



Yelp Recommender System

Frederick Tan, Jianqiao Ge, Yunyan Shi, Benjamin Wang

Yelp Open Dataset

An all-purpose dataset for learning



The Yelp dataset is a subset of our businesses, reviews, and user data for use in personal, educational, and academic purposes. Available as JSON files, use it to teach students about databases, to learn NLP, or for sample production data while you learn how to make mobile apps.

The Dataset



6,990,280 reviews



150,346 businesses



200,100 pictures



11 metropolitan areas

908,915 tips by 1,987,897 users

Over 1.2 million business attributes like hours, parking, availability, and ambience

Aggregated check-ins over time for each of the 131,930 businesses

Get Started

[Download Dataset](#)

Visit the [documentation](#) for information on the structure of the dataset and how to get started.



Motivation and Problem Statement

Inspiration: Large amounts of data available, leading to the rise of recommendation systems that use this data and predict what a user would like.

Recommendation now an integral part of the online shopping and entertainment experience

Build a recommendation system that provides a list of recommendations based on previous ratings user gave to certain businesses

Better suit customer needs of Yelp users and enhance their user experience



Background Information (AWS)



Elastic Computing Cloud(EC2)



A cloud computing platform that

- Allows users to rent virtual computers to run their own applications
- Encourages scalable deployment on a pay as you go model
- Gives developers the ability to control how many resources are in use at any given point in time



Amplify



A set of purpose-built tools and features that enable frontend web and mobile developers to quickly and easily build full stack applications.

- Manage production
- Add custom domain
- Add and test new features privately



Amazon RDS



Set up, operate, and scale a relational database in the cloud with a few clicks

Connect applications to any of the 7 Amazon RDS engines

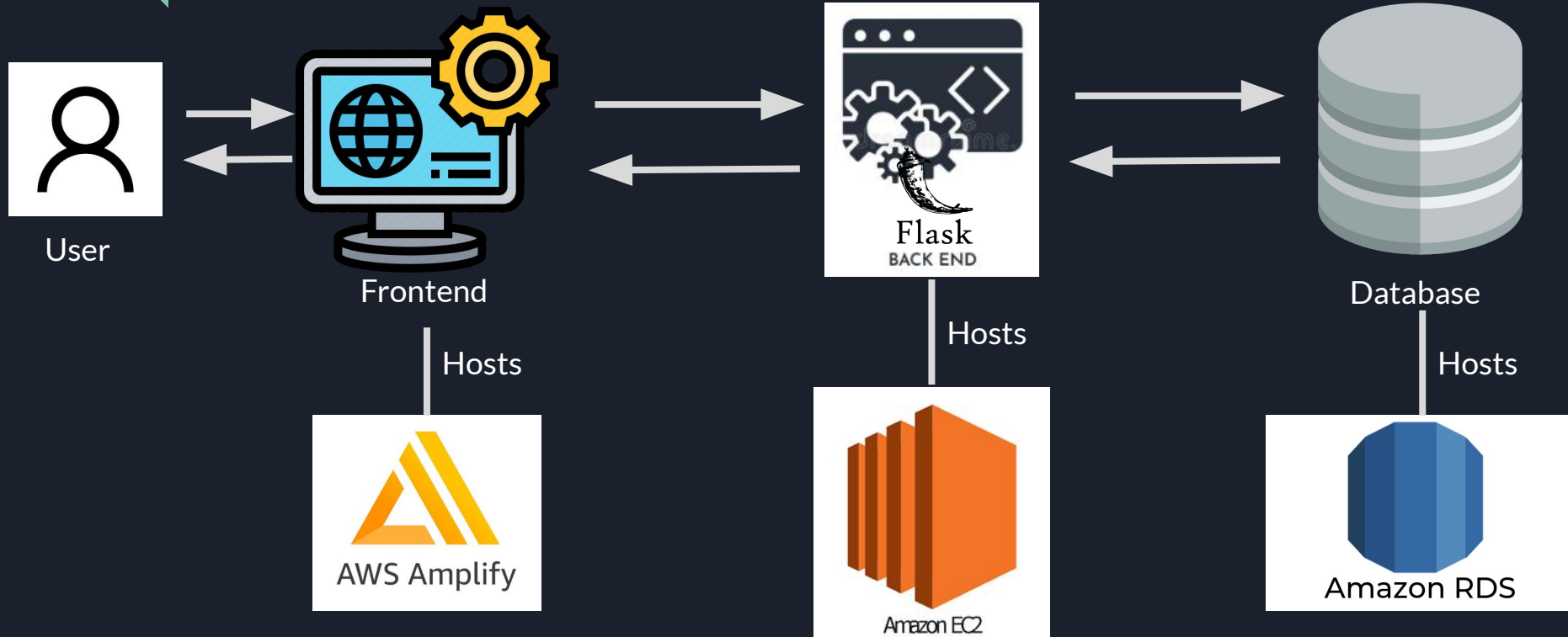
Support growing applications with high availability, throughput and storage scalability



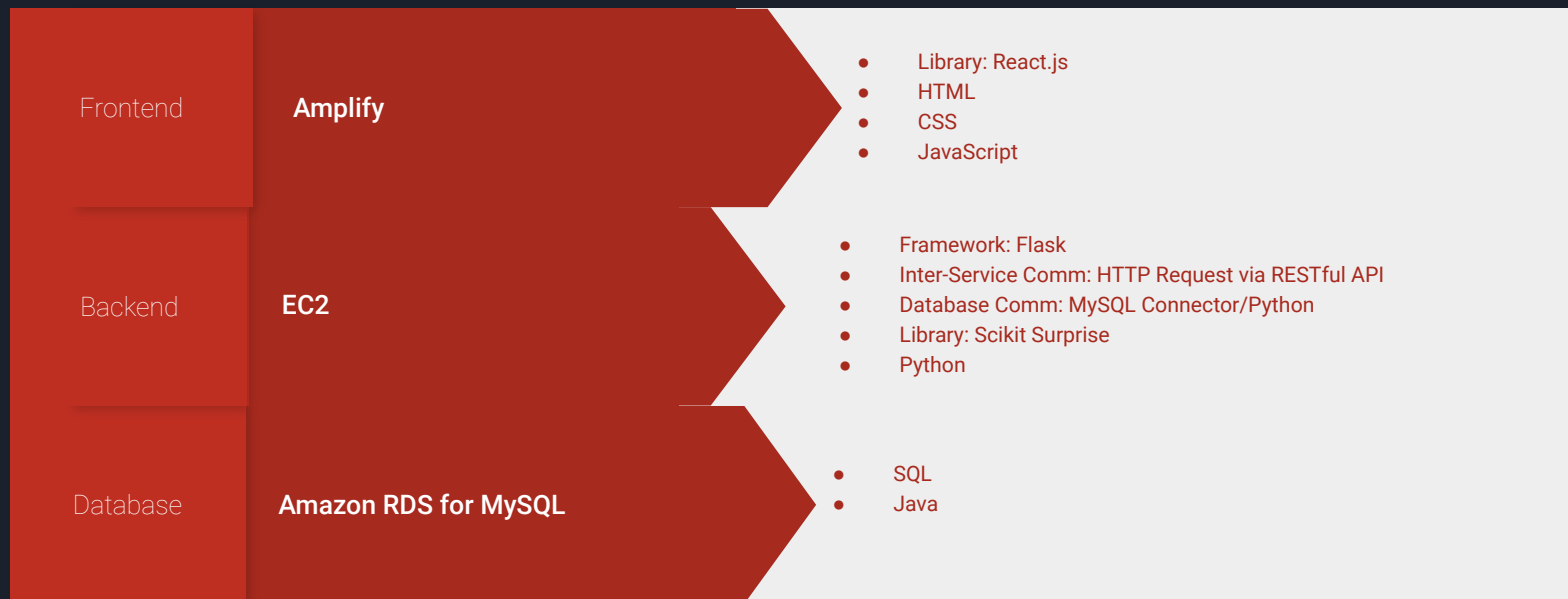
RDS for MySQL Use Cases

- Connecting from the MySQL command-line client
 - Create tables and add fields
- Connecting with the JDBC Driver
 - Populate tables with the data parsed and transformed from the Yelp Json dataset
- Connecting with MySQL Connector/Python
 - Query data

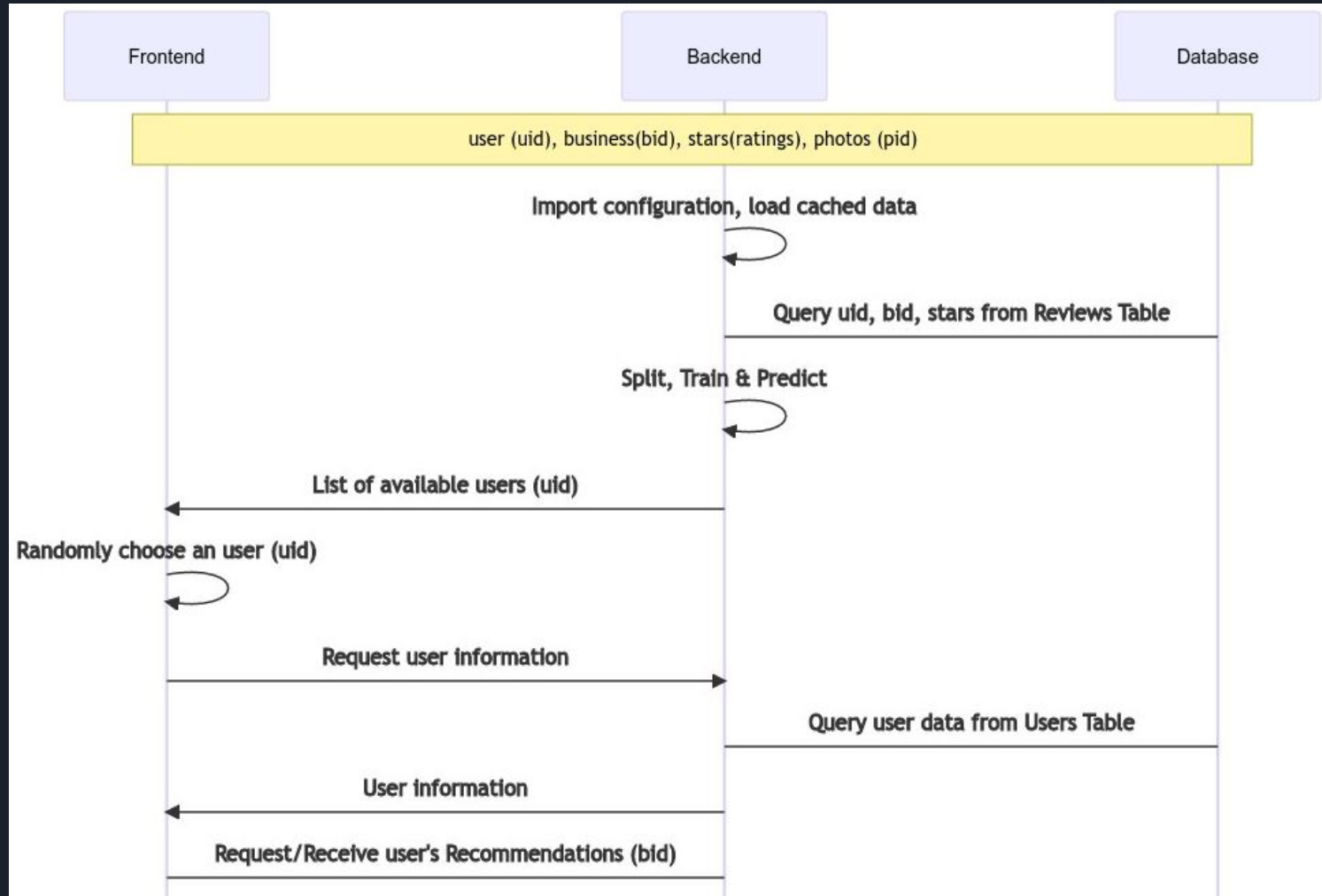
System Architecture



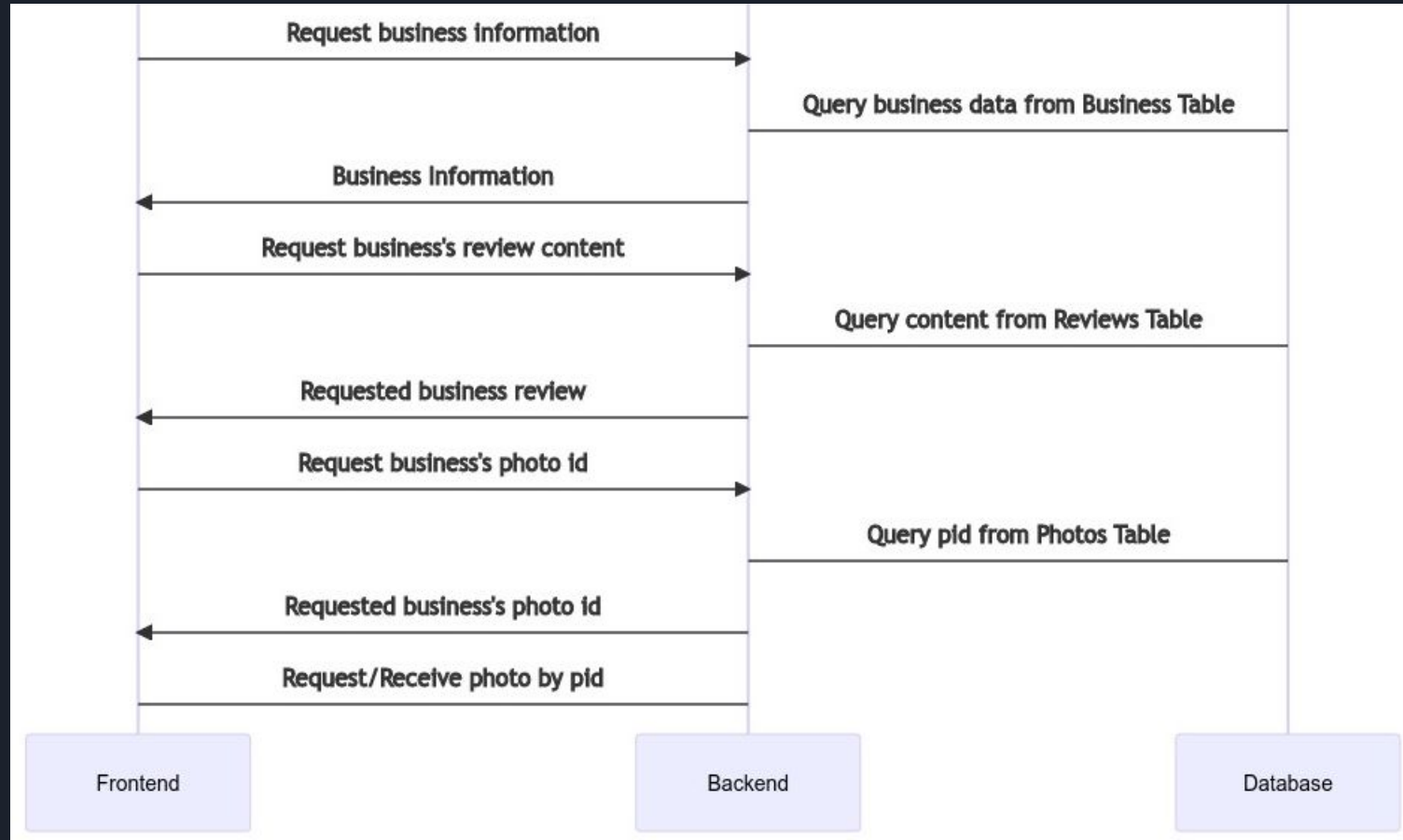
Technology Stack



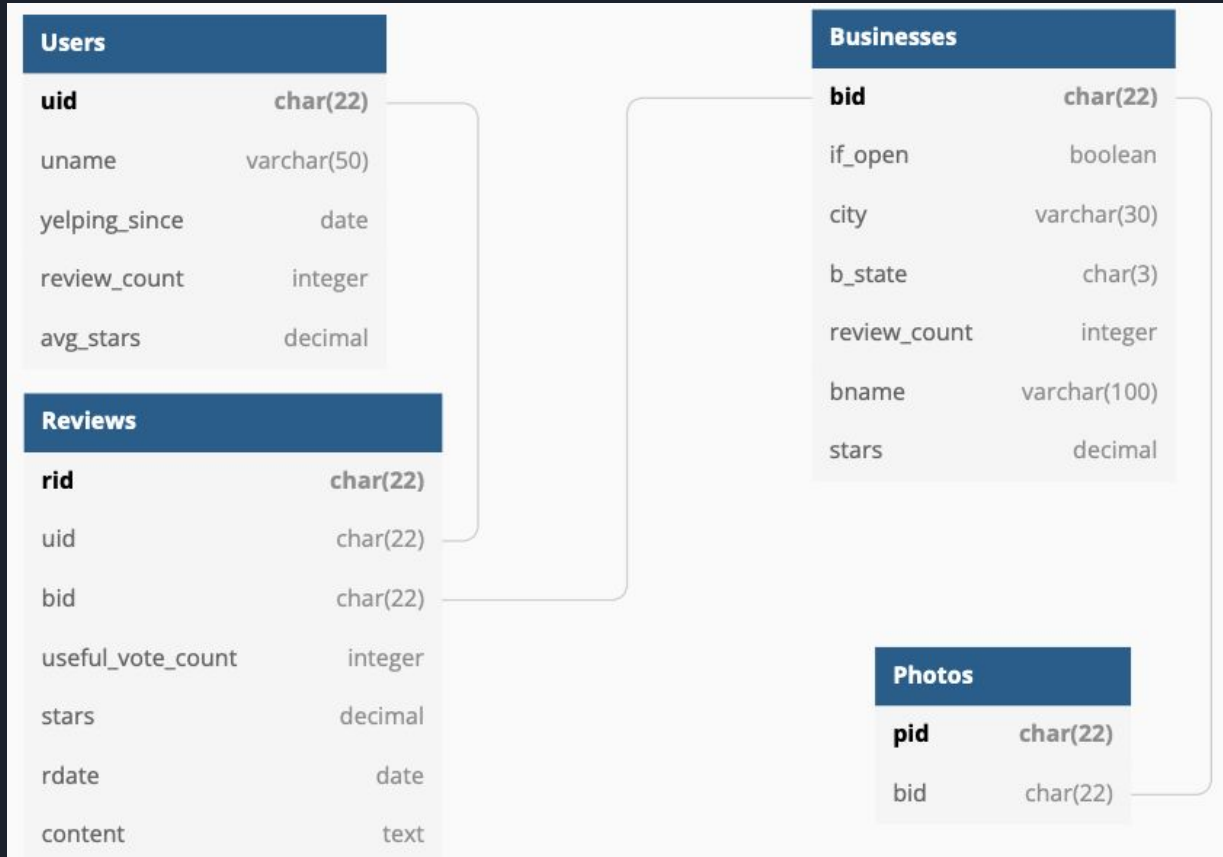
Software Architecture: Workflow



Software Architecture: Workflow



Database Schema





Outcome

- Functional Amplify App with random user selection
- Flask Delivery API that catches corresponding Business Picture
- Top Reviews of recommended businesses from Database query



Demo



Possible Future Work

- Use an optimized recommender library with multi-threading and sharding support
- Switch to ML model instead of traditional algorithm for recommendation
- Make recommendations with Novelty and better RMSE
- Add ability to create new users and login to see past reviews
- Make new reviews & ratings on businesses
- Add an event-driven hook to re-train the model and update predictions when the database is sufficiently modified
- Deploy the Flask backend on a (Docker) container and use K8s for orchestration so the application is scalable
- Or deploy using a Serverless service like AWS Lambda to make predictions upon frontend user request for recommendations



References & Related Work

1. Hug, Nicolas. "Surprise: A Python library for recommender systems." Journal of Open Source Software 5.52 (2020): 2174. <https://doi.org/10.21105/joss.02174>
2. Koren, Yehuda. "Factor in the neighbors: Scalable and accurate collaborative filtering." ACM Transactions on Knowledge Discovery from Data (TKDD) 4.1 (2010): 1-24. <https://doi.org/10.1145/1644873.1644874>
3. Sawant, Sumedh. "Collaborative filtering using weighted bipartite graph projection: a recommendation system for yelp." Proceedings of the CS224W: Social and information network analysis conference. Vol. 33. 2013. <https://snap.stanford.edu/class/cs224w-2013/projects2013/cs224w-038-final.pdf>



Work Split

Benjamin : Backend recommendations & management and EC2 setup

Yunyan: Database setup & management

Frederick: Frontend web application and Amplify setup

Jianqiao: Backend Flask API design and HTTP handler setup



Thank you!

Questions?