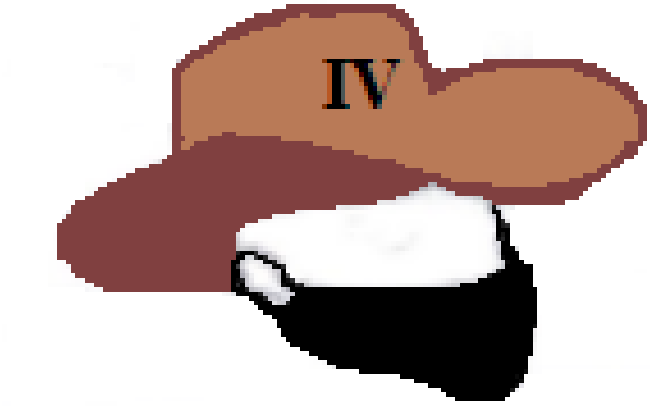
# What's For Lunch?




# Lunch Decider Mobile Application

## Final Project Plan

Andrew Harrison

Ashton Webb

Dylan Miles

Keaton Whitaker

Sunil Rasaily

02/23/2023

# Table of Contents

# 1.0 Introduction

## A. Purpose

The purpose of this document is to go over the overall final plan of the Lunch Decider Application. Requirements such as functional and non-functional requirements will be discussed. This document will identify the team's methodology that will be used to complete this application. The people that are working on the project as well as their roles will be presented. The team will identify any potential risks there are as well as how to address them. A breakdown of the schedule with a Gantt Chart will be provided.

## B. Objectives

The team's objective for this project is to have three prototypes to present to the customer. The third prototype will be the final prototype and will be used as the official application. The prototypes will include a user-friendly interface allowing the user to easily spin an option, search for a restaurant, create criteria, and more. The database that will be used will come from Yelp. There will also be a link to Yelp from the output choices given. The project should take only a semester to be developed.

# 2.0 Requirements

## A. Functional Requirements

### i. 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 the user 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.

### ii. 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", the application 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.

### iii. 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 the users 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.

### iv. 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 the users really love and enjoy. Regardless of what the wheel lands on the user will love it because it is a favorite of theirs.

### v. Add Criteria

The users will be able to add many different filters/criteria to better narrow down what the users 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.

## B. Non-Functional Requirements

### i. Usability

The app will be available on the Google Play Store for Android users.

If time permits, it will also be made web-based/ available to access via the web.

The interface will have friendly labels to reduce the learning curve.

### ii. Performance

The app will only take up to 5 seconds to output results/ respond to user input.

The app will be compatible with all recent android models.

The app will utilize the Android SDK, including its location API.

The app will utilize the Yelp API.

### iii. Reliability

The application must always be available every day during peak lunch hours of 11 am – 1 pm.

The application should be able to support at least 10,000 users at the same time while maintaining optimal performance.

The app should perform without failure 97% of the time.

The time it takes to restore the system if there is a failure should be no more than 10 minutes.

### iv. Security

App permissions: the app will not request many permissions and will ensure that users are aware of the permissions that the user's 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.

# 3.0 Methodology

The methodology that is being used for this project is an iterative prototype method. The team will have multiple prototypes that are split up into multiple parts. Once one piece of the prototype is complete, the team will move on to the next one until the whole prototype is developed. After the prototype has been developed the team will present it to the customer to get feedback. Once feedback is obtained, the team will start on the second prototype and repeat the same process.

# 4.0 Project Resources

## A. People

The team members working on this project each bring some experience:

One team member brings heavy knowledge of app development and Java.

One team member brings organizational skills for keeping up with the project.

One team member brings good communication skills for presenting.

One team member brings writing skills for developing the final report and other technical writing.

One team member brings good research skills for looking up any databases or figures the team may need.

Many of these experiences were linked to each member. Many tasks that used these experiences were due to a joint effort.

## B. Hardware and Software

The software that will be used is GitHub, which will be used for keeping all of our work. Microsoft Project will be used to make a timeline Gantt Chart. For developing the application, the team will use android studio to make working code. For the hardware, the team will use any type of android device that is up to date with the current version of Android 12. This will be used to test the app to make sure it works properly. The computer needed to complete this project is any windows system that runs a 64-bit version.

# 5.0 Risks

Table 1.0 shows a list of possible risks that could happen during the project. These possible risks are discussed more fully in the next section.

**Table 1.0**

| Risks | Likelihood | Mitigation |
|---|---|---|
| Lack of Communication | Medium | Meetings, Discord, Slack |
| Sickness | Medium – High | Online meetings, Backup members |
| Time Management | Low – Medium | Detailed Schedule |

## A. Identifying Risks

Many risks are possible when working on a project. One such potential risk is a lack of communication. This can lead to project failure when there is not enough communication between team members among each other and communication between the team members and customer.

Another potential risk could be if a team member gets sick. Covid-19 and other viruses can really hurt the team by putting a team member out of commission especially if that team member has a big piece of the project to do.

Time management can play a risk during this project. Many of the team members have other classes on top of this one. Some students also have part-time or full-time jobs outside of college. These issues can mess things up such as scheduling issues and communication issues.

## B. Addressing Risks

Addressing the potential risks is needed so that you can have a good backup plan. Lack of communication hopefully shouldn't be an issue for the team. The team will have a regular meeting every Wednesday and have a full detailed schedule for the semester. All of the team's tasks are divided up and backups are assigned as well.

When addressing sickness among team members, the team was able to come up with a quick solution. Almost 90% of the meeting taking place this semester happens on discord via voice chat. If a member were to fall ill, the member would still be able to attend normal meetings. The team will also have backups for each activity that is in progress, so if a team member falls ill someone else will be able to pick up the slack.

To face time management issues, the team has a few options. Luckily, most of the members of the team have taken a time management course in the past. The team already has a detailed schedule for the project. As long as most team members can keep up with the schedule then there should be no issues.

# Appendix 1 Schedule

Person Assigned | Backup

| Description | Date | Members Assigned |
|---|---|---|
| *Team Meeting #1* | 01/25/2023 | AH, AW, DM, KW, SR |
| *Customer Meeting #1* | 02/06/2023 | AH, DM |
| *Team Meeting #2* | 02/07/2023 | AH, AW, DM, KW, SR |
| *App Design Sketches/Logo* | 02/07/2023 – 02/21/2023 | AH, AW, DM, KW, SR |
| *Schedule Draft* | 02/10/2023 – 02/17/2023 | AH \| KW |
| *Team Meeting #3* | 02/15/2023 | AH, AW, DM, KW, SR |
| *Project Design* | 02/15/2023 – 03/24/2023 | AH, AW, DM, KW, SR |
| *Final Project Plan* | 02/15/2023 – 02/17/2023 | AH \| KW |
| *UID Displays Draft* | 02/07/2023 – 02/17/2023 | KW \| AH, DM |
| *Requirements Document* | 02/15/2023 – 02/17/2023 | AW, AH\| SR |
| *Use Case Diagram* | 02/15/2023 – 02/17/2023 | AH \| KW, SR |
| *User Display Coded* | 02/15/2023 – 02/24/2023 | DM \| AW, KW |
| *Customer Meeting #2* | 02/22/2023 | AH, DM, KW |
| *Team Meeting #4* | 02/22/2023 | AH, AW, DM, KW, SR |
| *Final Report* | 02/22/2023 – 04/21/2023 | AH, AW, SR \| DM, KW |
| *Prototype Part 1.1* | 02/22/2023 – 03/01/2023 | DM, KW, AW \| AH, SR |
| *Team Meeting #5* | 03/01/2023 | AH, AW, DM, KW, SR |
| *Prototype Part 1.2* | 03/01/2023 – 03/08/2023 | DM, KW, AW \| AH, SR |
| *Class Diagram* | 03/01/2023 – 03/08/2023 | AH, SR \| KW, AW |
| *User Manual Draft* | 03/06/2023 – 03/24/2023 | AH, AW \| KW, SR |
| *Customer Meeting #3* | 03/08/2023 | AH, DM, AW |
| *Team Meeting #6* | 03/08/2023 | AH, AW, DM, KW, SR |
| *Prototype Part 1.3* | 03/08/2023 – 03/15/2023 | DM, KW, AW \| AH, SR |
| *Testing Prototype 1* | 03/08/2023 – 03/15/2023 | AH, DM, KW |
| *Customer Meeting #4* | 03/15/2023 | AH, DM, SR |
| *Team Meeting #7* | 03/15/2023 | AH, AW, DM, KW, SR |
| *Prototype Part 2.1* | 03/15/2023 – 03/22/2023 | DM, KW, AW \| AH, SR |
| *Navigation Diagram* | 03/15/2023 – 03/29/2023 | AH, SR \| AW, KW |
| *Team Meeting #8* | 03/22/2023 | AH, AW, DM, KW, SR |
| *Prototype Part 2.2* | 03/22/2023 – 03/29/2023 | DM, KW, AW \| AH, SR |
| *Customer Meeting #5* | 03/29/2023 | AH, DM, KW |
| *Team Meeting #9* | 03/29/2023 | AH, AW, DM, KW, SR |
| *Prototype Part 2.3* | 03/29/2023 – 04/05/2023 | DM, KW, AW \| AH, SR |
| *Testing Prototype 2* | 03/29/2023 – 04/05/2023 | AH, DM, KW |
| *Customer Meeting #6* | 04/05/2023 | AH, AW, DM, KW, SR |
| *Team Meeting #10* | 04/05/2023 | AH, AW, DM, KW, SR |
| *Test Plan and Tests Designed* | 04/05/2023 – 04/07/2023 | DM, KW |
| *Final Prototype* | 04/05/2023 – 04/19/2023 | DM, KW, AW \| AH, SR |
| *Final Tests* | 04/05/2023 – 04/19/2023 | AH, DM, KW |
| *User Manual Report* | 04/05/2023 – 04/19/2023 | AH, SR \| KW |
| *Final Presentation* | 04/05/2023 – 05/01/2023 | AH, AW, DM, KW, SR |
| *Deliver* | 04/26/2023 | AH, AW, DM, KW, SR |

# Appendix 2 Gantt Chart


**Figure 1.0 Gantt Chart**

# 6.0 References

[Figure 1.0] Microsoft.com. Available: https://www.microsoft.com/en-us/microsoft-365/project/project-management-software [Accessed: February 14, 2023]