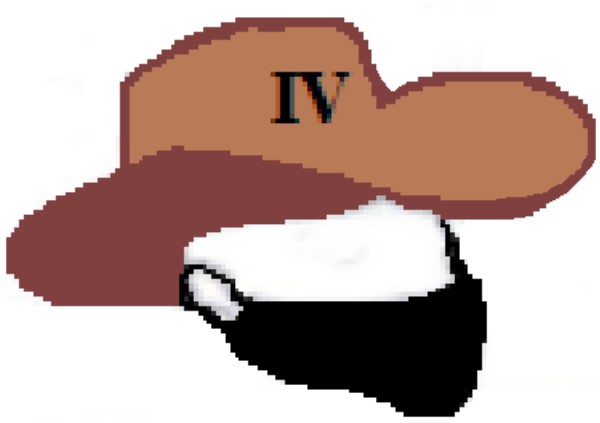
Lunch Roulette



Lunch Decider Mobile Application

Requirements Document

Team IV

Andrew Harrison

Ashton Webb

Dylan Miles

Keaton Whitaker

Sunil Rasaily

Table of Contents

1.0 Introduction

1.1 Purpose

The purpose of this document is to describe the requirements of the Lunch Decider mobile application which, is to take the ease off of indecisiveness. Choosing what you want to eat can be a difficult task. Are you hungry for multiple things? Or are you the type of person to just not want to make decisions? Luckily this app is here to take all that weight off your shoulders and choose for you.

1.2 Outline

The rest of this document will go into detail about the requirements such as functional requirements, non-functional requirements, major functional requirements, and minor functional requirements. Identifying and addressing risks will be described. Targeted Users will be identified. Lastly, the discussion of the project's Use Case Diagram.

2.0 Scope

Our client Dr. Lopamudra Roychoudhuri wants an application that will decide what restaurant the user should go to. The client would like for our team to create this app that allows the user to choose different filters based on the user's preferences. With this, the client would like the application to provide output that associates with the filters created. Lastly, if time permits the client would like for the team to develop a web application for computers.

3.0 Targeted Users

Are you unsure of where you want to eat and use an Android or own a computer? Then you're the application's targeted audience. Anyone who has trouble deciding where they want to eat can easily use the lunch decider application to figure it out for them. However, the customer wants the team to consider themselves as a representative of Yelp. A Use Case Diagram showing the relationship the user and server have with the application will be shown in Appendix 1.

4.0 Requirements

4.1 Functional Requirements

This section describes the major functions that will be performed by the application.

4.1.1 Spin the Wheel

Upon entering the application, the user will be able to automatically spin a wheel. This wheel will include the most popular restaurants/ user favorites. The user will also be able to re-spin as many times as they would like. The wheel will not always stay the same you will be able to add various filters (price range, mile radius, etc.). Nevertheless, upon initial entry into the app, you will have the ability to spin a completely random wheel. Ultimately deciding where the user will go for lunch.

4.1.2 Enable Location Services

Geographical data will be provided by the user. The user will be prompted when first opening the app to allow permission for their current location to be used. If the user selects “allow,” we will access their current location and populate into the wheel nearby restaurants, within the given mile radius. If the user selects “don’t allow” the app will ask for their current zip code and populate restaurants based on the zip code provided.

4.1.3 Search/Lookup Restaurant

Per the request of the client, there will be an option to search/look up restaurants in the area. The search/lookup option will be connected to Yelp. Users will be able to search for a restaurant and it takes them directly to the Yelp page of that restaurant, where they will be able to see reviews, menus, and directions and find out whether that restaurant offers delivery or not. The app should run smoothly with Yelp.

4.1.4 Add Favorites

Users will be able to add restaurants of their choosing to their favorites. This will not only make that restaurant's information easy to find but this leads to the implementation of a favorites wheel. The favorites wheel will include only the restaurants that the user has selected as favorites. Meaning that the user will be able to spin the wheel of only the restaurants that they really love and enjoy. Regardless of what the wheel lands on they will love it because it is a favorite of theirs.

4.1.5 Add Criteria

The users will be able to add many different filters/criteria to better narrow down what they want to be available on the wheel. Filters will include but will not be limited to price ranges, mile radius, how many stars the restaurant has, types of food, and much more.

4.2 Non-Functional Requirements

This section describes the minor functions that will be performed by the application.

4.2.1 Usability

- The app will be available on the Google Play Store for Android users.
- The app will have a web-based application if time permits.
- The interface will have friendly labels to reduce the learning curve.

4.2.2 Reliability

- The application will support up to 1000 users per day. However, with permission from Yelp through online request, could be upgraded to more than 10,000 users per day.
- The app should perform without failure at least 97% of the time.

4.2.3 Performance

- The app will only take up to 5 seconds to output results/ respond to the user input.
- The app will be compatible with all Android operating systems that have Android 5.0 or higher.
- The app will utilize the Android SDK, including its location API.
- The app will utilize the Yelp API.

4.2.4 Security

- App permissions: the app will not request many permissions and will ensure that users are aware of the permissions they are granting.
- The app will undergo regular updates and patches to ensure known vulnerabilities are fixed.
- The app will only use trusted up-to-date libraries or avoid third-party libraries that may have unknown vulnerabilities.
- The app will be regularly tested and audited to identify and fix any vulnerabilities.

Appendix 1.0 Use Case Diagram

Figure 1.0 represents our Use Case Diagram. This diagram shows the relationships between the user and our server, as well as how they can interact with Yelp.

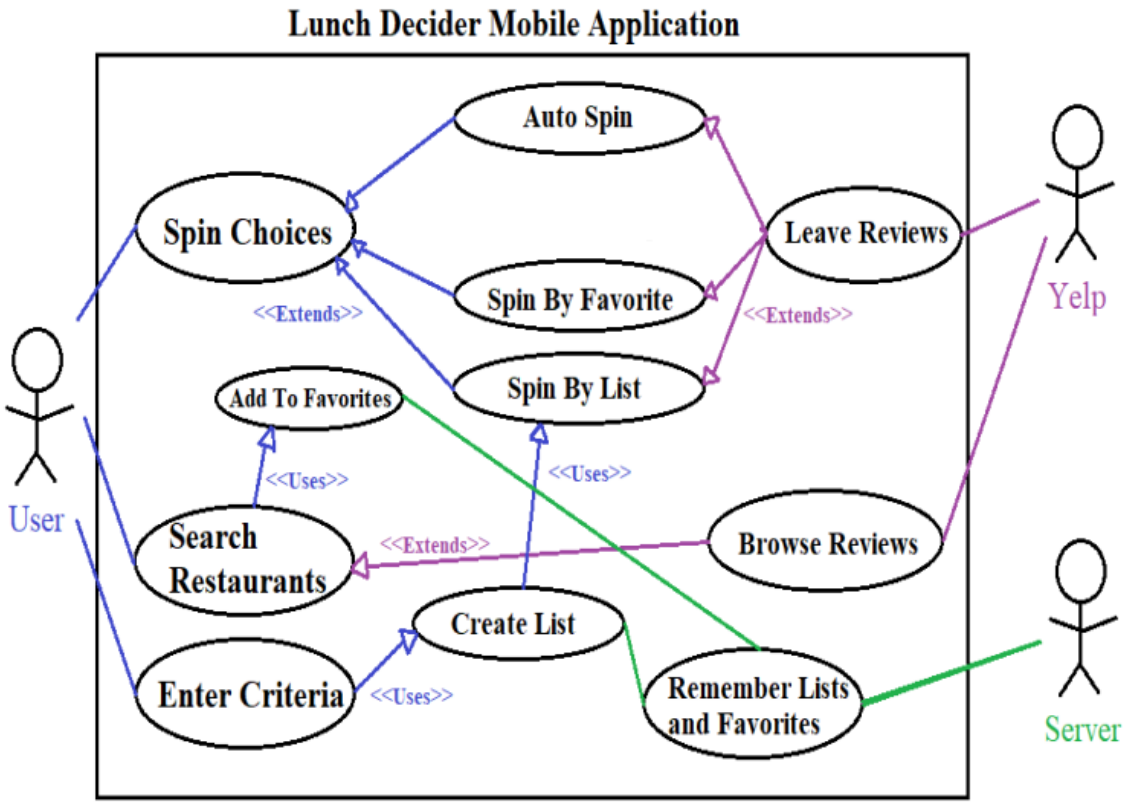


Figure 1.0 Use Case Diagram