FOODECISIVE

Adam Greenfield, Aneesh Komanduri, Huy Mai, Tay Pham, Zhi Han Weng

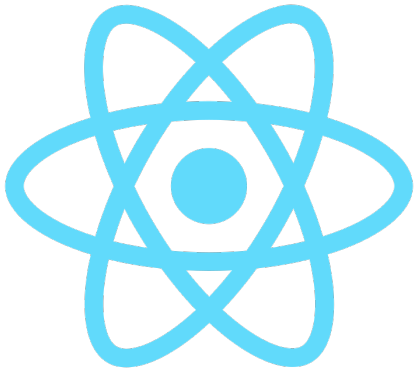
PROBLEM

- Choice Overload - when people have trouble deciding when faced with many options.
- According to the US Bureau of Labor Statistics, the time people spend on eating and drinking, and food preparation and cleanups are around an average of two hours per day.
- Average American couple spends 132 hours a year deciding what to eat.

OBJECTIVE

- Effectively provide *list of restaurants* to users based on their preferences
  - Improved recommender system to consider additional preferences
  - Complemented with map containing pinned restaurants
- Give users the option to *view and save multiple lists*
  - Requires authentication
- *Interactive* website

TECHNOLOGIES



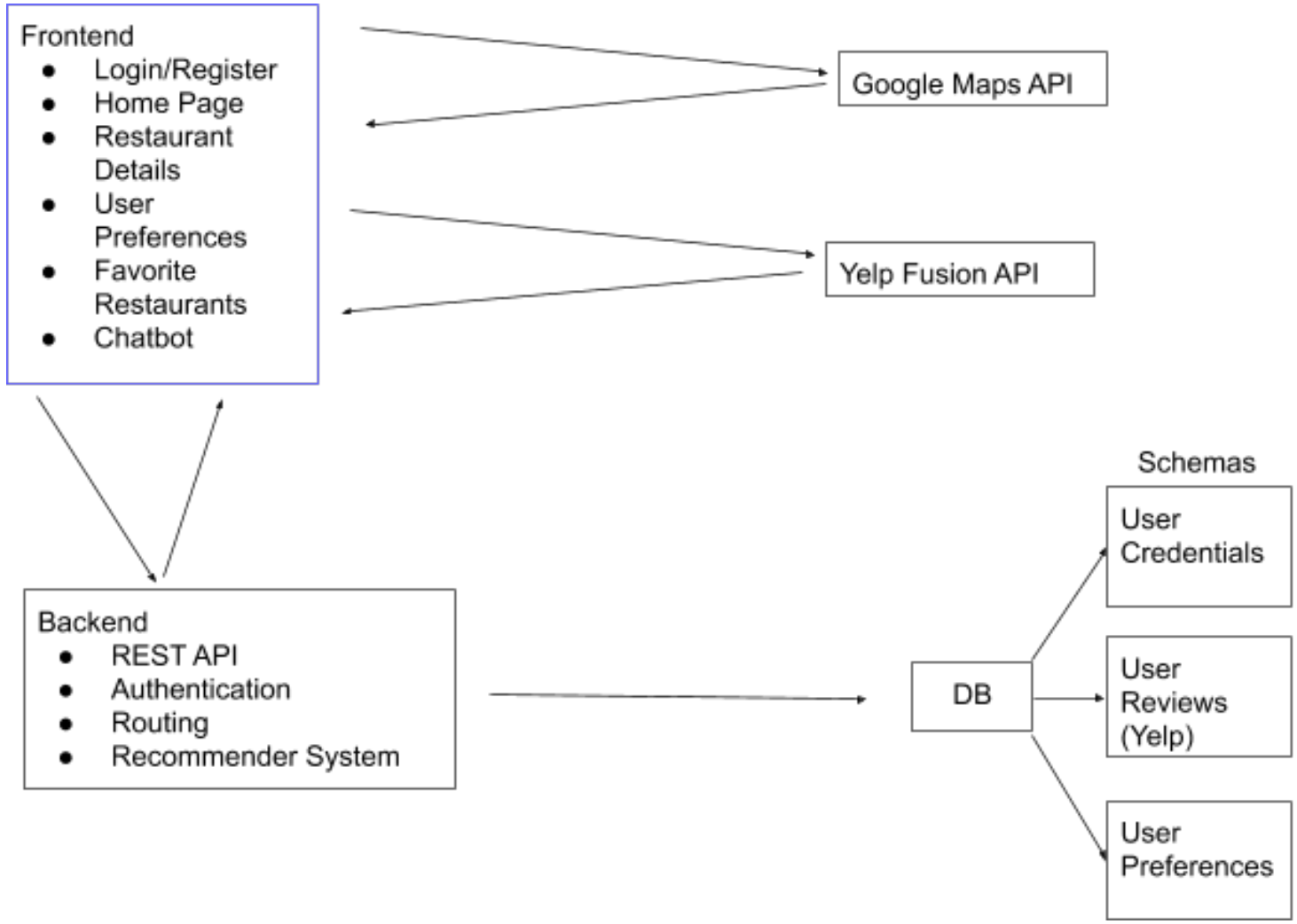
RELATED WORK

Wheel of Dinner (https://wheeldecide.com/wheels/food-drink/wheel-of-dinner/) - Basic implementation, does not take preferences into account.

Yelp (https://www.yelp.com/) - Uses effective collaborative filtering but can be improved.

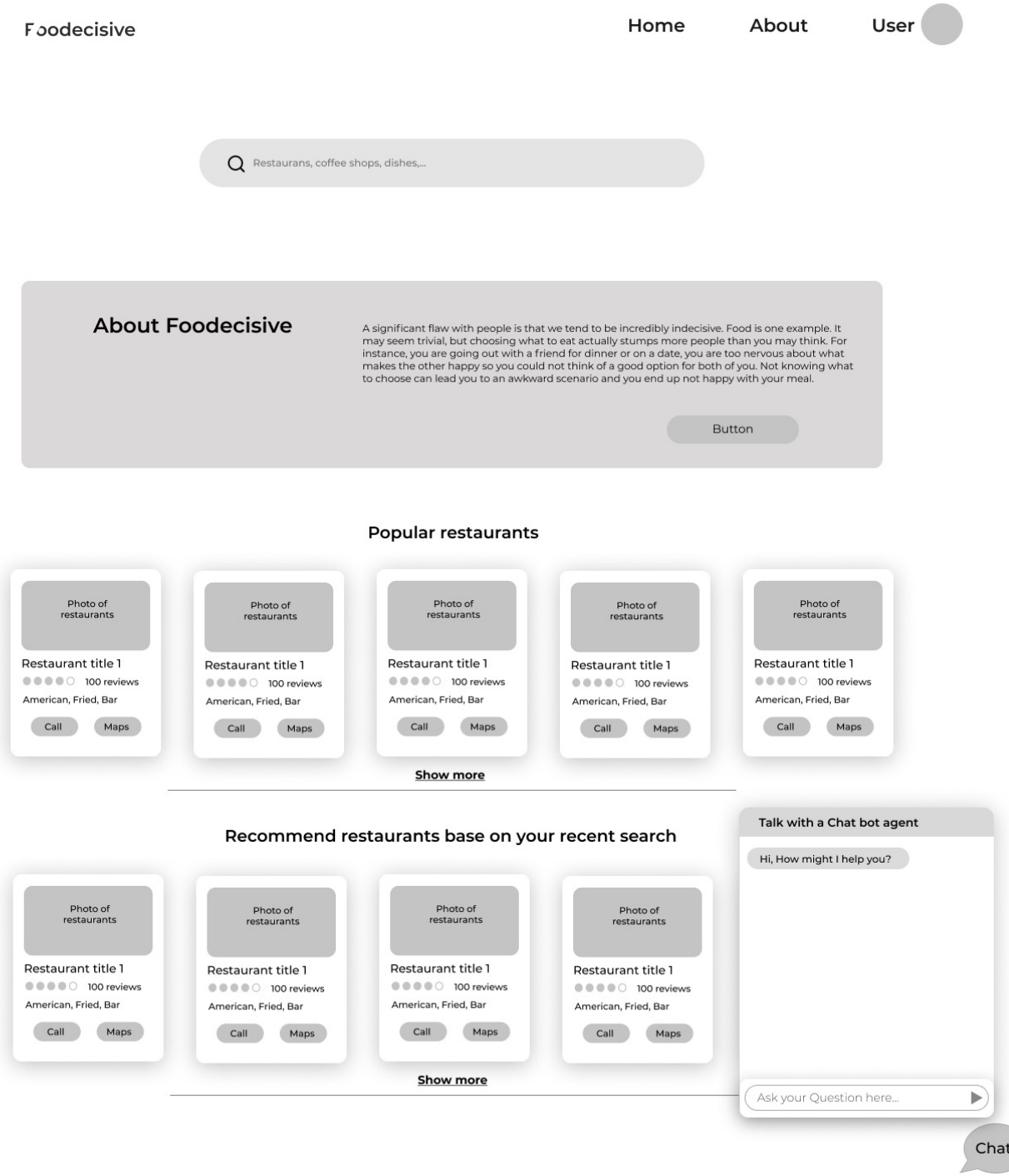
TripAdvisor (https://www.tripadvisor.com/) - Similar to Yelp, however some useful preferences are missing. Restaurants are just one part of Yelp and TripAdvisor.

ARCHITECTURE



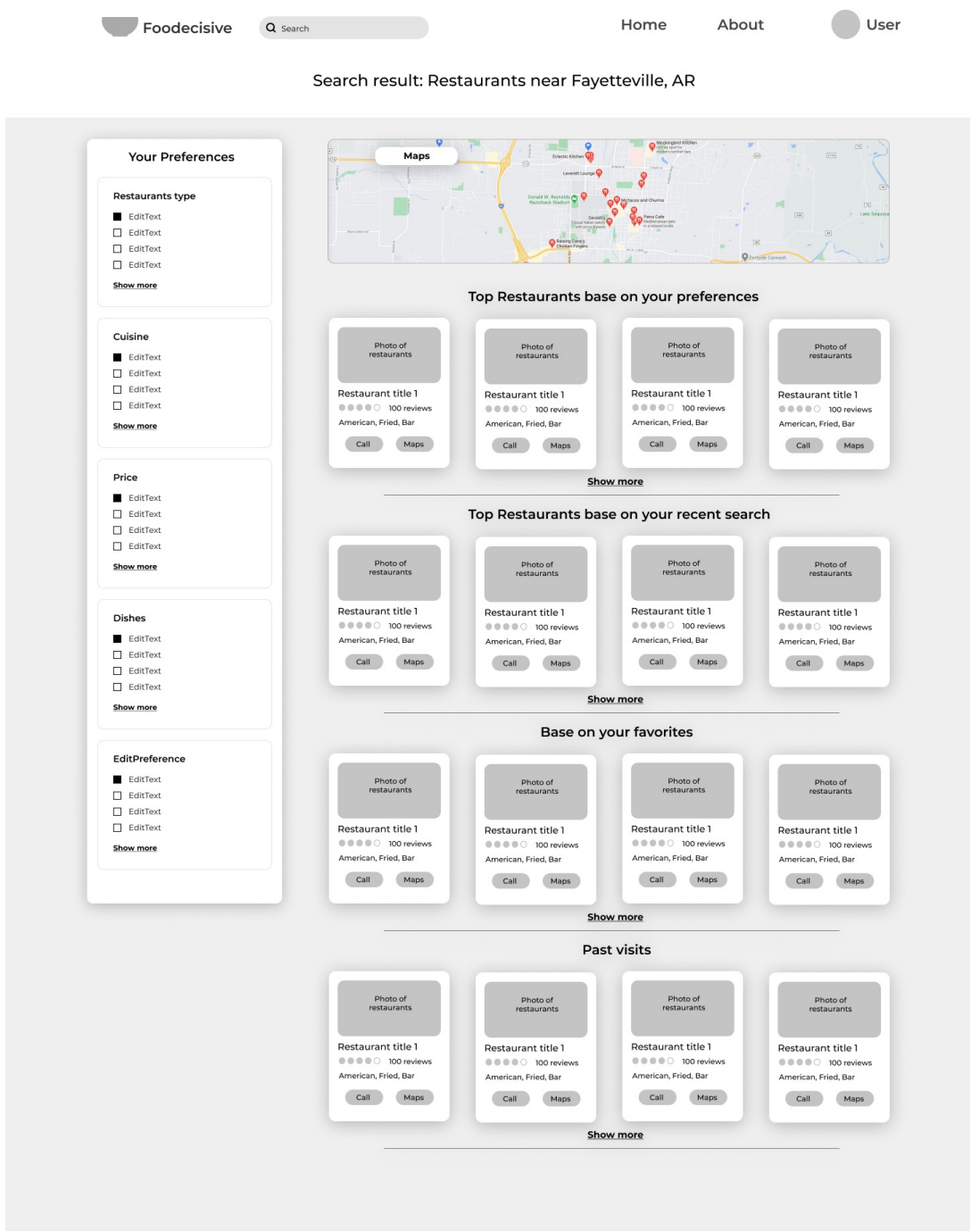
DESIGN (HOME PAGE)

- Register/Login Page and authenticate credentials
- Provide a search bar to search restaurants
- Provide a list of top-k recommended restaurants personalized to user
- Integrate Chatbot feature to request for restaurant information



DESIGN (DETAIL PAGE)

- User should be able to enter their preferences
- Integrate Google Maps to provide directions to a given restaurant (shortest path)
- Should be able to view location, ratings, genre, contact info, covid-19 protocols



TASKS/SCHEDULE

Tasks	Dates
1. Research recommender systems and design sketches of UI	11/14-11/28
2. Create React Skeleton, setup basic views, setup backend and start preprocessing of Yelp data	11/28-1/11
3. Establish database schema and connectivity to backend and start recommender system development	1/11-1/25
4. Implement Home Page (Search Bar, Initial Restaurants recommended, Toolbar, Preferences page) and continue recommender system development	1/25-2/8
5. Implement User Auth and Login/Registration Page	2/8-2/22
6. Connect frontend and backend and test data retrieval from frontend	2/22-3/8
7. Implement the Chatbot and integrate Google Maps API	3/8-3/22
8. Test functionality of Recommender System given user preferences	3/22-4/5
9. Polish up frontend for aesthetic look	4/5-4/19

DELIVERABLES

- Design Document: Explains what our code does...
- UI drawings and flowcharts
- Database schema and initial data
- React and Flask code
- Collaborative Filtering Recommender System code (Python)
- Final Report
- Research Paper **(IF TIME PERMITS)**