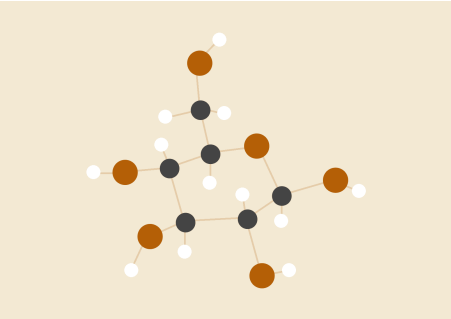
# Group 5 - Endurance

**CEN 4010-Principles of Software Engineering**

**Milestone 3- More Detailed Requirements, Architecture and a Vertical Software Prototype**




**Andy Cheng-** wcheng3@fau.edu

**Andrea Malvezzi-** amalvezzi2019@fau.edu

**Georges Bourques-** gbourque2015@fau.edu

**Noa Abiri-** noaabiri@protonmail.com

**Ellie Moffett-**amoffett2018@fau.edu


| Revision: | Date: |
|---|---|
| M3 | 7/ 6/ 2021 |
| | |
| | |
| | |
| | |

## Executive Summary (GB)

For those who like to spend time outdoors and give back to the community, there aren't many options for combining charity, time with family and friends, and enjoying good weather. With beneCycle, users will have it all. Whenever a charity event is held, the application will bring people together and for some, give them the chance to push themselves with several routes that offer different levels of donations.

beneCycle is a web-based application that gives users the ability to participate in local cycling or pedestrian events. Cyclists also have the opportunity to help support charities that are within their community. They will have several options of set paths that they may choose from. Each path offers a challenge in itself and comes with a higher donation. The application offers an intuitive interface, and many features that make the outdoors more exciting.

The application's purpose is to give its users as many planning options as possible. With the tap of a finger, streets are highlighted and the route tailored for the individual's level of riding is selected. Share and comment on these routes with your family and friends. Compare distances, waypoints, times, and goals. Sync up routes for group runs or rides. Design your avatar so that you can exercise in teams or as a team. Everyone's avatars will be present on your map and you can even leave notes for others. Want to plan ahead of time? Create your route and set an event reminder. Events can be coordinated with friends through the web based app. We can achieve milestones together, and support each other.

The idea for this application came about when one of our team members described a charity event where people would cycle throughout the city on different paths depending on an entry fee. We sought to give people the ability to not only come together and make their own events, synchronizing their routes with others, but also to provide individuals with a way to push themselves to go the distance.

## Competitive Analysis (GB)

| | beneCycle | Google Maps | Apple Maps | Waze |
|---|---|---|---|---|
| Time Estimations | 4.2 | 4.5 | 3.8 | 4.5 |
| Traffic Markers | 4.5 | 4 | 3.5 | 4.5 |
| Sharing | 5 | 3 | 3 | 2 |
| Integration | 4 | 3 | 3 | 3 |
| Customization | 5 | 3 | 3 | 4 |
| Saving Options | 5 | 3 | 3 | 2 |
| Mean: | 4.6 | 3.4 | 3.2 | 3.3 |

beneCycle provides a progress bar that calculates the percentage of the trip that has been completed, the percentage of the trip that is left, the time elapsed, and the remaining time. The other applications may not always provide the most accurate time estimations as the rate at which a person is traveling is not taken into account.

Google Maps - https://www.google.com/maps

Google Maps has a limited amount of knowledge about the current traffic. They might not include traffic lights, road work, and are not suited for cyclists and pedestrians. The main audience are those who are not in cars, and therefore need to know how to navigate their area likewise.

Waze - https://www.waze.com/

Waze lacks a depth of options that enable users to fully personalize their experience. They need to be able to save various routes, add waypoints, track and compare their target destination to previous destinations, and more. Predetermined routes are limited at best, and the user should be able to edit their trip on a whim.

Apple maps - https://www.apple.com/maps/

There isn't any way to collaborate with other people to create events where groups can walk or ride together along the same path. The app needs features that allow groups to compare their progress. Also integrating social media into the application would keep them connected.

Advantages: beneCycle focuses on the social aspect as well as the personal aspect of travel. It is meant for the trailblazer who wishes to test their speed and push the boundaries of where they can go. beneCycle gives the user the flexibility to decide where to go and how to get there, while offering avatars, route sharing, commenting, and much more to keep people connected.

## Data Definition (AM)

| Name | Meaning | Usage | Comment |
|---|---|---|---|
| Rider | actor | Use Case Scenarios | This person logs to become a member and select a path. |
| Merchant | actor | Use Case Scenarios | This person logs in to list location & product/services |
| Charity Worker | actor | Use Case Scenarios | This person logs in to view all users, contributions & communications |
| Member | actor | Use Case Scenarios | A user who is registered with the system |
| Non-Member | actor | Use Case Scenarios | A user not registered with the system |
| Path | data | Navigation | Delineates chosen path that the Rider/s must navigate. Perhaps in the future audio/visual can navigate the Rider through each of the paths. |
| Attraction | data | Activity Type | Charity-approved merchant product/service |
| Account-Rider | data | Use Case Scenarios | Store rider info |
| Account-Merchant | data | Use Case Scenarios | Store merchant info |

| | | | |
|---|---|---|---|
| Calculator | data | Use Case Scenarios | Keeps a running total of contributions & total number of volunteers and riders. |

| Name | Meaning | Usage | Comment |
|---|---|---|---|
| Payment Page | User Interface | User Interface | Allows a user to buy a ticket or donate via link to an external site like Square. |
| System | Platform & Hardware Services | Use Case Scenarios | The mySQL database, code, front-end design and back-end supporting services |
| Benecycle | Domain Name | Use Case Scenarios | A unique and functional name that can represent the website and all pages. |
| Web address TBA | Production Server | Use Case Scenarios | Data Server |
| Donation Page | User Interface | User Interface | Allows a user to navigate directly to the payment page without becoming a member. |
| Filter | Service | Site User Service | Filters attractions by date |

| | | | |
|---|---|---|---|
| Municipal Worker | data | Use Case Scenarios | Is able to be contacted via the system should assistance be required |

## Overview, Scenario and Use Cases (AM)

Non-Member Expectation

1.) Create account

1.1 The system shall allow the user to create an account by entering first name, last name, phone number, and email address. Page will specify that the transaction must be complete to allow registration. A button marked “Complete Transaction” will be provided. Clicking here will redirect to the Payment page, where an ID will be generated by an external site and returned to the system for storage with the other data. A welcome email will also be sent to the new user upon completed registration. The system will not allow a user to progress to the Payment page if preliminary data are not filled in. The system will not create a new user if a user id is not returned upon successful completion of payment by external site.

1.2 Stimulus/Response Sequence

i.) User enters first and last name

ii.) User enters phone number

iii.) User enters email address

iv.) User clicks on the “Complete Transaction” button.

v.) System redirects the user to the payment page.

vi.) System stores user ID returned upon transaction completion with the other data fields,

and the User is added to Members. Path selection is conveyed by user ID.

vii.) System will confirm this to the user, and provide a back button to navigate to the main page.

1.3 Function Requirement Label

i. ) REQ 1.1 Creating Account

2.) View Information Page, Paths and Attractions Page and Contribution Page

2.1 The registered user may browse to these pages from the main page, and from these pages back to the main page or to the membership or donation page. Users may donate or view information, contributions, attractions and paths without becoming a member.

2.2 Stimulus/Response Sequence

i.) User scrolls to the desired pages.

ii.) User clicks on  a button to donate, become a member, view paths, or information about the charity.

3.) Search Page

3.1 The user may search for participating merchants by clicking on a date. The system will return a filtered list.

3.2 Stimulus/Response

i.) User scrolls to the search page.

ii.) User clicks on a date.

iii.) System returns a filtered list of participating merchants.

3.3 Function Requirement Label

i.) REQ 3.1 Browse by search

4.) Donate

4.1 The user may browse to this page from the main page, and be directed to the Payment page without entering additional info. The user may also enter additional info if they wish. A 'Donate Now' button is available for this purpose.

4.2 Stimulus/Response

i.) User navigates to the Donation page.

ii.) User presses 'Donate Now' button

iii.) User may enter personal information or bypass.

iv.) User is redirected to the Payment page.

v.) System will confirm payment to the user, and provide a back button to navigate to the main page.

Member Expectations

5.) Post Feedback and Media

5.1 The user may post comments, videos or photos visible to other members. The system will not allow non-members to navigate to the Media/Feedback page, or to post feedback/media of any sort.

5.2 Stimulus/Response

i.) The user may navigate to the Media/Feedback Page and enter a comment using a comment box or select ‘upload other media.’

ii.) The system will store comments, photos or videos.

iii.) The system will display stored media at the bottom of the screen, sorted by newest to oldest.

iv.) The page will provide a back button to navigate back to the main page.

5.3 Function Requirement Label

i.) REQ 5.1 Member Post

6.) Help

6.1 A member may post to Help screen to indicate a need for assistance. The message will transmit with their location on the path.

6.2 Stimulus/Response

i.) The member navigates to help screen and presses a button labeled “Help” to ask for assistance.

ii.) The system transmits the signal along with the member location to municipal and charity workers (perhaps by SMS, that is TBD.)

ii.) Once the signal is sent, the system confirms with the member.

iii.) The page will provide a back button to navigate back to the main page.

6.3 Function Requirement Label

i.) REQ 6.1 Help

7.) Map

7.1 A member may navigate to the Map page from the main page, and view their location on their path as well as starred nearby attractions.

7.2 Stimulus/Response

i.) The member navigates to the Map page.

ii.) The system presents a real-time view of their location on a map, with nearby attractions marked.

iii.) The page will provide a back button to navigate back to the main page.

7.3 Function Requirement Label

iv.) REQ 7.1 Map

## Initial List of High-Level Functional Requirements (NA)

1) Will create, customize, and store user profile and route history
2) Social Aspect:
    a) Take user input to display social-media type posts with text and photo capabilities
    b) Search for posts, users and events
    c) Information page with feed of contacts/friends posts and events in user's area
    d) Achievement sharing and group competitions based on stats with contacts
    e) Group Runs/Events with reminders
        i) Possible integration with calendar app
        ii) Route sharing and organizing

    f) Rankings daily/weekly
    g) Popular routes in user's area
    h) Customizable avatar for each user which shows up on the user's location on the map
3) Send user to an external service to process donations
    a) Donations and amounts associated with certain routes
4) Map:
    a) Uses google maps
    b) Waypoints marked on map
    c) Social posts visible and accessible from associated geotag
    d) The system will operate in a specific geographical location of the current event, and will not operate outside of it
    e) Routes
        i) Route shading
        ii) Route choosing and payment
        iii) Route suggestion based on user input or profile preferences
5) Help
    a) Organizers contact information listed and redirect to external phone to call
    b) Municipality contact information listed and redirect to external phone to call
    c) 911 with redirect to external phone to call
    d) FAQ page
    e) Feedback page
        i) Form which allows users to send message to app developers or IT
        ii) Email listed with redirect to external email app to send message to app devs or IT
6) Other functionalities:
    a) Timer/Stopwatch
    b) Distance logger
    c) Calorie Estimator
    d) Route history with statistics (mph) and record times/distances

e) Waypoint notifications

## List of Non-Functional Requirements (NA)

1) Product Requirements:
    a) Speed - Refresh and processing times should be less than 3 seconds.
    b) Size - Less than 10 MB
    c) Ease of Use - intuitive controls
    d) Reliability - Mean time to failure 8 hours, about twice the amount of time a person may be at an event
    e) Robustness: Failure rate should be less than 1%
    f) Portability: Functional for IOS and android OS
2) External Requirements/Organizational Requirements
    a) Android compatible
    b) IOS compatible, follows ios design themes and guidelines
    c) Compatible with Square for payment processing
    d) Access to system contacts and location, potentially calendar app

## High-Level System Architecture (EM)

1. **Lamp Server**: we will be using a Lamp Server to host our project after its completion.

2. **WhatsApp Messenger**: WhatsApp is what our group will be using to communicate with each other

conveniently and share progress. We can use this to notify and update each other on plans and keep up with schedules.

3. **mySQL Database**: we will be implementing a mySQL database to keep track of the information required for this project. Organizations will be able to post events and users will be able to keep track of login information, donation and participation history, and social media interaction. All relevant information will be stored in the database for retrieval and use by the respective parties.

4. **GitHub**: this will be used to upload and share code with the other members of the group and keep track of progress made in a transparent format that each of us can view and edit without interfering with one another.

5. **Brackets**: for the development of the web pages associated with this web app, Brackets will be used as a convenient text editor allowing easy navigation of folders, useful autocomplete features, and, most importantly, a live preview of each stage of the web pages as they are being developed.

6. **Apache Cordova**: this is a mobile application framework that will allow us to keep our app compatible with the Android, iOS, and desktop platforms. It enables us to build hybrid web applications using CSS, HTML, and JavaScript

7. **Jira**: for the planning and organization of this project's development schedule, we will be using an agile development methodology, which will be kept track of using Jira. This platform features planning boards and objective lists and is a useful tool for coordinating the future steps of our team planning.

8. **Google Maps API**: we will be using this API to embed Maps capabilities into our application. Since location routing and geotagging are cornerstones of our requirements, this API will be necessary to implement to meet our goals. ( https://developers.google.com/maps )

9. **Square API:** Payments will be handled by the ecommerce platform Square, who supplies a package to enable a page that requires hosting upon our site, and provides access to their external databases. ( https://squareup.com/us/en/payments/online-payments **)**

## High-level system architecture and database organization

High level Architecture of the code has been made consistent with UML class diagram. (AM)

### DB organization

#### Main database (high-level) schema/organization

Benecycle is a web application that synchronizes routes for cycling, and similar charity events (of different charities) with the participant's donation levels. A level, determined by the amount donated, is assigned to a route. More challenging routes require higher donation levels. These routes are a certain distance in length. These users may create an account, however doing so is not mandatory. Users with an account do have the benefit of social networking. This includes having a list of prior events, route for the current event, groups, rankings (most popular route, best times, etc). Each event has designated charity workers who will set up and manage the event. Merchants are also able to sponsor events and can be found using the sights search feature. These merchants may offer a product or service for the event. Each event has a total number of contributions from participants. Municipal workers from the city in which a given event occurs coordinate with the charity workers. For each event there is also a total number of volunteers (which are made up of municipal and charity workers) and a total number of participants. Event's locations need to be kept.

Here is a breakdown of the Relationships and Entities:

Relationships:

-Participates_In (Riders participate in Events)

-Sponsors (Merchants sponsor Events)

-Volunteers (Charity_Workers manage/oversee Events)

-Contains (Events contain Routes)

-Coordinates (together, Charity_Workers and Municipal_Workers Coordinate Routes)

-Navigates (Riders navigate Routes)

-Donates_To (Riders pay a donation in order to ride in the Event)

-Hosts (Charities host Events)

Entities:

-Routes (attributes-Route_Id, Distance, Difficulty, Best_Time, Rating, Donation)

-Riders (attributes-R_Name, R_Email, R_Phone_Num, R_User_Id, Prior_Events, Groups, Ranking, Best_Times)

-Events (attributes-E_Name, Time, Date, Location, Contributions, Participant_num)

-Merchant (attributes-Me_Name, Offer, Me_User_Id)

-Charity_Worker (attributes-C_Name, C_Email, C_Phone_Num, Job, C_User_Id)

-Municipal_Worker (attributes-Mu_Name, Mu_Phone_Num, Mu_Email,

Mu_User_Id)

-Charity (attributes-Char_Name, Char_Phone_Num, Char_Id)

Here is the schema for the database:

-Routes(Route_Id: string, Distance: real, Difficulty: integer, Best_Time: time, Rating: real, Donation: real, Primary Key(Route_Id))

-Riders(R_Name: string, R_Email: string, R_Phone_Num: string, R_User_Id: string, Prior_Events: string, Groups: string, Ranking: integer, Best_Times: string, Primary Key(R_User_Id))

-Events(E_Name: string, Time: time, Location: string, Contributions: real, Participant_Num: integer, Primary Key(E_Name, Location))

-Merchant(Me_Name: string, Offer: string, Me_User_id: string, Primary Key(Me_User_Id))

-Charity_Worker(C_Name: string, C_Email: string, C_Phone_Num: string, Job: string, C_User_Id: string, Primary Key(C_User_Id))

-Municipal_Worker(Mu_Name: string, Mu_Phone_Num: string, Mu_Email: string, Mu_User_Id: string, Primary Key(Mu_User_Id))

-Charity(Char_Name: string, Char_Phone_Num: string, Char_Id: string, Primary Key(Char_Id))

-Participates_In(R_User_Id: string, E_Name: string, UNIQUE Location: string, Foreign Key (R_User_Id) references Riders, Foreign Key(E_name) References Events)

-Sponsors(E_Name: string, UNIQUE Location: string, Me_User_Id: string, Foreign Key(E_Name) References Events, Foreign Key(Me_User_Id) References Merchants)

-Volunteers(E_Name: string, UNIQUE Location: string, C_User_Id: string, Foreign Key(E_Name) References Events, Foreign Key(C_User_Id) References Charity_Workers)

-Contains(E_Name: string, UNIQUE Location: string, Route_Id: string, Foreign Key(E_name) References Events, Foreign Key(Route_Id) References Routes)

-Coordinates(Route_Id: string, C_User_Id: string, Mu_User_Id: string, Foreign Key(Route_Id) References Routes, Foreign Key(C_User_Id) References Charity_Workers, Foreign Key(Mu_User_Id) References Municipal_Workers)

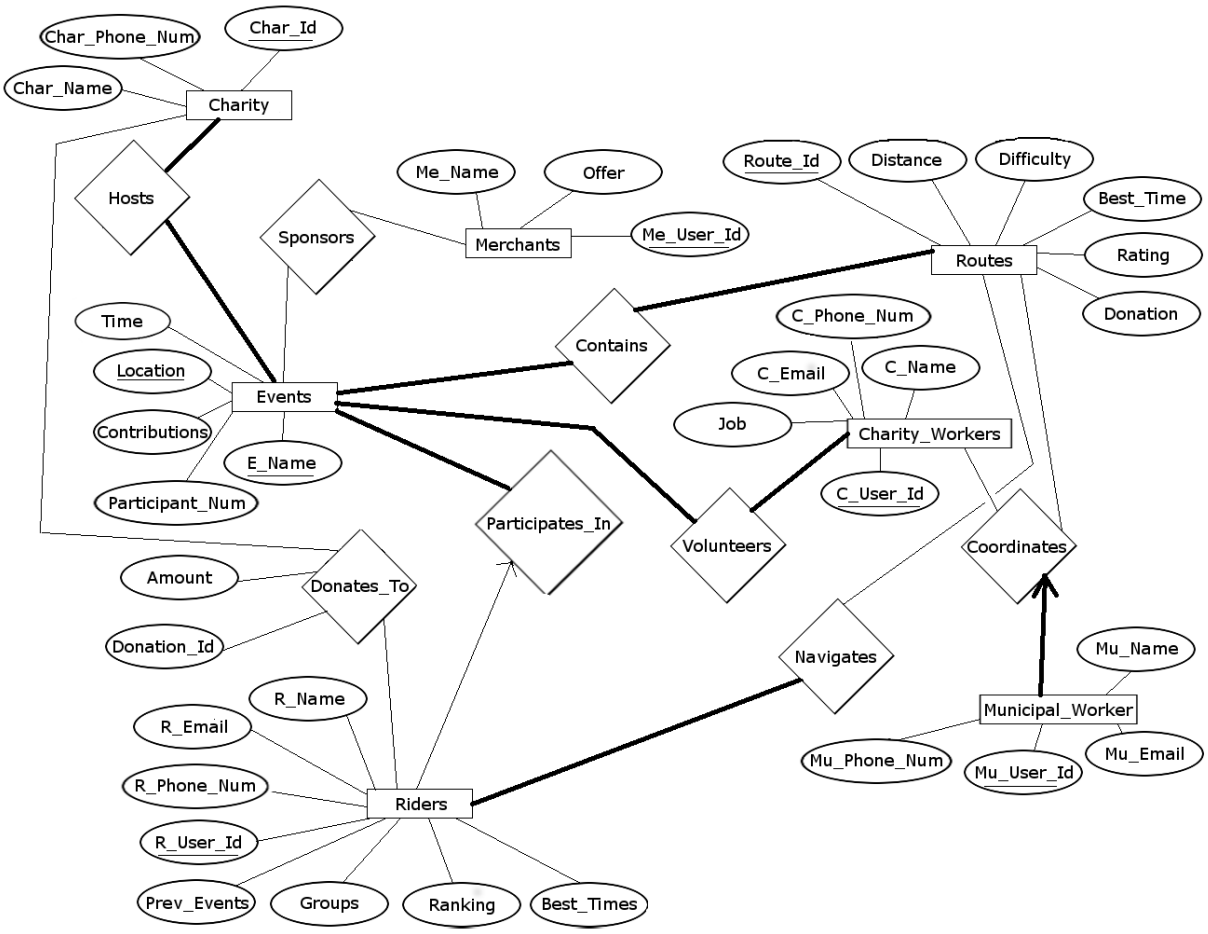
-Navigates(R_User_Id: string, Route_Id: string, Foreign Key(R_User_Id) References Riders, Foreign Key(Route_Id) References Routes)

-Donates_To(R_User_Id: string, Charity_Name: string, Location: string, Amount: real, Donation_Id: string, Foreign Key(R_User_Id) References Riders, Foreign Key(Charity_Name) References Charity)

-Hosts(Char_Id: string, E_Name: string, UNIQUE Location: string, Foreign Key(Char_Id) References Charity, Foreign Key(E_name) References Events)

## Main DB tables/items in each DB table

Here is the Relational Diagram (the schema reflects the main db tables and their contents):




## Media storage (NA)

All media associated with posts, profile, and event information will be stored in a database because they are well defined and structured. All possible inputs and relations are predetermined. GPS data can be stored in the standard latitude/longitude format as a series of integers.

Search/filter architecture and implementation (AC)

This algorithm has yet to be implemented. However, the algorithm will first use ***mysqli_real_escape_string()*** function to make sure that the user search term does not contain any injection attack against the database. Then the search term will be broken down using the mySQL query mechanism to see if any part of the search term is related to the attributes of a registered member (rider). A member's attributes contain (temporary):

- ID: varchar(11)
- First Name: varchar(50)
- Last Name: varchar(50)
- E-mail: varchar(50)
- Phone Number: varchar(50)
- Date Registered: date
- Sex: varchar(11)
- Age: int(3)
- Address: varchar(50) **(To be added)**
- Path: varchar(50) **(To be added)**

The result is then returned using the ***mysqli_query()*** function, and then ***mysqli_num_rows()*** function is used to check the number of results returned. If the number of results is 0 then a text saying "No matching results" will be displayed on the webpage, otherwise the matching result will be displayed on the page. For now, the registered members (riders) will be filtered/sorted based on their attributes. In addition, the search terms to be used will be the one or more attributes of the members. In the future, this function will be expanded to posts/articles made by the members (riders), sponsors, and etc.

APIs (AM) EM

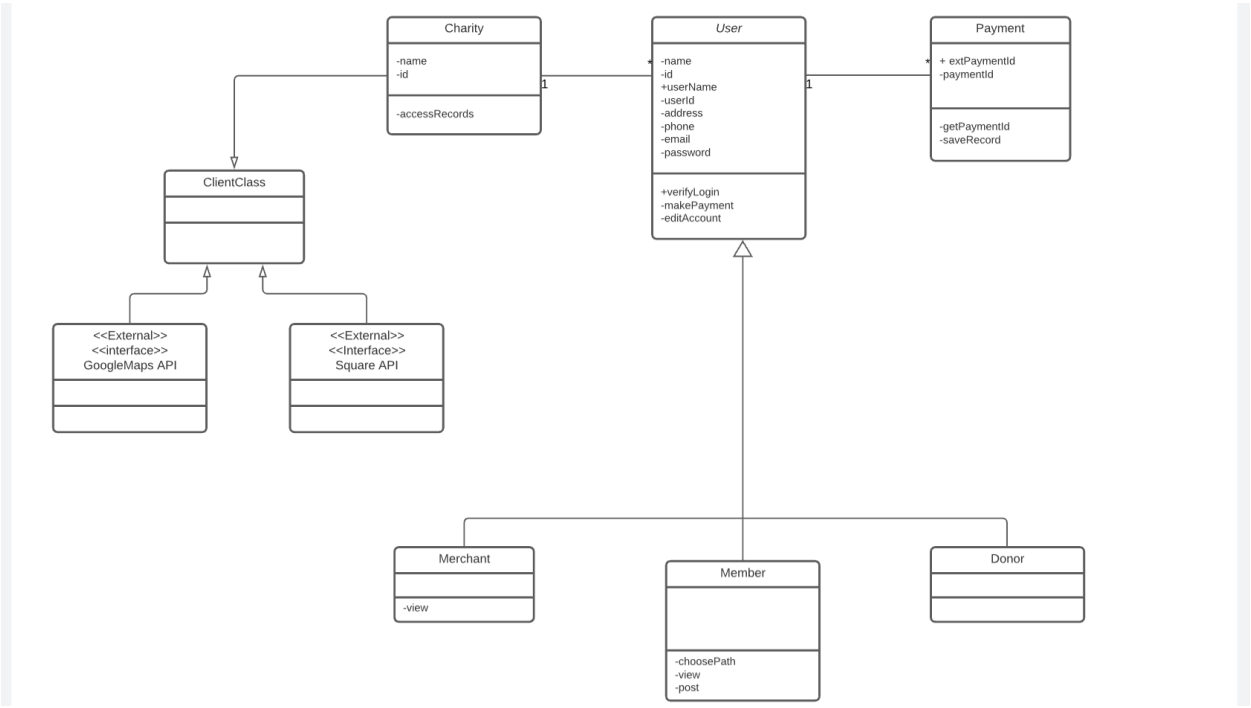
We will not be developing custom API's to assist in this project. Rather, we will be reliant on Square to process payments, and GoogleMaps for path navigation functionality. In the event that we cannot find a free or trial version of a third-party payment platform, we may have to revisit this.

Other algorithms/processes (NA)

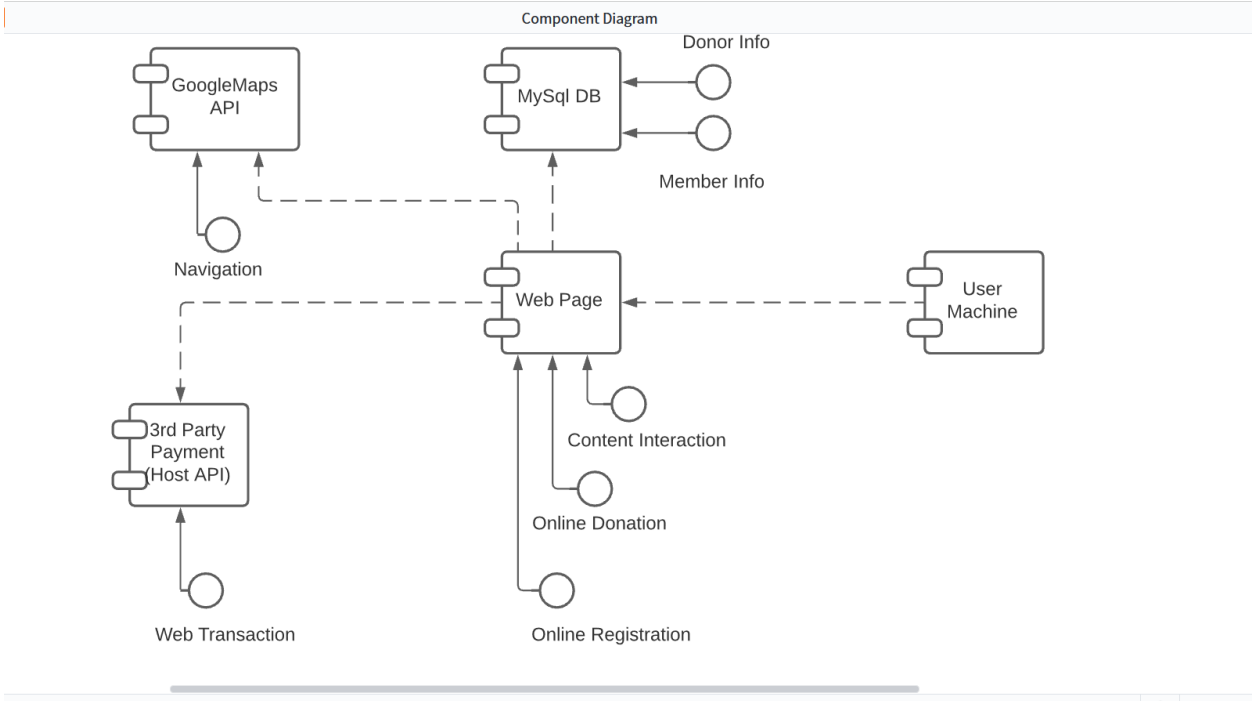
6) Posts will be able to be ranked by recency and popularity, filtered by day, week, or month. Future events will be ranked by proximity of time and/or distance. Recommendation of a specific path can be done using a tally dependent on the features of the path and the preferred features of the user. The path that most fits the preferences of the user will have the most points, and be recommended.

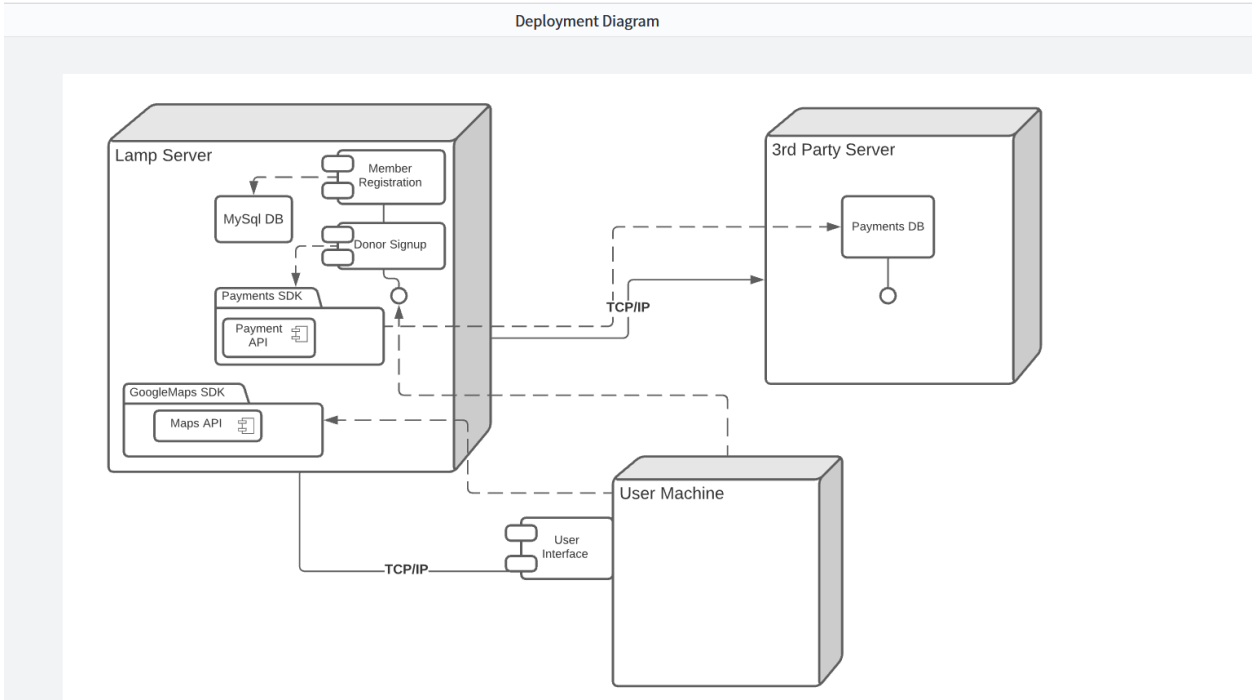
## High-Level UML diagrams (AM)

1. ) UML class diagram:



## 2.) UML Component and Deployment diagrams



## Current Risk Assessment (AM) EM

### Skills

Current risks involving skills center around the fact that none of us possesses a native knowledge of PHP. We have been able to create the vertical prototype, but will have to focus more resources to add greater functionality.

### Schedule

Though we are on task, there are significant risks to our schedule due to issues with intercommunication amongst the group. There is an unacceptable time delay between the instance of a challenge and the coordination to address it. As a group, we will have to reevaluate the essentials of our project, and see if we can strip it down to fit the time frame.

### Technical

There are potential risks associated with the integration of the API's we are incorporating in our project. We lack experience with such integration, so there is uncertainty about the process of making it work. Additionally, one of the API's is proprietary; we will have to find an implementation of this API that will work for our project without paying for the service.

Teamwork

Currently our team is functioning suboptimally. Collaboration is poor, and the majority of team members are neither motivated nor proactive. This makes assessments of the potential of future individual contributions difficult.

Legal/content risks

We do not anticipate risks due to proprietary content; Google Maps should be free for our purposes, and we are aware of having to approximate a third-party payment API.

There are inherent risks associated with the nature of our product. The key to resolving these lies not within our product, but in legal documentation. This is why we have chosen to leave certain processes outside the scope of our app. For example, Charities may not create themselves as users, but must be vetted before the app can be distributed in connection with their verified account. This is to be certain they are actually a designated charity, and in compliance with all local laws and regulations regarding a cycling event.

## Vertical Prototype (AC)

Our team has successfully created a vertical software prototype to test the infrastructure and frameworks. It exercises the full deployment stack and is deployed from the team account. It allows one to enter a term in the browser, get a response from the DB, and then render it back on the browser.

Link (No audio): https://youtu.be/8AEZm9IAf1M

## Team

1. **Andrea Malvezzi** - Product Owner
2. **Andy Cheng** - Scrum Master
3. **Georges Bourque**- Back end Lead
4. **Noa Abiri**- Front end Lead
5. **Ellie Moffatt**- Software Engineer