Computer Science 130, Homework 1

Michael Wu
UID: 404751542

April 17th, 2019

1 Introduction

1.1 Purpose

We will create a application that helps customers find good restaurants and reserve tables at them. This application will combine features from Yelp and OpenTable to ensure that users are able to find the best restaurants. Currently Yelp's system is biased because restaurant owners can pay to have their restaurant listed higher or appear better than their competitors. This process is detrimental to the customer experience, since the best restaurants do not get the recognition they deserve. Our application will not allow this so that our customers know they are getting the most accurate reviews. Yelp also does not have a reservation system, so when users find a restaurant they like they have to call to make a reservation. Our platform will allow reservations directly within the application to avoid this inconvenience. Restaurants will also be able to manage their reservations better on our platform. Our system will be available as both a mobile application and a web application.

1.2 Intended Audience

Our intended audience is customers who are actively looking for restaurants, and restaurant owners that want to increase their publicity. Both these groups would benefit from this application since customers would receive better restaurant results while owners will receive more customers. We will also set up an interface for our own administrators in order to manage customer and restaurant accounts.

1.3 Intended Use

The customers will create accounts to use our application to search for restaurants that are highly rated. They are able to search by location, cuisine type, price range, restaurant name, or specific food items. When they find a restaurant, they can view more information such as a description, reviews, pictures, hours, contact information, and a button to make reservations. When the user clicks the reservation button, they can select a given time and the reservation request will be sent to the restaurant. Users can leave reviews with a score and text on a restaurant's profile.

Restaurant owners will create accounts prior to being listed with our platform. In the restaurant owner interface they can upload data about their business such as contact information, pictures, and hours. They also have a system to manage reservations and will receive any reservation requests from customers. They can confirm or cancel reservations inside our application.

1.4 Scope

Our application will consist of a mobile application and a web portal. Customers and restaurant owners can log into accounts on these platforms that allow them to use the review and reservation system. We plan to first launch this service within the United States, and it would include all categories of restaurants. Even if a restaurant does not take reservations, we would include it within our service. This way we can have a wider appeal as there is more variety in restaurants. We will not try to include extra features such as ordering for delivery in our initial build, as these can be developed later.

2 Overall Description

2.1 Product Perspective

Our product should allow customers to simplify the restaurant finding process. It makes it easy to find unbiased reviews and make reservations at a restaurant. Restaurant owners will also use our platform to seamlessly manage reservations and upload information to promote their restaurants.

2.2 User Needs

The main user need is to easily find quality restaurants with minimal hassle. They must have confidence that the reviews on our platform will be accurate and reliable. Additionally the user should be able to find most restaurants through our service, since we do not want to leave any good restaurants out. Users also need to be able to make reservations with the click of a button so that their dining experience is seamless. The response time of our reservation system should be fast.

For restaurant owners, our platform needs to make it easy to set up their restaurant profile and reservation system. We should help them learn how to use our application and make it easy to upload information about their business.

We also need method for administrators on our end to detect and delete spam accounts or fraudulent reviews.

2.3 Assumptions and Dependencies

We assume that customers will be able to call a restaurant if they are unable to confirm a reservation. Since we will not implement a messaging system, customers may be confused about why a reservation is declined. In this case only a call can resolve this issue. Additionally, we assume that restaurants will integrate our platform's reservation system with their current reservation system.

2.4 Constraints

Our application requires internet connectivity, so it may not work if either a customer or restaurant has no internet. We also want our servers to be available as much as possible so that our service does not experience outages. This means that we should implement a way to test and deploy changes to our service without interrupting our production build.

3 System Features and Requirements

3.1 Functional Requirements

User Class	Title	Description
Customer	Search	Restaurants can be found by location, cuisine type, price range, name, or specific food items.
Customer	Review System	People can leave and view reviews of restaurants. Users that leave many quality reviews will have theirs shown first.
Customer	Reservation System	People should be able to schedule a reservation with a restaurant and have that restaurant confirm the reservation.
Customer	Location Finder	Our platform should recommend nearby restaurants using some sort of location services.
Customer	Account Management	Customers can see their account info to see past reviews, change their password, etc.
Owners	Business Profile	Restaurant owners should be able to manage their business info and reservation list.
Admins	Moderation System	Admins should have an interface to remove unwanted spam and fake accounts.
Admins	Spam Detection	We should have some automated flagging system to mark potential spam and fake accounts.

3.2 External Interface Requirements

3.2.1 User Interface

The main user interface will be both a mobile application and a web application. The mobile application should work on iOS and Android, as these

are the most widely used mobile platforms.

3.2.2 Hardware Interface

We will set up a web server to store the information that we collect in a database. This can be either in the cloud on in a data center that we set up ourselves. Our mobile and web clients will communicate with these over the internet.

3.2.3 Software Interface

Our web server will have API calls that enable users to perform actions that retrieve or modify the information in our database. These calls must include authentication information.

3.2.4 Communication Interface

Our client software will make API calls to our database in order to send and retrieve information over the internet. Without internet our application will simply show an error message

3.3 System Features

Other features that were not already mentioned in the functional requirements include a set of tutorial messages that helps owners and customers learn how to use our application.

3.4 Nonfunctional Requirements

Title	Description
Scalability	Our service performance should scale reasonably as the number of users and restaurants increase. Our service architecture should be easily expandable.
Speed	Our service should be able to send and receive data with minimal delay.
Uptime	Our service should stay available at all times if possible. We should not have to take our servers down often for maintenance.
Reliability	Our service should store data in multiple locations in order to prevent loss of data. It should also work as expected and have minimal errors.

3.5 Design Constraints

The main constraints on our development come from limited budget and developer time. We cannot begin working with a large team, so we must find ways to prototype our application with a small team with a single server. Afterwards we can scale up our development with multiple servers and a larger database before releasing our application.

4 Change Management Process

Changes will go through the client Aashika. For each change we will have a meeting and draft a change proposal in order to document the reasons why we are making a change. During our change meetings we will also specify the additional costs of making a change.