Thomchelik Tunca
Eco-Friendly
Capstone Senior Design
11/11/2015

Specification Document

Eco-Friendly is an interactive Android application that follows and tracks users' use of transportation during the day to figure out how much carbon dioxide that specific user emits. The application does this by first deducting when the user starts a trip, then computing what kind of transportation the user is using, whether it be by car, bus, metro, biking, or walking, and lastly calculating the carbon dioxide emissions throughout that trip. The main objective of this application is to influence users to emit less carbon dioxide during their day-to-day trips, hopefully in the long-term changing their use of transportation to be more sustainable. To do this, the application will create a personalized profile and show the user if they are emitting more or less carbon dioxide during the day in comparison with the sum of each days' carbon dioxide emission from when they first used the application. Graphically, the user is shown each day's carbon emissions and trips with colors indicating which days they emitted less and which days emitted more carbon dioxide. Moreover, the application will notify the user how they can better their emissions during a day, such as tell them to walk or carpool to work instead of taking a bus or taxi.

There is one specific group of user that this application is targeting, and that is anyone who seeks to be more conscious about the health of the planet or wants to do their part in cleaning up the atmosphere. Each instance of the application (one per phone) has one user so there is no need for an admin or any other type of users. Each phone will be that user's profile and support only that specific user. As the topic of climate change and carbon emissions are more prevalent in the daily news, in the United States and worldwide, it is clear that the pool of users will not diminish anytime soon. Likewise, the group of users is also growing at an astonishing rate as the health of the Earth is worsening each year, evidenced by increasing atmospheric and oceanic temperatures, an influx of tropical storms, and more. A diminishing state of the planet demonstrates to people exactly how their impact is truly affecting the planet.

The start of every use case will be the same. The application will wait in an inactive state until it assumes that the user is starting a trip to travel to another location. This pre-condition will be initialized when the user moves out of their current 50-foot radius (travels more than 50 feet from where they previously were) or moves at an average speed of more than 20 mph. The trip will end when the person is again stationary at a location, emulating a real life example of someone leaving their current location to go to another location. After the end of a trip, the application starts to compute which kind of transportation they are using whether it be a bike, car, bus, metro, or just walking. This is where the different use cases apply.

The simplest use case to figure out is if the user is walking to their destination. To do this the application will use the user's phone's GPS to figure out if their average speed is 5 mph or less. Similarly, if the use case is that the user is using a bike for transportation, the user's average speed would range from 10 to 15 mph. If the application understands that the user is walking or riding a bike then it assumes that the user is emitted no carbon dioxide.

The use cases for the mode of transportation being a car, the metro, or a bus is more complicated. The application will distinguish if the user is using one of these types of transportation with the precondition that the user is going at an average speed of more than 25 mph with the phone's GPS. Once it understands the user is traveling at that speed, then it then works to determine which of these three types of transportation is in use.

The application will understand if the user is on a bus or metro by using the Google Maps API. The Google Maps API has the DC area metro and bus routes, stops, and times that the application will utilize to figure out if the user is on a bus or train. If the user is on a bus then the GPS will follow the path of a bus route, stopping at only the bus stops for an elongated period of time, and also traveling at a speed that is greater than or equal to 25 mph. Similarly, if a user is on the metro the GPS will follow the path of a metro line, but will have another condition than the bus option: when the user is on a metro they lose service until they get service back at another metro stop.

Using a bus or a metro train will trigger the application to seek to find out how many people are also using this means of transportation. This is important because the user is actually using less carbon dioxide when in a multi-person vehicle. The way the application will do this is by prompting the user: "How many people are on the bus/metro-car with you?" The user will have four options to click, either 0-10, 11-20, 21-30, or 31-40 people.

The last use case of this application is a user is in a car for transportation. Unlike the bus or metro, a car can travel anywhere on the road, not just on Google Maps API-specified bus or metro route. An additional step needs to be taken in order to distinguish between a car and bus mode of transportation. If the user is tracked on a bus route for more than 1 minute, the application will prompt the user: "Is your mode of transportation car or bus?" The user selects the appropriate option and the use case continues.

Once the application distinguishes which use case applies to the user's movement, it computes how much carbon dioxide is emitted depending on how far the user travels in that vehicle. For example, if a bus emits 25 pounds of carbon dioxide every mile and a person is on it for 5 miles, then the total carbon dioxide emission is 125lbs of carbon dioxide. Washington D.C. Transportation uses the same types of buses and metro trains. This means that each vehicle type emits the same amount of carbon dioxide; therefore it is simple to calculate the carbon emissions emitted by these vehicles. If the car is the mode of transportation, the user will be prompted what type of car they are in, as well as the make of the car and year of the car. Information will be stored in the application's database with a variety of vehicles' emissions data, used to determine how ever many pounds of carbon dioxide that certain car emits. After the application gathers all the information about the trips the user has taken in a day and how many pounds of carbon dioxide they emitted, it will neatly list them in a separate page of the application. This page is key to display and explain to the user the total picture of their day.

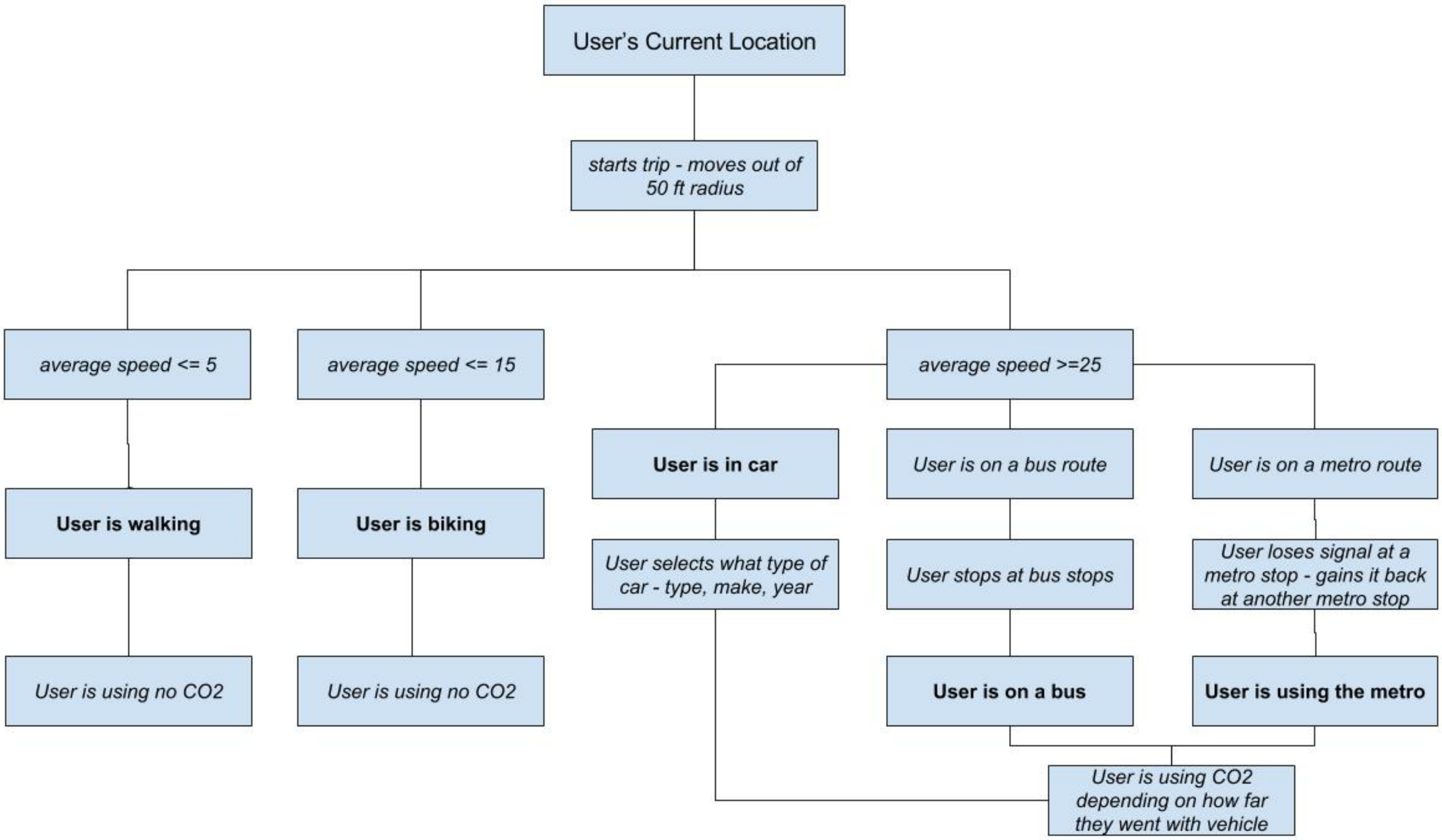

Figure 1: Workflow

As summarized in Figure 1, the use cases for the application depend on the mode of transportation the user uses to calculate the carbon dioxide emissions during each trip taken in a day. These cases are if the user is walking, biking, taking a car, bus, or metro train to get to their destination. The functional components of each transportation source is similar, the application receives input from the phone's GPS for its location and speed and outputs the type of transportation the user is using in order to calculate the total amount of carbon dioxide the user uses emits during a trip in the day. When the user is walking, the functional component is that the application needs to understand the user is walking from their speed and thus does not to compute any carbon dioxide emissions during that whole trip. Similarly, with biking, when the user is traveling at an average speed from 10 to 15 mph the application does not compute any carbon dioxide emissions during that whole trip. If the mode of transportation is a vehicle the functional component is different, but still result so that the end result will be calculating the carbon emissions of the user.

If the user is travelling faster than an average speed of 25 mph then they are using a vehicle and thus the application knows to start to track how much carbon dioxide is emitted. This makes the functional component of the vehicles understanding what vehicle is used and how much carbon dioxide that vehicle emits every mile. The user will input what type of car it is (whether it be an SUV, sedan, or compact car), the make of the car (what company makes the car – Toyota, VW), and an estimate of the year of the car.

The functional components of the bus and metro differ compared to the car; the application has to analyze input from the Google Maps API and the GPS determine if the user is on a bus or train route. Once the application understands if the user is on a bus or train, it will compute the total mass of carbon dioxide emitted per mile, depending on the accepted exhaust emission value of type of bus or train car Washington D.C. employs. Another functional component of using the public transportation is figuring out how many others are using that specific mode of transportation with the user, so that the application can divide the carbon dioxide emitted by that mode by the number of people in the shared vehicle.

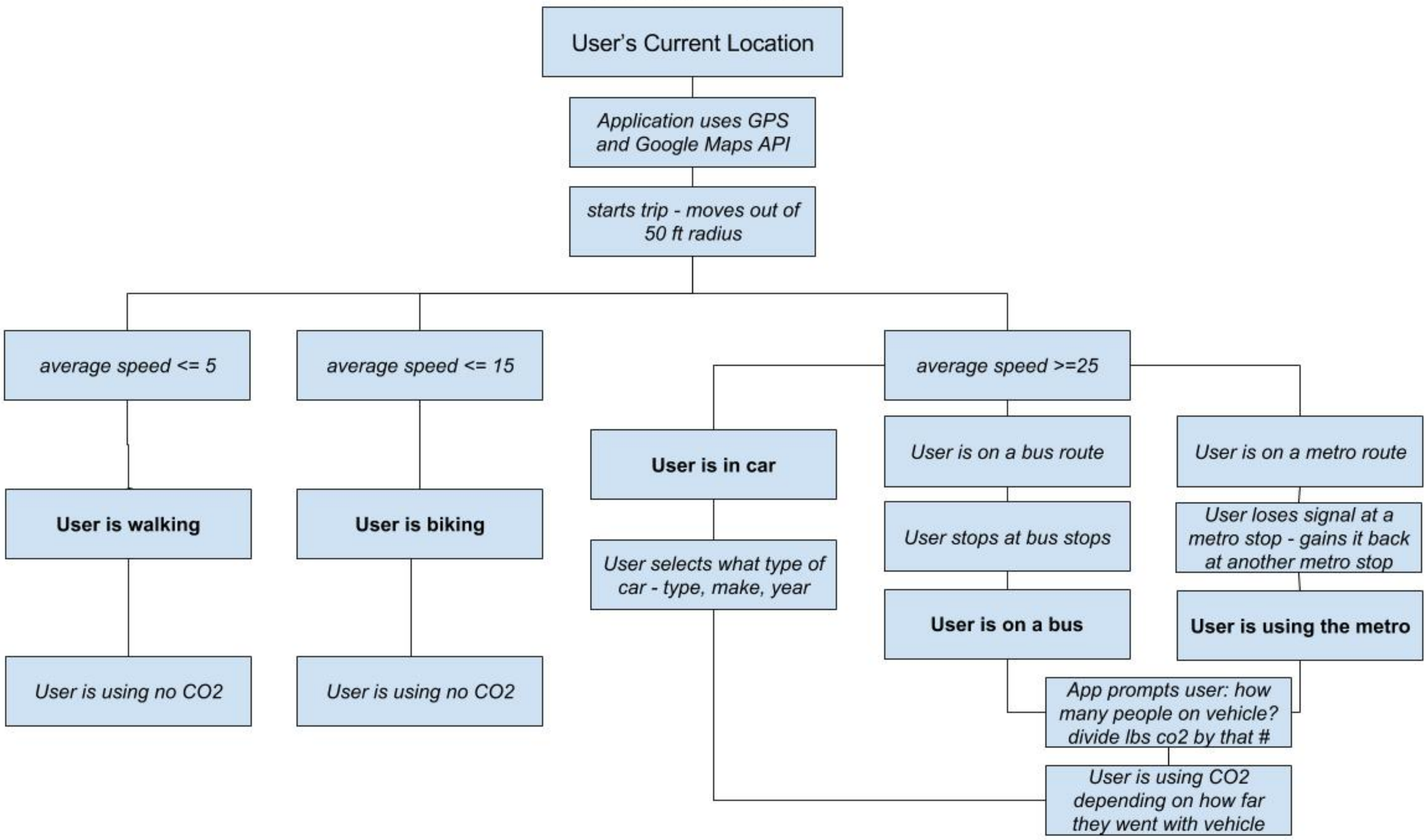

Figure 2: Functional Flow Chart

The application also makes use of its non-functional components. Usability is a huge asset to this application. The application will be user-friendly and have an intuitive layout so people are not frustrated and subsequently choose to stop using it. The majority of its functionality is automatic (using the phone's GPS and the Google Maps API) so the user will not need to worry about constantly notifying the application if they are about to take a trip, nor will the user need to manually enter in how many pounds of carbon dioxide they emitted in a trip. Another non-functional component is its layout. The application's main purpose is to show the user how many pounds of carbon dioxide they emit on a daily basis; it is necessary to have a user interface that is easy-to-follow, and simply concludes if they are doing better or worse each day. Performance and reliability is also moreover incorporated in the application. Eco-friendly must be reliable in its algorithm to automatically detect a trip beginning and end, as if a user is on a trip and the application does not understand, then the whole premise of the application fails. The application also depends on the automatic trip detecting as this process is integral in taking accurate readings of how many pounds of carbon dioxide the user emits.