library**.

Home Page showing Customer login
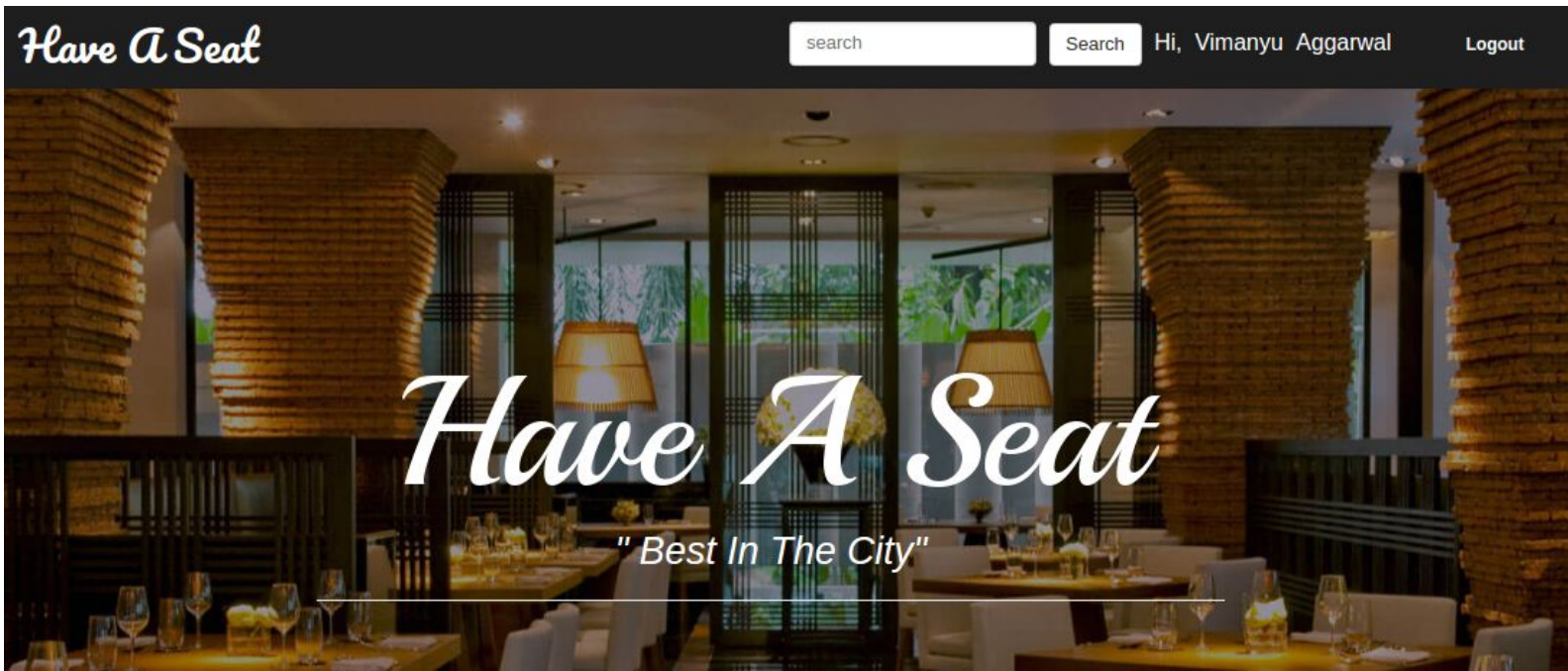
User login can be verified from the header. Customer is greeted with a Hi message in the header followed by his name. It is only after the successful login of the user, it is possible for him to book a table.

Customer successfully booking the Tables
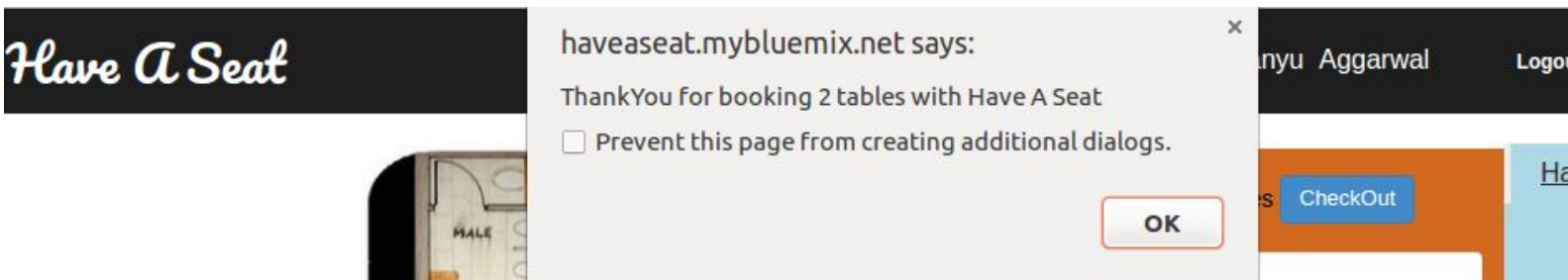
This message depicts that users booking has been successful and he is now ready to dine at his chosen tables at the restaurant. The message also tells how many tables were booked in his checkout. The status of the seats selected by him change from "Available" (White) to "Orange" (Booked). This change is reflected across all displays in real-time. **Customer also gets an e-mail confirming the booking.** After this page he is redirected to the Thankyou page.

Thank You Page for customer with timer
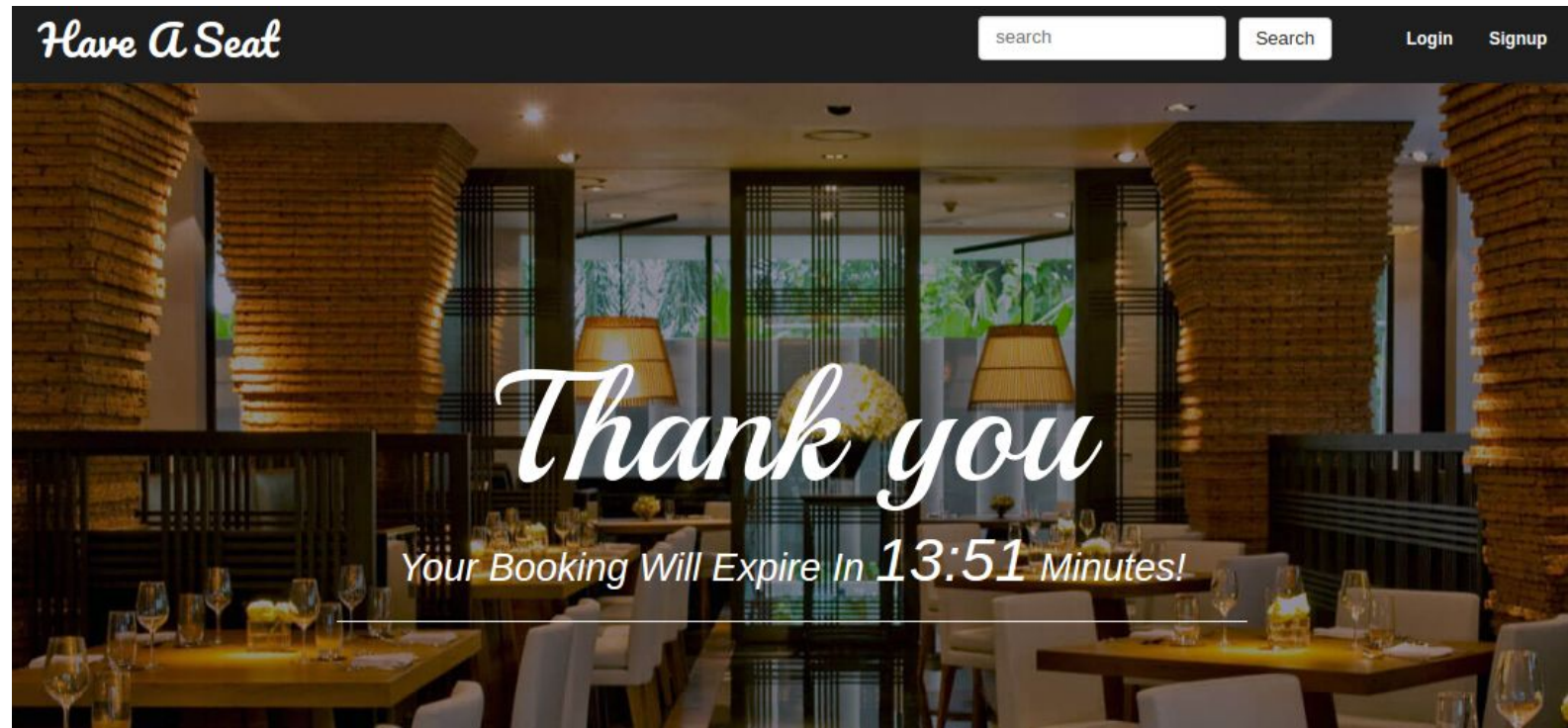
This is the Thankyou page to which the user is redirected after his successful checkout of tables. He is also given a timer at this page indicating the time left for his booking to be valid. After the timer expires the user's booking is no more valid.
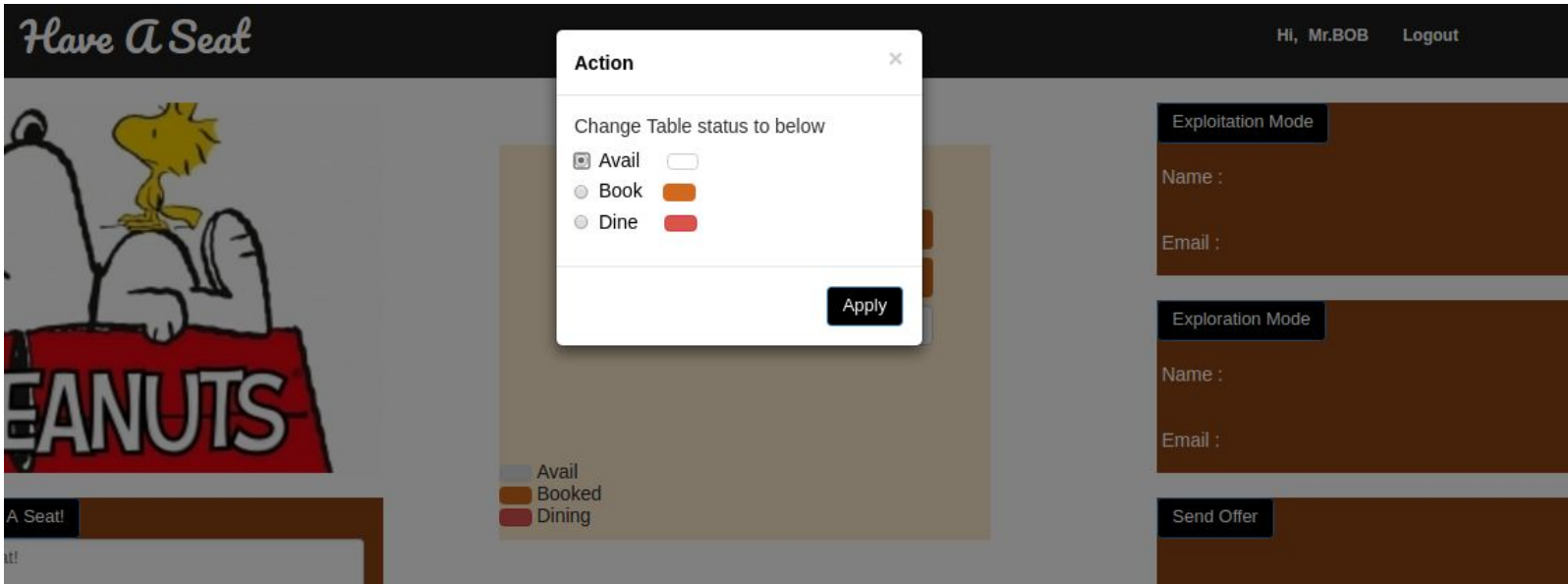
Owner Dashboard View of Restaurant

As he logs in the owner of the restaurant is redirected to the restaurant he owns, this is the home dashboard of the owner, where he can see the current status of all the tables in the restaurant. He can manage all the tables in the restaurant from this dashboard.

Owner making the changes in seat availability status

Owner can change the state of the table at the restaurant and it is reflected in the system instantly. He can make any table available, Booked or Dining as he feels appropriate. These changed state of tables will be reflected to the user in real time.

An Amazon Echo device will be deployed at the restaurant which will be helping the owner with confirming booked seats. Alexa will be assisting the owner by confirming customer details when they reach the restaurant after booking seats using the Have A Seat application. Alexa just need customer's name and the table number they booked and Alexa will confirm the booking if the details presented by customer are valid..

Tables Changed Status after owner made changes

Owner tweeting about his Restaurant

Owner can also tweet about the instant offer that he wants to give to its customers. Tweet can be send using haeaseat_team5 twitter handle. Customers will be benefited from this functionality as they can get instant tweets about the latest offer that the restaurant has to offer.

Various modes available to owner

Marketing strategy can be implemented in two ways - Give huge discounts to the low frequency customers (**Exploration Mode**) or give moderate discounts to high frequency customers (**Exploitation Mode**). **Epsilon-Greedy Algorithm is a simplistic algorithm that came out as the direct result of Multi-Armed Bandit Problem posed by Microsoft Research.** It has been proven that this method beats A/B Testing everytime.
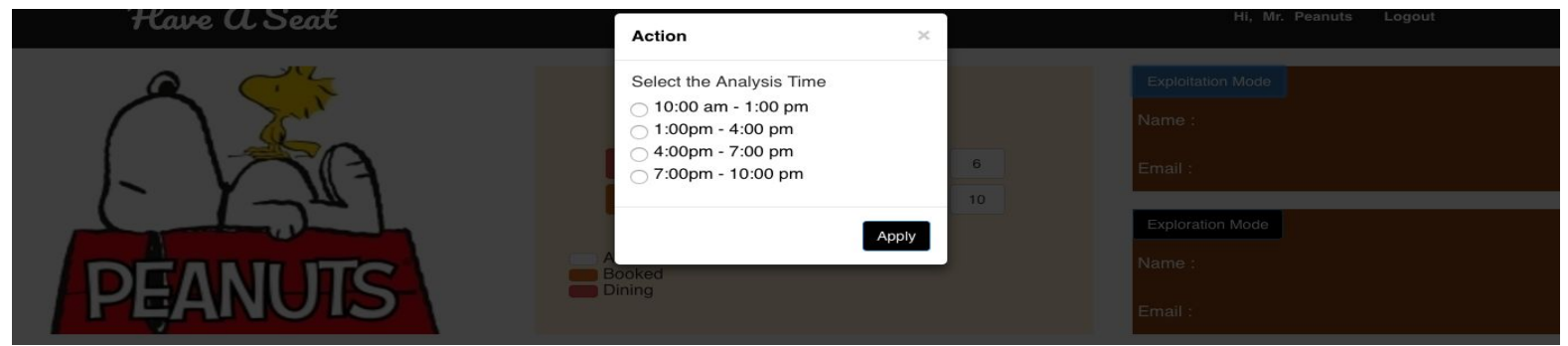
Emails crunched out by Epsilon-Greedy Algorithm enables the owner to do targeted marketing. This is further refined by providing time-slots to the owner which specify *in which time slot* does the owner wants to do targeted marketing. Because the algorithm plays on large data set - **the output is formed by a cron job which runs on BlueMix Platform at mid-night and is stored on Flask Server.** This saves the owner from waiting while the algorithm is crunching out the emails.

Owner gets the sentimental analysis of the customer reviews.
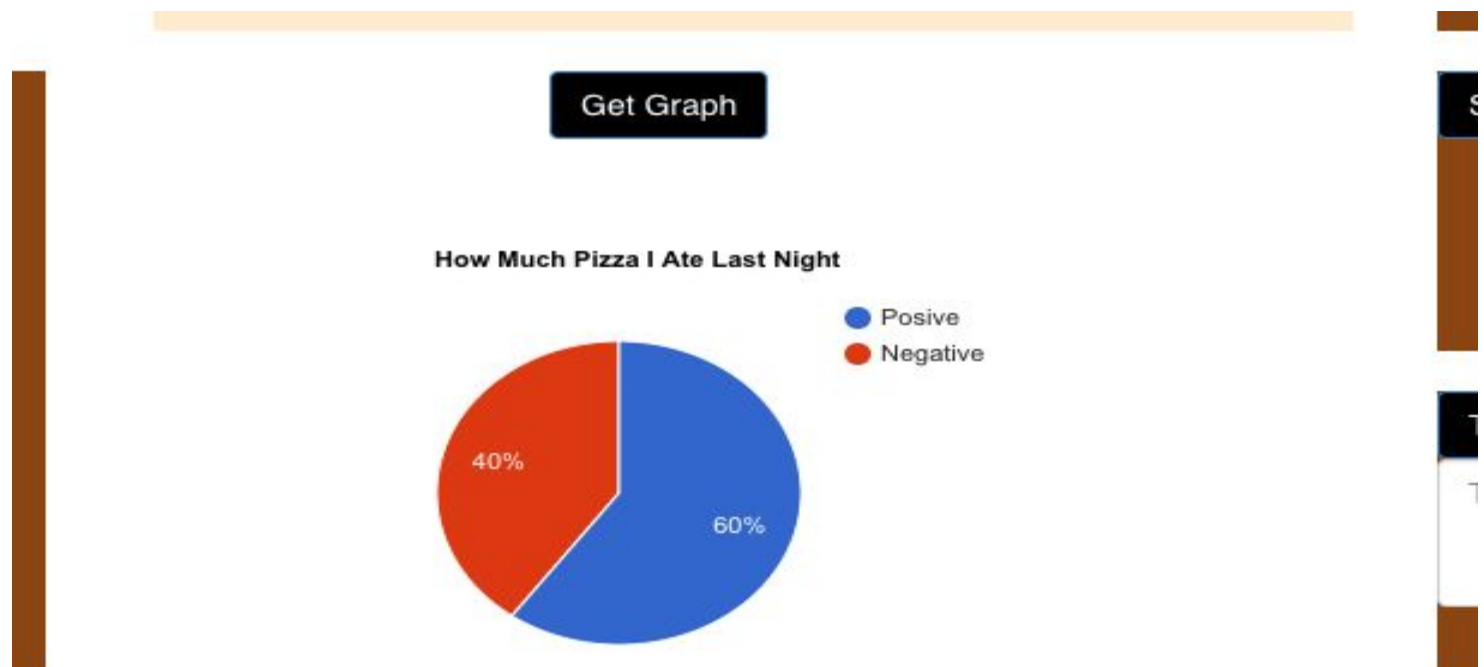
Owner can click on the Get Graph button to get the comparative analysis of the reviews that the customers have submitted on the Have A Seat website. The Have A Seat team provides this graphical view to the owner by performing sentimental analysis using NLTK (Natural Language ToolKit) library on the reviews submitted by the customers.

Owner Sending Email to *Have A Seat* Team.

Owner can send email to Have A Seat team, as per his requirements & needs. And team will be available to resolve their issues. This functionality can also be used to share instant feedback between Owner & Have A Seat.

Benefits to different personas from Project *Have A Seat* :
- Increased revenue of the restaurant owners by allowing them to contact *prospective* customers thereby allowing them to fill up the seats at the last moment.
- Improve the opportunity costs for the restaurant owners when they have seats available.
- Enables the owners to do targeted marketing and gain automated insights from the user reviews in graphical format.
- Owners can reach out the customers which were previously out-of-scope by using Have A Seat Database.
- Lots and lots of happiness to the Foodies!