Integrated Project Plan

|  |
| --- |
| Project Name: Food For Friends |
| Date: 1 November 2019 |
| Project Manager: David Batdorf |

# 1.0 Introduction

This project is designed to tackle two large issues that face society today, by bringing people together. The first issue is food waste. According to the USDA somewhere between 30 and 40 percent of the US food supply is lost to waste. The second issue is hunger, which according to USDA 11.1% of Americans lacked enough food to sustain themselves at some point last year. That means over 36 million Americans lack access to an adequate and stable food supply.

We at KISS are developing the “Food For Friends” web application to coordinate individuals to combat these societal issues. The application will use geolocation services to pair individuals, givers, with those who are in need of a food, [recipient](https://www.thesaurus.com/browse/recipient)s. Users will be required to create an account, with authentication provided by Google. Once a user is authenticated, they can set the app to either giver, or recipient mode. Giver mode will allow the user to post available food to users in the [recipient](https://www.thesaurus.com/browse/recipient) mode. Recipient mode allows users to view a local map with pins denoting the recently posted food offerings.

This document is designed to clearly communicate with both the project team and the stakeholders. This document will outline how to communicate with the project team, and provide a development timeline. The document will also define the project risk, along with change procedures and control procedures for cost, and time.

# 2.0 Project Contacts and Roles

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Organization/Department** | **Phone/E-mail Address** |
| Project Manager | David Batdorf | Project Management | afdave5124@gmail.com |
| User Interaction | Andrew Delgado | Requirements | andrewdelgado88@gmail.com |
| Technical Writer | Ray Gutierrez | Requirements | raygoot@outlook.com |
| Software Engineer | David Markowski | Development | markowskidavid77@gmail.com |
| Integration Engineer | Marcus Jones | Development | marcus.njones2713@gmail.com |
| Test Engineer | James Boehm | Test | boehm22@hotmail.com |

# 

# 3.0 Communications Management

This table lists the different communication items needed for this project. A communication item may be a Word document, an e-mail, a meeting, or anything called for by another plan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **What** | **When** | **How** | **Responsible** | **Audience** |
| *Project Kickoff* | *24 October 2019* | *Virtual Meeting* | *Project Manager* | *Request comments* |
| *Team Progress Sync* | *Weekly* | *Virtual Meeting* | *Project Manager* | *Update Management on deliverable progress* |
| *Team Deliverable Review* | *Weekly* | *Virtual Meeting with Github collaboration* | *Project Manager* | *Review and critique weekly deliverable.* |

# 4.0 Risk Management

**4.1 Risk Management Strategy**

* Risks will be identified both initially during the planning, and analysis phases of the development lifecycle. Secondly risk will be identified during weekly team syncs, where project management will review weekly work plans.
* The risks associated with this project vary, both in the type and severity of the risk. First the risk exists to users when they meet strangers via the application. The users also have risk from potentially malicious users, such as poisoned food. Risk exists when the application interfaces with outside modules, such as geolocation, authentication and database interface. Risk also exists to the database that contains user data, and current food posts since the database must accept user inputs. Finally, risk exists in the inability to test the full variety of commercially available computing devices.
* The risk associated with the large variety of commercial computers is one that simply must be accepted due to budgetary constraints. The risk to users, both in meeting strangers and accepting food items will be mitigated. The risk associated with external interfaces will be mitigated, and remaining risk, while minimal must be accepted. The risk to the database will be full mitigated to prevent external tampering.

**4.2 Mitigation and Contingency Plans**

To mitigate the risk to users via meeting and accepting food from strangers, a rating system will be implemented. This system will allow users to not only rate the person they interact with, but they can also leave feedback. The risk via external dependencies will be mitigated by choosing a reliable source. The risk associated with database access will be fully mitigated by implemented extremely stringent data input parsing, to detect and

# 5.0 Change Management

The project will follow the standard Software Development Life Cycle change management convention. Any changes made to the project will also require a full team code review, before any changes are committed to the code base.

# 6.0 Time and Cost Management

* The most important aspect to cost control is limiting product changes during deployment.
* Time and costs will be controlled by defining clear, and concise requirements.
* Any changes to requirements require Project Manager approval.
* In the advent of required changes, additional personal will be moved from other roles to ensure a timely development schedule.

**6.1 Sprint schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Cycle | Deliverable | Date Range | Assignees |
| 1 | GIT setup  Trello setup | October 21 –  October 27 | Project Manager |
| Project Direction  Plan | All Team Members |
| 2 | [Project Plan](https://learn.umuc.edu/d2l/lms/dropbox/user/folder_submit_files.d2l?db=830386&grpid=0&isprv=0&bp=0&ou=415364) | October 28 –  November 3 | Project Manager |
| 3 | Peer Review #1 | November 4 –  November 10 | All Team Members |
| Test Plan | Test Engineer  Technical Writer |
| 4 | Project Design | November 11 –  November 17 | Project Manager  Technical Writer  Integration Engineer  UX/HCI |
| 5 | Peer Review #2 | November 18 –  November 24 | All Team Members |
| Phase #1 Source | Project Manager  UX/HCI  Software Engineer  Integration Engineer |
| 6 | Phase #2 Source | November 25 –  December 1 | Project Manager  UX/HCI  Software Engineer  Integration Engineer |
| 7 | Phase #3 Source | December 2 –  December 8 | Project Manager  UX/HCI  Software Engineer  Integration Engineer |
| 8 | Peer Review #3 | December 9 –  December 15 | All Team Members |
| Finial Product  Submission |

# 7.0 Quality Management Plan

Software quality assurance is crucial to the software development lifecycle and must exist during every step of the lifecycle. Deliverable reviews, and project management will be directly part of each step of the development processes to ensure that specifications are followed and potential errors are minimized

# 8.0 Procurement Management

All external procurements will be vetted by program management, and assessed for stability, security, and cost viability compared to market competitors. External procurement procedures will be used not only for physical procurements such as database servers, but also for digital procurements such as external code libraries.

# 9.0 Scope Management

Managing the scope of a project is integral to ensuring that time and costs are kept in check. The requirements for the project fall into two categories. First is the requirements for the application that define what it will do. The second category is the requirements that the user must adhere to, in order to be able to use the application

Each deliverable will be subject to a full team review 3 days prior to submission to ensure that it meets all requirements for the deliverable, without exceeding the scope of the deliverable. Requirements for each deliverable will be fully defined before any part of the creation of that deliverable begins. For the application itself, a full user guild, and test plan will be created before the application is coded. This is done to ensure that the final product meets the initial design fully. This also prevents requirements creep, by limiting new features to only those that will be tested.

In order to reach as broad of a user base as possible the technical requirements to run the web application are very general, and the team will be using code that is as portable as possible. For a user to be able to use the web application “Food For Friends” they must have a computer, or smart phone, capable of running the latest version of either FireFox, Chrome, Edge, or Safari. Users must have a google account for authentication. Finally, users must allow the Web Application to access their current location. Users will have some limitation on how they may use the application to ensure that it is only used to give away unused food to those in need. All user entries will be based off drop down menus, not user text entry. This design decision was made to ensure that only intended items may be posted on the application.

# 10.0 Data Flow and Sample Interactions

//add arrows

User Authenticated Using Google Account

Receiver Mode displays your local map, with pins indicating available food

Giver Mode displays Options related to posting an available food offering

User sees the splash screen allowing for user type selection (Giver or Receiver mode)

Receiver Mode users have the following options.

1. Set View Radius

(Drop Down Menu)

1. Select Food Offerings to

receive the offerings location

1. Sign Out
2. Change to Giver Mode

Giver Mode users have the following options.

1. Post Food Item
2. Sign Out
3. Change to Receiver Mode
4. View Users Posts
5. Sign Out

Posting Food displays the following fields.

1. Food description

(Drop Down Menu)

1. Number of servings available

(Drop Down Menu)

1. Food availability window

(Drop Down Menu)

1. Radius of availability

(Drop Down Menu)

Sample User Interaction.

1. Kate signs into the web application “Food For Friends”
2. Kate selected Giver Mode
3. Kate Selects Post Food Item
4. Kate Enters the following information
   1. Food Description: Meat Free Chili
   2. Number of Servings Available: 5 (from drop down menu)
   3. Food availability window: 3 hours (from drop down menu)
   4. Radius of availability: 10 miles (from drop down menu)
5. Kate clicks the post button
6. John signs into the application
7. John selects Receiver Mode
8. John sees Kates nearby post as he is located 2 miles away.
9. John selects Kates post via a pin on his local map.
10. John receives the location of Kates offering and is able to input the location into a mapping application of his choice
11. Internally the application links John and Kate, allowing them to rate the interaction.

Recourses

Food Waste FAQs. (n.d.). Retrieved October 31, 2019, from https://www.usda.gov/foodwaste/faqs.

Frequency of Food Insecurity. (n.d.). Retrieved October 31, 2019, from

https://www.ers.usda.gov/topics/food-nutrition-assistance/food-security-in-the-us/frequency-of-food-insecurity/.