LifTOvers Web App

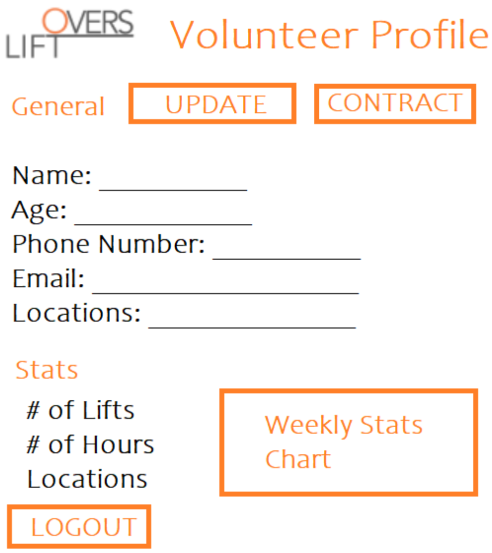
Product Details

Q1: What are you planning to build?

We are planning to build a web application for administrators to manage the process of connecting leftover food from donors to community food providers, and for volunteers to log on and view their volunteer history. This process is a lift request, which will be managed by the admins that use our web application. Sample use cases include: admins being able to create lift requests, track the lifecycle of the lift request, add other admins, add other volunteers, add food providers, add food donors, and edit/view/delete all information.

Besides admin, volunteers will be able to log in and access relevant information such as number of lifts completed and number of hours. This may be useful to them to keep track of their personal stats for personal goals. The following diagram is not the complete design but represents the information the volunteer will be able to see.

Currently, the lifTOvers website only contains forms for food donors to submit lift requests and volunteers to sign up to get involved, but the application we are developing will enable lifTOvers to manage and monitor all information collected. The lifTOvers team also mentioned in a meeting that lift requests on their current website could also be diverted straight to the web application we are building.



Volunteer Profile

General

Name: ___________
Age: _____________
Phone Number: ___________
Email: ____________________
Locations: ________________

Stats

# of Lifts
# of Hours
Locations

Q2: Who are your target users?

The following are three user personas that reflect the three types of users that will be engaging with the web application we will be building:

Name: Anna Smith   Age: 25

Occupation: Administrator   Technical Proficiency: Intermediate

About: Anna currently works as an administrator at lifTOvers. She meets with companies to partner with them to reduce food waste from their meetings and events, and she meets with donors and food providers where lifts can potentially be sent to. Anna also volunteers in her free time.

Expectations: Anna wants to be able to create and modify lifts, manage information about food providers donors, as well as see all activity that takes place on the site.

Name: Steve Jones   Age: 21

Occupation: Student   Technical Proficiency: Intermediate

About: Steve is currently a fourth-year student studying Kinesiology at the University of Toronto. He is passionate about reducing food waste, which is why he volunteers for lifTOvers in his free time as well as during breaks at school.

Expectations: Steve wants to be able to submit photos of his lift after he picks up food. He is also hoping for a way to see his total number of lifts as well as the total number of volunteer hours to meet a personal goal.

Name: Cathy Williams   Age: 35

Occupation: Cofounder   Technical Proficiency: Moderate

About: Cathy is a cofounder of lifTOvers. She manages the activities of the entire company and monitors other admin.

Expectations: Cathy wants to be able to manage lift requests as well as have the ability to add or delete admins as a new role of super admin on our web application.

Q3: Why would your users choose your product? What are they using today to solve their problem/need?

Food waste represents a serious problem in Canada. In Toronto alone, the municipal government reports that over 50% of food wasted is avoidable and can be repurposed. To bridge the gap between excess food that is still fresh and those in the community who are in need of food, we are building a product for lifTOvers where administrators can oversee the matching process and ensure the repurposement of food. It also provides volunteers in the community with the unique experience and opportunity of a self-assigned commitment. By registering a primary (and possibly secondary) postal code as well as availabilities, the volunteer is able to determine their own hours and level of commitment. This provides volunteers with flexibility, as well as an ongoing opportunity to help their immediate local community. For managers, it provides an overview of all ongoing members involved in the lifecycle of a lift request and information about the involved parties.

There are other organizations (such as FoodRescue, etc.) that share lifTOvers' ideals of rescuing leftover food, but lifTOvers is singular in that it capitalizes on the community and allows volunteers to choose their own hours in the immediate local community of their permanent adderss. lifTOvers' mission is to re-distribute food to community providers from donors by utilizing local volunteers and by creating an administrative system, information for this process can be managed both efficiently and effectively while volunteers can gain a greater sense of involvement and impact by accessing their profile.

Q4: How will you build it?

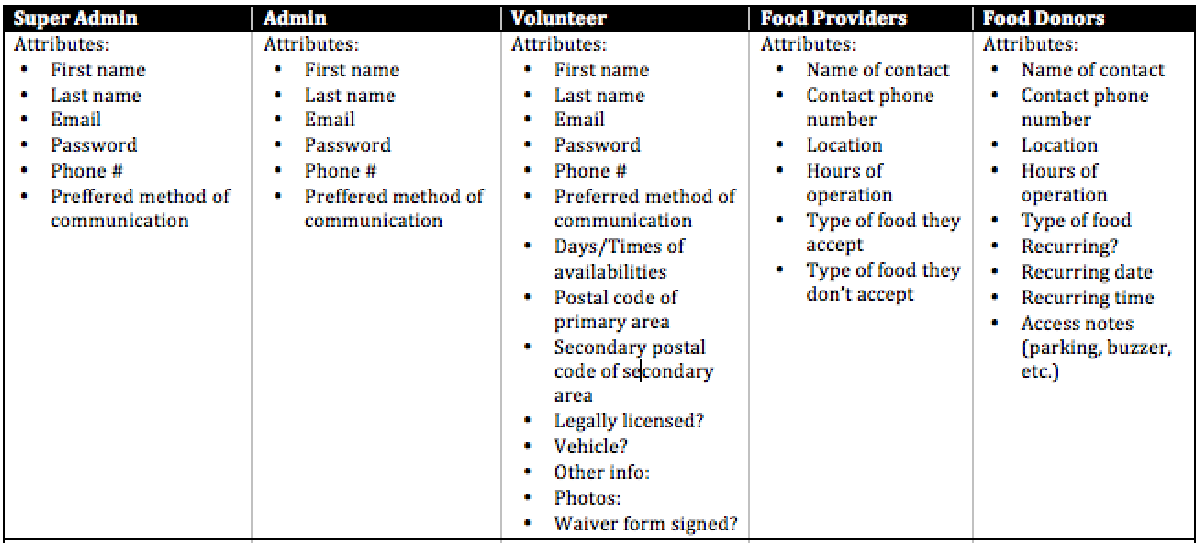
The current technology stack that lifTOvers is using is NodeJS, React, and Twilio (for SMS message management) using a Single Threaded Event Loop architecture, as per NodeJS standards. We chose to develop our web application using these as well. Even though none of us have extensive experience with Javascript, we are eager to learn more and Abu, the tech lead has offered his expertise to help us get started. For user authentication, the package we will be using is the NPM encryption package, suggested to us by the lifTOvers team. Other packages we will be using will be from NPM, and we will also be using Google Maps API to configure distances between pickups and volunteer locations.

For deployment, we will be using Heroku, as lifTOvers is currently doing. Starter code will be provided to us from Abu, and we will be integrating the web application with the REST API from their current website.

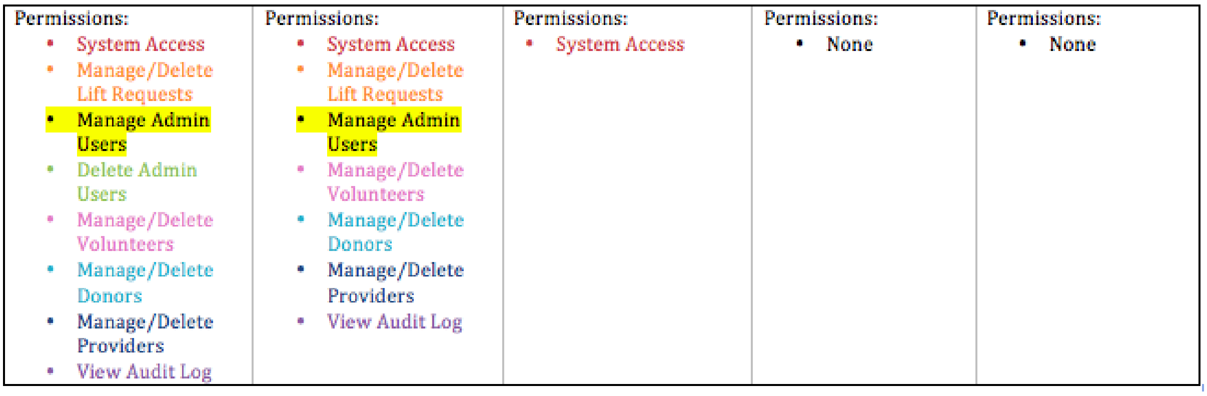
Our testing strategy involves unit tests based on coverage %, and each group member is expected to write tests when working on functional aspects of the code. We also have a group member in charge of enforcing this policy when pull requests are made. Testing UI will be done by demo-ing features to each other, and then to the lifTOvers team for approval.

Below are diagrams of various development categories. We created these based on a specification document provided to us from lifTOvers.

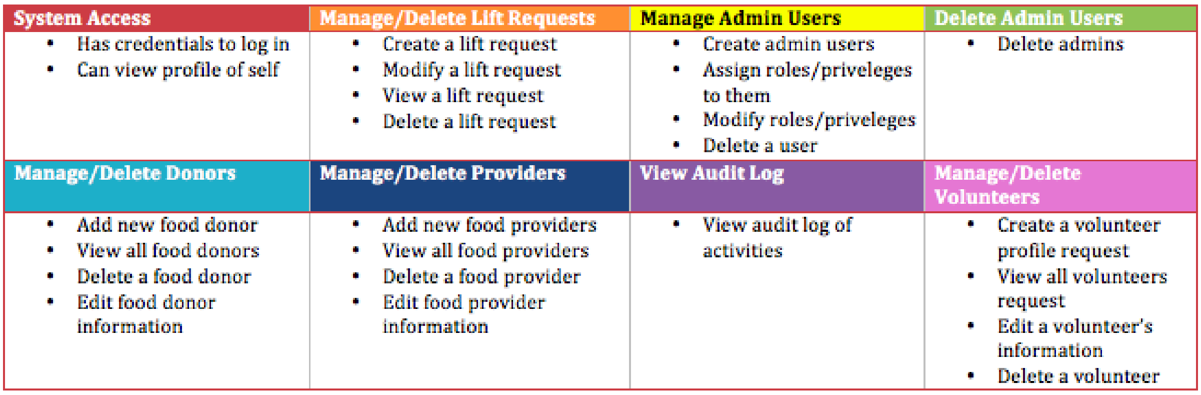
Table of User objects, their attributes and permissions (color-coded).



Super Admin	Admin	Volunteer	Food Providers	Food Donors
Attributes: • First name • Last name • Email • Password • Phone # • Preffered method of communication	Attributes: • First name • Last name • Email • Password • Phone # • Preffered method of communication	Attributes: • First name • Last name • Email • Password • Phone # • Preferred method of communication • Days/Times of availabilities • Postal code of primary area • Secondary postal code of secondary area • Legally licensed? • Vehicle? • Other info: • Photos: • Waiver form signed?	Attributes: • Name of contact • Contact phone number • Location • Hours of operation • Type of food they accept • Type of food they don't accept	Attributes: • Name of contact • Contact phone number • Location • Hours of operation • Type of food • Recurring? • Recurring date • Recurring time • Access notes (parking, buzzer, etc.)



Permissions: • System Access • Manage/Delete Lift Requests • Manage Admin Users • Delete Admin Users • Manage/Delete Volunteers • Manage/Delete Donors • Manage/Delete Providers • View Audit Log	Permissions: • System Access • Manage/Delete Lift Requests • Manage Admin Users • Manage/Delete Volunteers • Manage/Delete Donors • Manage/Delete Providers • View Audit Log	Permissions: • System Access	Permissions: • None	Permissions: • None



System Access	Manage/Delete Lift Requests	Manage Admin Users	Delete Admin Users
• Has credentials to log in • Can view profile of self	• Create a lift request • Modify a lift request • View a lift request • Delete a lift request	• Create admin users • Assign roles/priveleges to them • Modify roles/priveleges • Delete a user	• Delete admins
Manage/Delete Donors	**Manage/Delete Providers**	**View Audit Log**	**Manage/Delete Volunteers**
• Add new food donor • View all food donors • Delete a food donor • Edit food donor information	• Add new food providers • View all food providers • Delete a food provider • Edit food provider information	• View audit log of activities	• Create a volunteer profile request • View all volunteers request • Edit a volunteer's information • Delete a volunteer

Lift Request object, its attributes and life-cycle:

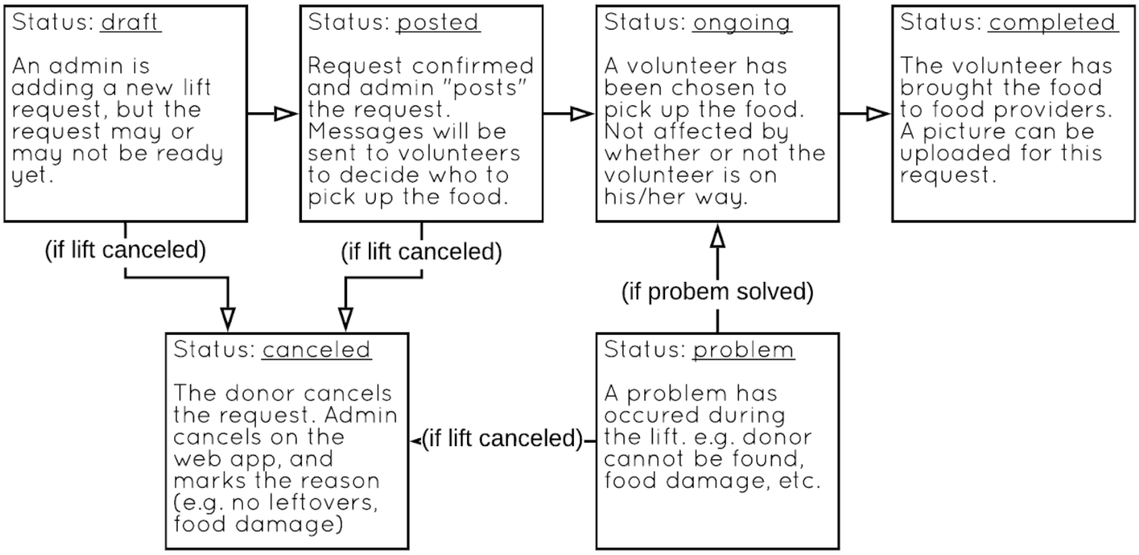
Lift Request (Object)
All attributes:

- Name (first and last)
- Email
- Telephone
- Preferred communication method (email or phone)
- Date of pickup (yyyy-mm-dd)
- Time food will be served (hh:mm)
- Time for pickup (hh:mm)
- Address for pickup
- Postal code
- Additional Details (not required)
- Description of food & anticipated amount of leftover
- Checked disclaimer
  (1. food is good for consumption
  2. food is prepared in restaurant/catering company)

[The above information are filled in by doners on the current lifTOvers website.]
[Attribtues below are for programming use:]

- status (of this request i.e draft/posted/ ongoing/completed/canceled/problem)
- notes (if the lift is canceled, for what reason?)
- photo (of the food, which later can be uploaded by the volunteer)

Lift Request states:



Status: draft

An admin is adding a new lift request, but the request may or may not be ready yet.

Status: posted

Request confirmed and admin "posts" the request. Messages will be sent to volunteers to decide who to pick up the food.

Status: ongoing

A volunteer has been chosen to pick up the food. Not affected by whether or not the volunteer is on his/her way.

Status: completed

The volunteer has brought the food to food providers. A picture can be uploaded for this request.

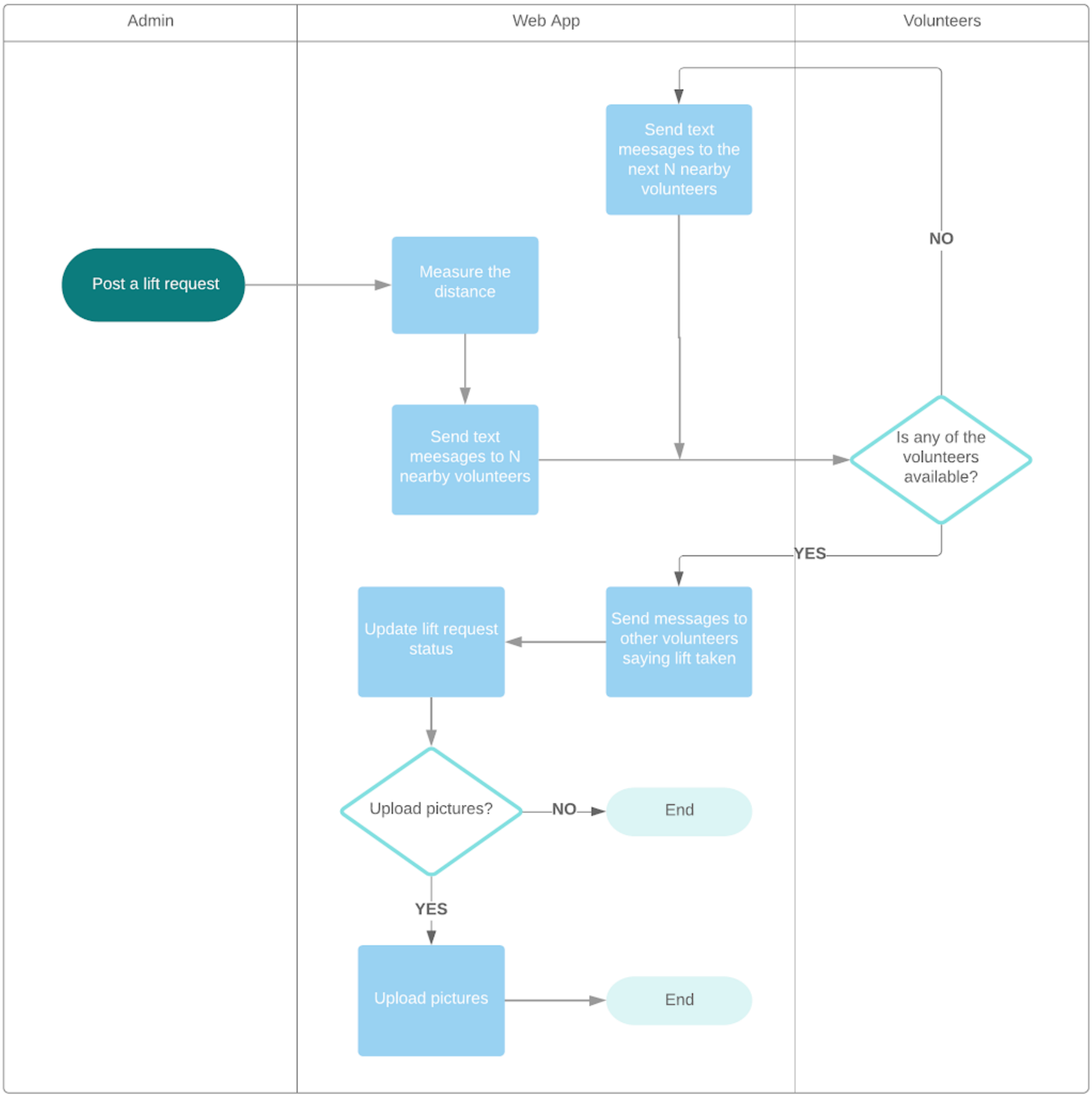
Status: canceled

The donor cancels the request. Admin cancels on the web app, and marks the reason (e.g. no leftovers, food damage)

Status: problem

A problem has occured during the lift. e.g. donor cannot be found, food damage, etc.

Action flow of the life-cycle of a lift request.


Admin
Web App
Volunteers
Post a lift request
Measure the distance
Send text meesages to N nearby volunteers
Send text meesages to the next N nearby volunteers
Is any of the volunteers available?
NO
YES
Send messages to other volunteers saying lift taken
Update lift request status
Upload pictures?
NO
End
YES
Upload pictures
End

Q5: What are the user stories that make up the MVP?

User Stories

The following are user stories provided to us from the lifTOvers team (document here: https://docs.google.com/document/d/16JjUwRTNOCfvuxBWofQcs9yl2AAQgdTfgqd4TVySxbA/edit?usp=sharing)

Liftover Admin

- As a liftover admin, I would like to log-in to the application so that I can manage lifts, volunteers and other users
  - Design Mockup of login page

        - Login page should contain email, password fields and a button showing “Sign in”
    - Convert mockup to React
        - The react component should look as close to the mockup as possible
    - Implement api endpoint to log user in and return a valid token
        - Endpoint should accept email, password and return a valid jwt token on success, or return an error payload
    - Integrate frontend with api
        - After a user clicks on the login button, the loader should display, showing that an api call is being made; On success, the user should be routed to a page inside the application, else an error notification is displayed to the user
- As a liftover admin, I would like to view a lifts page, so that i can see all the lifts and view statistics regarding all the lifts on the system
    - Design mockup of lifts page
        - The lifts page should show a table with the following columns: Name, email, telephone, date of pickup, time of pickup, address, postal code. There should be a search bar and a create button
    - Convert mockup to React
        - The React implementation should look as close to the mockup as possible
    - Implement api endpoint to return paginated list of lifts
        - The endpoint should return all the lifts on the system and ensure it is paginated
    - Integrate frontend with api
        - When the page is mounted, there should be a loader on the page as the api call is being made and on success, should return a table showing the lifts, or an error notification
- As a liftover admin, I would like to view a volunteer page, so that i can see all the volunteers on the system and view statistics regarding the volunteers
    - Design mockup of volunteer page
        - The Volunteer page should show a table with the following columns: Name, email, phone, primary postal code, secondary postal code, drivers license, vehicle. There should also be a search bar and a ‘create’ button to add a volunteer.
    - Convert mockup to React
        - The React implementation should look as close to the mockup as possible.
    - Implement api endpoint to return paginated list of volunteers
        - The endpoint should return all the volunteers on the system and ensure it is paginated
    - Integrate frontend with api
        - When the page is mounted, there should be a loader on the page as the api call is being made and on success, should return a table showing the volunteers, or an error notification

Public User

- As a public user, I would like to View the liftovers website so that I can learn more about the initiative
    - Design mockup of website
        - The Website should have a navigation menu with the following: liftovers logo, home, about us, volunteer, get in touch. There should be a section that explains why liftovers and the problem the initiative seeks to solve
    - Convert mockup to React
        - The React implementation should look as close to the mockup as possible.
- As a public user, I would like to register as a volunteer, so that I can pick up food and deliver to the required destinations

Volunteer

1. As a volunteer I would like to view a list of all my previous, and upcoming lifts so that I can better manage my time
2. As a volunteer, I would like to edit my personal information so that my most recent information is on the system

Process Details

Roles & responsibilities

We chose to have each team member in charge of an aspect of a project, and with that comes responsibilities to remind other team members about. All team members will be coding, and involved with every aspect, the following is simply a list of who is responsible for what aspect:

- UX/UI Design: Faye
- Front-end: Tosin
- Back-end: Angela
- Communication with lifTOvers/Meeting set-up: Stacey
- Scrum master: Ryan
- Github Project Management: Andries
- Testing: Fillis

Each member on the team is expected to code and contribute to each of the categories determined above and contribute to team discussions.

Faye:

In charge of making sure UX/UI Design is approved by lifTOvers and complies with user design principles, will be responsible for organizing the completion of design section from team members. Previously completed an internship at a research facility building a user portal web application.

List technical strengths and weaknesses:

1. Strengths: Python, Java, HTML/CSS, Android Studio, REST API
2. Weaknesses: C, Javascript, XML

Tosin

In charge of Front end completion and delegating front-end tasks. Responsible for organizing the completion of front end section with team members. Previously completed a PEY at a Development Firm for the Ministry of Education.

List technical strengths and weaknesses:

1. Strengths : Databases, Android Studio, html
2. Weaknesses: C, C++, Javascript

Angela

In charge of Back-end end completion and delegating back-end tasks. Responsible for organizing the completion of Back-end section with team members.

List technical strengths and weaknesses:

1. Strengths: Database, anything Java (Android Studio), Python also fine
2. Weaknesses: C, Javascript

Stacey

In charge of communication with LiftOvers team. Responsible for taking minutes and keeping track of LiftOvers requests, data and information.

List technical strengths and weaknesses:

1. Strengths: Java, Python, Android
2. Weaknesses: Javascript, XML, databases

Ryan

Will be the Scrum master, In charge of tracking project progress and calling meetings depending on needs for project. In charge or maintaining proper progress of project in accordance with course requirements.

List technical strengths and weaknesses:

1. Strengths: Java, Python, Android, C
2. Weaknesses: Java Script

Andries

Will be in charge of maintaining github synchronization. Also in charge of making sure github code is able to run smoothly. Tracking errors and Recording them. Will be in charge of testing code as well.

List technical strengths and weaknesses:

1. Strengths: Java, C, Android Studio, Python
2. Weaknesses: JavaScript

Fillis

Will be in charge of testing code, making sure everything works. Also responsible that all functionality is tested before merging to master branch.

List technical strengths and weaknesses:

1. Strengths: Java, Python, Database, Android Studio, C
2. Weaknesses: HTML, JavaScript, XML

Team Rules

Communication:
We created a group chat on messenger to schedule team meetings which will happen every Friday at 3-5pm. If team members have questions about code, instead of using messenger we will be using Slack. We are also using Slack to communicate with lifTOvers for technical and functional questions. In order to come into contact with lifTOvers to set up meetings, Stacey is responsible for sending emails to them and corresponding.

Meetings:
Set up an Agile Project on Github to keep track of tasks certain team members are responsible for and their progress. If a team member is struggling with their assigned task and notifies this problem to the team early, we can assign another team member to either assist or swap tasks with each other, so we meet our goals for the week. During meetings we will also have standups facilitated by the scrum master to report what we have completed and plan for the next sprint. Sprints will be 2 weeks in length. Team members are expected to attend weekly team meetings on Fridays from 3pm-5pm, as there are no class conflicts then. If there is an emergency and

someone is unable to make it to the meeting, that person is expected to communicate their absence as soon as possible and prior to the meeting (by 12 hours if possible). Members will rotate taking notes during meetings for future reference, and if a member was unable to make it they are responsible for reviewing the meeting notes.

Conflict Resolutions:

- Non-responsive team members
    - If a team member remains unresponsive after 24 hours, as a team we will re-delegate tasks to someone else so we won’t fall further behind
    - When the non-responsive team member becomes available, we are going to have a serious team discussion
        - If they are consistently non-response, we will inform a TA/Instructor about this
- Personality Conflicts
    - If team members have an issue with each other, they can talk to a neutral third person to help solve the issues they have and learn to work with each other
- Scared to ask for help
    - Let everyone know on the team that its ok to ask for help and that everyone is willing to help each other out.

Events

- Meetings with lifTOvers tech group:
    - 1st: Oct 7th @ 6:30 p.m., on Google Hangouts
      Purpose: Introduce ourselves, and get to know more about the details of the project, i.e. requirements & what has already been done.
    - 2nd: Oct 12th @ 11:30 a.m., in person @ OISE
      Purpose: Discuss further on project details (as we have just received the document sent from out partner), as well as discussing on the D1 draft.
      Post - meeting: Join Slack, receive data structure & done code from partner.
    - Future meetings will reoccur Fridays 6-7 p.m. on Google Hangouts. (After our group meetings) on a biweekly basis. We may have meetings in person depending on the situation.
      Purpose: let our partner know how far are we on the project, and knowing any comments & anything we should add/change. Also for demo-ing progress to lifTOvers for approval.
- Group meetings (despite tutorial times, and after project was chosen):

    - 1st: Oct 10th from 4-5 p.m., @ BA 3175
      Purpose: discuss & split up work for D1
    - 2nd: Oct 12th @ 12:30 p.m., before & after the partner meeting, @ OISE
      Purpose: refine D1
    - Future meeting dates are currently set to Fridays 3-5 p.m. every week, and we are meeting in person at Bahen Center, hopefully room 3200.
        - Purpose mainly for a brief code review and stand-up, learning how everyone is doing, and depending on different situations, work distribution might have a small change (e.g. someone has midterms/ other assignments due all week and therefore not able to fully complete his/her part of the coding)
        - Also organize and report progress to lifTOvers if we are meeting with them remotely afterwards.
        - Larger tasks will be divided up, and once one large task is done, the meeting will also focus on ensuring every part is working as intended after merging & combining.
    - Group coding sessions will be held on a case-by-case basis. If a few people are working together on a feature they are expected to coordinate a coding session together. For tasks that involve integration between multiple members' code, an in person p

Artifacts

The main project management tool we will be utilizing is Github's Project Management.

We chose this because it is located where the repository is and is convenient and cohesive for keeping all the information together.

- Keep a detailed log of important tasks that need to be completed for each project milestone
- Tasks will be assigned 'points' which reflect the perceived difficulty of the task, and we will aim to complete a certain amount of points per spring, as per Agile.
- Each team member will be assigned tasks based on what we perceive to be the required skill set for the given task
    - The teammates more apt to working with that set of skills will complete that task
    - If a task requires additional work, or if help is needed, find a team member with similar strengths to assist in the work
    - A task of lower priority may be postponed if more important tasks are not completed, but NOT the inverse

- Tasks prioritized based on the following logic (along with team consensus):
    - Tasks that allow the program to function (Highest priority)
    - Tasks that would give some practical purpose to the program, according to the users' perspective
    - Tasks that would make the program convenient for users
    - Tasks that would improve user experience beyond a fundamental level (Lowest priority)

Highlights:

The tech stack decision:

Since none of the team members have extensive experience in Javascript, we were considering using Python and creating the web application using the web framework Django or Flask. Django because one of the team members have professional experience in it, Flask because we wanted to explore it further after CSC301 Assignment 1. In the end, we decided to continue using NodeJS and React as Abu has been doing so far. This decision was made because the whole team is open to learning something new in this project to maximize our learning, and also because Abu kindly offered to provide us with starter code and examples to guide us in development. Additionally, the current website lifTOvers operates with is built with NodeJS and React, so integration would be simpler.

The algorithm challenge**:**

As computer science students, we were interested in being challenged algorithmically. At our second meeting at OISE library, Abu stated that we would be given the challenge of creating an algorithm to calculate the closest *n* number of volunteers to the postal code of the food donor's location. This is a very interesting problem to us, as after taking so many algorithm courses, we are thrilled to have a problem to apply our theoretical knowledge. We will be implementing this solution using Google maps API, as a few of us have experience using Google's API.

The project management board**:**

We browsed through multiple software project management platforms, some of which include: Zenhub and Trello. In the end it was mentioned that since we were all students, we had access to Github Student Pro and therefore could access Github's project management tab. We decided to use Github's tool, since it would keep everything cohesively on the same website. The decision to not use anything physical (such as sticky notes) was made with this same thought, that information may be repeated or lost if it was on a physical medium. In the spirit of cohesiveness, we are keeping all code and planning on Github. For communication, we decided to use the same philosophy and keep meeting details on Facebook messenger, and technical details on Slack.