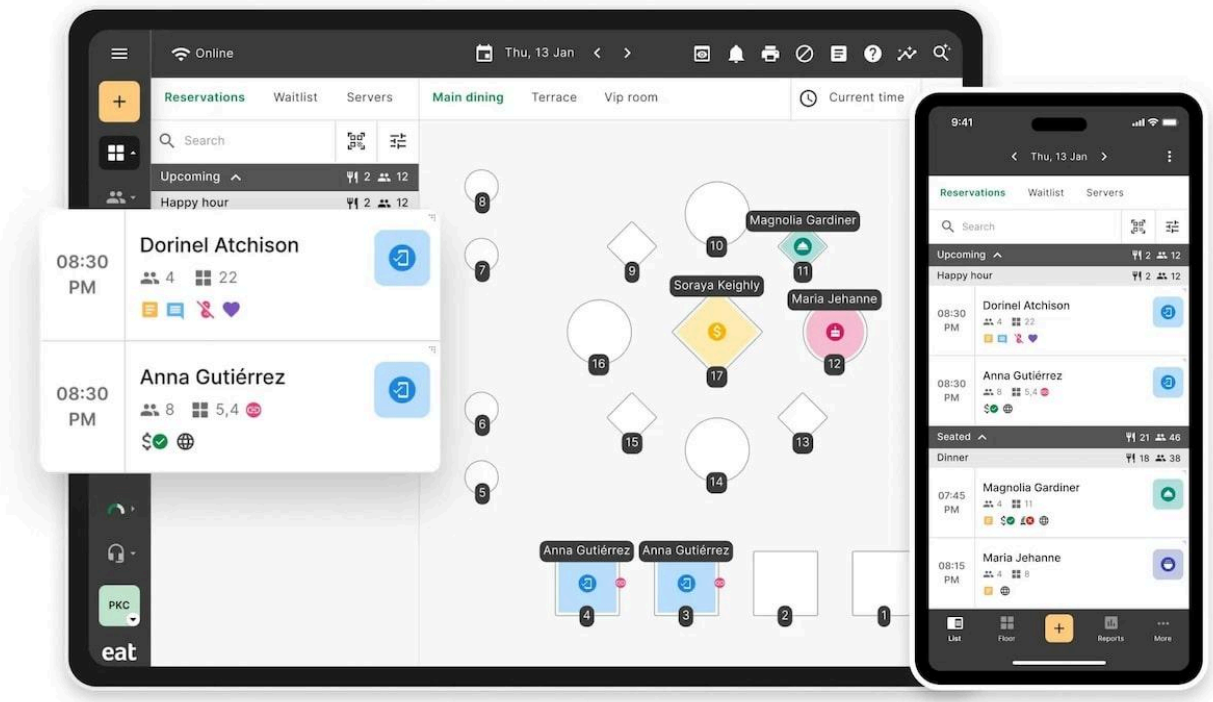
SoftDev P02: Makers Makin' It, Act I
TopherAcademy: Mark Ma, Anastasia Lee, Andy Shyklo, Victor Casado
TARGET SHIP DATE: 2025-01-17



Overview

Restaurant reservation app for customers and managers. On the customers side, they should be able to select # of people they expect to bring & the time, then the app should return available slots. They can claim/change a reservation through accessing their account. On the manager's side, they will be able to set up the layout of the restaurant (i.e. place tables, customize minimum capacity of each table), manage the waitlist queue, and view reserved tables.

Program Components

- Login Page
    1. Two types: Manager Account and Customer Account
    2. Email & Password login format
- Register Page
    1. Register for a new account as either manager or customer
    2. Same system as login
- Home Page
    - Once logged in, a customer could select and search for a restaurant by name, and then click on it to go to the restaurant page itself.

    - A manager could view their own restaurant and create/edit parts of it, as well as view the status of their tables.

- Algorithm for managing reservations
    1. Each restaurant is set up with a "time between reservations" variable. This variable represents the gap between each consecutive reservation at each specific table. For example, if Alice reserved table 8 at 9 AM, Bob can't reserve table 8 until 9:30 AM if the variable is set to 30 minutes.
    2. Returns a list of all available times when there exists at least 1 available table with minimum capacity >= number of people expected from reservation.

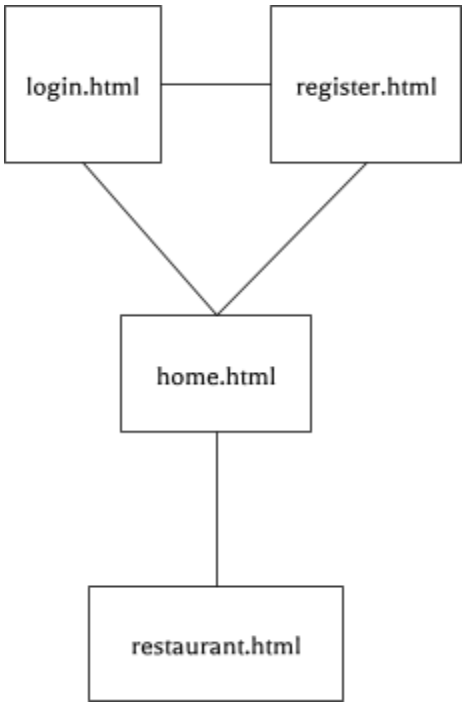
- If logged into Manager Account in Restaurant specific page:
    1. Setup Restaurant Layout
        - Create tables with maximum seats
        - Extra: The layout should be a 2D coordinate grid. Upon click, a table should be created at that coordinate.
        - Extra: Specify minimum capacity for each table. The minimum capacity should be taken into account for our algorithm for processing reservations.
    2. View the info of already processed reservations at each table at any inputted time (e.g. how many people expected? Email)
    3. Manage waitlist (EXTRA)
        - If no available preferable time slots, users should be able to input preferred date & time.
        - Managers should be able to view everyone who has an available waitlist at any requested time.
        - Managers should be able to remove people from the waitlist.

- If logged into Customer Account:
    1. Select a restaurant from the list of restaurants in the list.
    2. Immediately prompted to enter the date, time, and number of expected people for the reservation.
    3. User should be able to select table to reserve, prompted by our algorithm
    4. (EXTRA) If no tables are available, on the bottom, there should be an option "Add me to waitlist".
- Python/Flask components:
    1. We will have a main python file that will run the entire python-flask interaction. This will also run our algorithms and decide which site to display per type of user.
    2. Other python files will allow for interaction with the databases for both customers and managers.

Component Map

```
Frontend (HTML/CSS/Bootstrap/JS)
   |
   v
Backend (Flask) --> login/register/logout
   |
   +-- SQLite Database
   |      - User table
   |      - Restaurant table
   |      - Tables table
   |      - Reservations table
   |
   +-- Functions for managing restaurant setup
   +-- Functions for managing reservations
   +-- Functions for managing waitlists
```

Site Map



Database Organization (Column names with grey background, entries without)

- User Database

email	password	type
Topher@gmail.com	*********	customer
Mykolyk@hotmail.com	*********	owner
...	...	...

- Restaurants Database

Restaurant ID	Restaurant Name	Open Time (military)	Close Time (military)	Time Between Reservations (minutes)	Owner User
37	Berri's Berries	8	18	30	Mykolyk@hotmail.com
12	#GUDFAM Bagels	7	19	62	Mykolyk@hotmail.com
...	...	...		...	...

- Tables Database

Table ID	Restaurant ID	Number of Seats
23	37	2
2	12	3
...	...	...

- Reservation Database

Reserver User	Table ID	Number of People	Time (military)
Topher@gmail.com	23	2	9
Topher@gmail.com	2	2	17
...	...	...	...

APIs

- We do not plan on using any APIs.

Front-End Framework: Bootstrap

Why

Bootstrap is versatile and easy to use and comes with many predefined elements. Its designs automatically adjust to different screen sizes and are highly customizable, and do not depend on CSS as much as the other FEFs do.

How

We plan to use the following Bootstrap elements:

- Top navbar to navigate the site
- Forms for login, registration, restaurant creation, reservations

Task Assignments

Victor - Database Lead

- Create database and create functions to add / modify / get data

Andy - Python implementation from Databases/Flask

- Integrate databases within python and create access methods
- Make flask implementation using methods
- Focus on customer and manager specific methods

Chongtian - Flasky PM

- Facilitate seamless collaboration
- Manage Timelines
- Ensure Clear and Consistent Communication
- Work on back-end functionalities with Flask

Anastasia - Frontend Development

- Design responsive HTML pages with Bootstrap styling
- Add JavaScript features to make pages interactive