Course Title: 14:332:452 Software Engineering

Group 4

Onward (Traffic Monitoring)

Report 1

February 19, 2017

GitHub (e-archive): https://github.com/solejar/traffic_ru_ece
Website: http://www.onwardtraffic.com/

Team Members

Name	Email
Ridwan Khan	ridwankhan101@gmail.com
Brian Monticello	b.monticello23@gmail.com
Mhammed Alhayek	almoalhayek@gmail.com
Sean Olejar	solejar236@gmail.com
Lauren Williams	laurenwilliams517@gmail.com
Shubhra Paradkar	shubhra.paradkar@gmail.com

Individual Contributions Breakdown

All team members contributed equally.

Table of Contents

Customer Statement of Requirements

Modern navigation applications allow the user to enter a starting location A and a destination B and then calculate the best route between both locations, all the while giving an estimation of how long the trip may take. These applications do not, however, allow for users to see the future conditions of traffic and get an idea of how long their trip may take beforehand. As a result, users' ability to effectively plan ahead and adjust for traffic related issues is limited. Being able to more accurately measure trip time is an incredibly valuable asset for both business and commercial users. As a result, an application that allows for a user to view traffic conditions in an area or along a route days before their travel would be meaningful and effective for a large group of users. Although there are several applications that individually display current traffic and weather conditions, they do not take into account historic traffic and weather, and will not calculate the cost of a particular trip. Most importantly, these measures cannot be taken days in advance, which is a necessity for most users. An application that takes into account all of these factors simultaneously will create the most appropriate solution for every-day users.

Currently, most mapping applications are mobile oriented and often restrict the user from entering their lengthy travel plans in favor of ease-of-use for immediate, simple travel plans. A web-based application that focuses on helping accommodate users' longer plans would be more helpful in specifying the entirety of the user's trip. This way each location, date, time constraint, and weather condition could be thoroughly described, and a route could be planned that would take each of these factors into account. This application would be more fitting for a traveler that is in constant need of adjusting to different traffic patterns and is inconvenienced by limiting mobile-only software that only allows for basic itinerary information. With this in mind, however, as traveling users often do also spend time away from a computer, it is important that the web-application be responsive for mobile users.

In order to select the fastest **route** between two locations ahead of time, it is necessary to predict the traffic along all possible routes between those locations. But how can such a prediction be made? Historic traffic and weather records can be an effective indicator of future traffic in any given area. By using this historic information, this application would allow the user to see predicted traffic along the three fastest routes connecting two locations once they enter the time, day, and expected weather. Allowing the user to see predicted traffic along these routes will allow them to select the route that will likely minimize time wasted in traffic. A system that uses data from over a large period of time to identify patterns of traffic would likely be accurate enough to consistently help users do this. This would be useful if the user were moving to a new area and wanted to see what their commute would look like in the morning. They could enter their home address and work location, and would be able to see what their average daily commute may look like based on past traffic data. They could then adjust their route accordingly, depending on which routes have the least traffic during rush hour. In addition to having a starting location and destination and intended time of travel, the user may have a weather they expect to travel in. Furthermore, if they have a specific day within 10 days in mind, then the application could use a weather forecast instead of user input for the weather parameter. In either case, using the historic traffic and weather records, an accurate representation of traffic conditions along a route can be shown. Roads along the route can be highlighted based on traffic severity. And if a route contains excessive traffic, a detour route can be suggested along with traffic conditions on that route as well. A user would also be in need of this type of application if they wanted to plan a road trip. The user could figure out the optimal departure times on each day of their trip that would minimize time spent in traffic. This application would give meaningful information to users looking to plan a trip ahead of time, or investigate general traffic conditions along a certain route.

Along with predicting the future traffic along a specific route, it would also be useful for the user to see the general traffic trends in a larger area. The user should have the ability to enter a zip code (or city), a radius around that zip code, a date and/or time, and weather conditions in order to see the predicted traffic for the designated area. A traffic **heat map** could effectively accomplish this, allowing the user to see how much traffic is expected in a particular area at a particular time. By color-coding the traffic severity, users can easily get a visual overview of how congested certain roads are at a certain time. Particularly congested area would be identified as traffic **hot spots**. These would allow the user to identify and avoid routes that are expected to have heavy traffic, if possible. If the user doesn't want to manually enter the weather conditions, the application should also be able to use the weather forecast if the date of travel is less than 10 days away. The traffic will only be shown on major roads, in order to avoid visual clutter. A major road is defined as any interstate, highway, or county road. If the user-specified area does not contain any such roads, the heat map will simply select the roads with the highest percentile of average traffic. The user would also benefit from the ability to show or hide the traffic on this map by the severity of the predicted traffic. That way, the user can more effectively exclude lighter traffic from the visualisation, and focus on the areas with heavier traffic. Combining all of these tools, the user would then be able to use this **heat map** to make educated decisions about where and when they would like to plan their travels.

Another useful extension upon the basic route feature would be the ability to accommodate multiple stops in a user's travel plans. In other words, many users will not only need the ability to plan their route out from point A to point B, but also to plan out their **agenda** if they have multiple stops throughout their day or week. A user may have to run to the grocery store, go to the gym, and go to the mall all in one day. The user would like the ability to enter in all of these locations and find the route with the least amount of traffic between them. The user would benefit from being able to include time constraints to describe this trip. Examples of these constraints are the time frame in which they would like to accomplish all of their tasks, and the time at which they want to arrive at one of their locations. Considering these constraints, the user would like an application that can generate an agenda that minimizes total time in traffic. They would like their destinations displayed in order along with the optimal routes and suggest time of departure. This agenda feature would also allow the user to detail weather conditions for a particular journey so that they can receive as accurate a trip agenda as possible. This feature would be effective for users who generally travel with multiple destinations in mind and do not have the ability to plan ahead without knowing how much traffic they will face. These users are also generally unaware of the best time to depart for each destination that would allow them to avoid traffic. Currently applications exist where the user is only asked to specify one destination and there is no intuitive way to add multiple destinations. An application that can create an agenda with multiple waypoints would be a huge convenience to a user over having to plan each route one at a time. Looking at the overall trip, rather than the individual routes, allows the application to more effectively minimize total time in traffic.

Currently available applications provide a good estimation of trip mileage, but they do not go any further in turning this information into something usable for the user. The user wants to know not only the mileage of their trip, but also details regarding gas consumption for their trip and therefore the costs associated with this gas consumption. The user can also use the cost of gas as a heuristic for Onward's best route/agenda recommendation. This feature is especially useful in the context of the recent increase in taxi and delivery services such as Uber and Lyft. This feature would allow companies to both accurately predict overall gas expenses and to also suggest routes for their driver that would minimize gas expenses. By calculating the cost along with the other individual features mentioned, the user would have a very direct plan of how to travel, when to travel, and the cost of travel. This application would provide that, allowing the users to see a route for their trip, the predicted cost of gas based on the route, and the predicted traffic based on the historic traffic records. Furthermore, if detour routes are suggested due to excessive traffic on primary route, the user can compare the cost of travel between suggested routes.

The above paragraphs demonstrate the need for an application that can fill in the gaps where traditional navigation applications are lacking. While identification of routes and traffic at the current point in time is meaningful, users also have a need to understand and predict traffic ahead of time. Being able to understand future traffic trends on both specific routes and general areas would allow users to be more effective travellers. In addition, an application that gives users the ability to create, plan, and analyze their multi-stop trips and the traffic they will encounter ahead of time would be exceptionally beneficial to have. Many modern users strive to use technology to make conscientious, well-informed decisions about their day-to-day activities. Onward facilitates this part of the user's life, allowing them to make an accurate and efficient decision about their travel plans ahead of time.

Glossary of Terms

Incident - Any event which contributes to an increase in traffic severity. Examples include, but are not limited to, congestion, inclement weather, and accidents.

Location - A user inputted point on the map. It is allowed to be defined on any level of granularity from as specific as an address, to as general as a city or zipcode.

Heat Map - A heat map refers to a city or zipcode scale map which contains color indicators along the major roads of the designated area. The color indicators vary along a spectrum, and serve to indicate the average traffic conditions on the roads.

Major Roads - Where applicable, defined as all interstates, highways, and county-roads. Where not applicable (i.e. few relevant roads of the above categories), Onward may also consider roads with the highest amount of traffic in the area as major roads.

Hot Spot - Either a subsection of a road or an intersection which is known to regularly exhibit very congested traffic.

Route - A path that connects any two locations together.

Primary Route - A route that is recommended to the user based on the user-determined heuristics (i.e. total time in traffic, gas cost).

Detour - An alternative to the primary route.

Agenda - An element which consists of several, individual, contiguous routes which are combined together. While an agenda will typically end at the same location it starts at, this is not required by definition.

Heuristic - A tool that allows Onward to make decisions regarding how to predict future traffic.

Highlight - A continuous-color marking, along a road segment. Indicates traffic severity.

Landing Page - The homepage of the Onward website that the User first encounters.

System Requirements

Functional Requirements:

Table 1: Enumerated Functional Requirements

Identifier	Priority	Requirement
REQ1	5	The system shall take user input for a zip code or city in the tri-state area, and a value for the radius of desired traffic information.
REQ2	3	The system shall provide an option to take user input for day of week to travel and user expected weather.
REQ3	3	The system shall provide the option to automatically select weather conditions based on a 10 day forecast.
REQ4	2	The system shall allow a user to choose to show/hide information corresponding to one of 4 different levels of traffic severity.
REQ5	5	The system shall generate a map that shows traffic trends for the area specified by the user. It should highlight all major roads, and use different colors for different levels of traffic severity.
REQ6	2	The system shall generate statistical data about traffic patterns in a large scale area of interest.
REQ7	5	The system shall take input for a starting point and destination address as well as desired time of travel.
REQ8	2	The system allow a user to choose to show/hide information corresponding to the 4 different levels of traffic severity along their route.
REQ9	5	The system shall suggest a route which connects starting point to destination and shall display this route on a map.
REQ10	3	The system should offer up to 2 alternative routes for the user to choose from, in addition to the primary route.
REQ11	2	The system shall display statistical data about traffic patterns along a route in the form of a graph.
REQ12	2	The system should provide an estimated time to get from starting point to destination(s).
REQ13	3	The system shall take as input multiple locations, and any time constraints for the travel.
REQ14	3	The system should allow users to input specified times they must get to certain destinations and constraints on the order to reach

		each destination.
REQ15	3	The system shall display the optimal route to reach the specified destination(s) while stopping at all the requested stops based on inputted constraints.
REQ16	2	The system shall suggest the optimal time to depart in order to encounter the least traffic based on inputted constraints.
REQ17	4	The system shall offer users the option to input vehicle gas economy information in order to receive information about the estimated cost of all suggested routes.
REQ18	2	The system shall provide current traffic conditions from starting point to destination.

Nonfunctional Requirements

Table 2: Enumerated nonfunctional requirements

Identifier	Priority	Requirement
REQ19	3	The system shall provide a mobile-responsive interface.
REQ20	5	The system shall collect traffic and weather data every hour and add any new incident information to the database.
REQ21	4	The system should be able to suggest routes or produce heat maps within 10 seconds of user inputted parameters.

On-Screen Appearance Requirements

Table 3: Enumerated functional requirements

Identifier	Priority	Requirement
REQ22	5	The system shall provide a graphical representation of the user's route in map form.
REQ23	5	The system shall provide a form for the user to input his/her information about the route
REQ244	4	The system shall provide 2 graphs of statistical information about the user's trip based on the specified route details alongside the map interface.

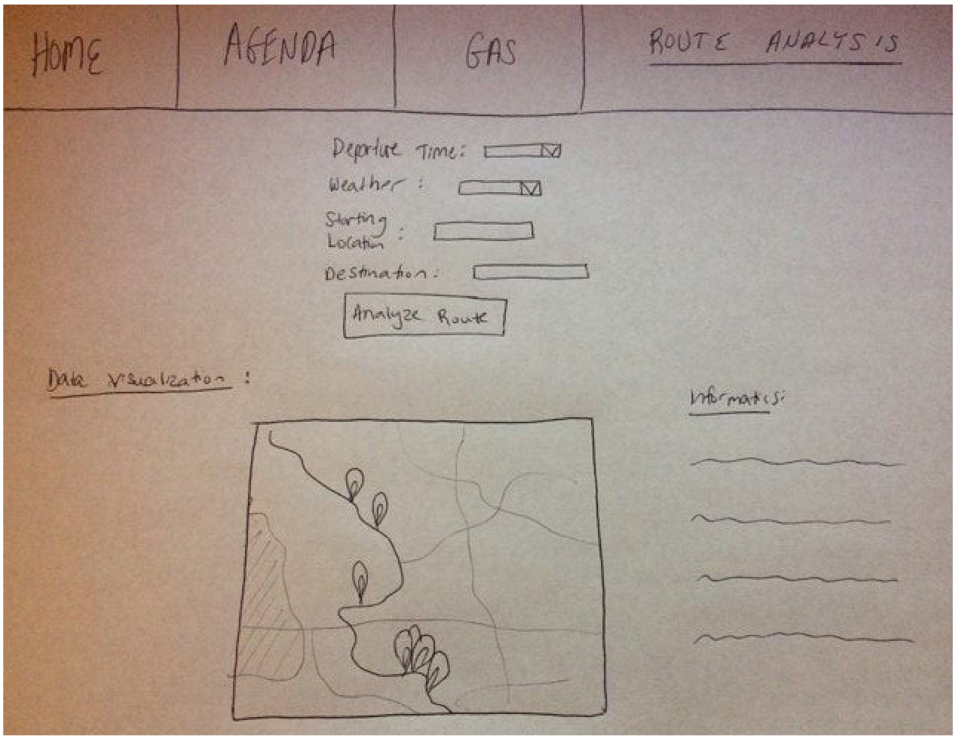

Figure 1: Web Application Main Page

The above image represents the user interface for basic route analysis. On this page, users will be able to enter in the relevant information about any trip. Upon doing so, and hitting "Analyze Route", Onward will visualize historical traffic trends that best fit the user's trip. The departure time drop down box will include the date, as well as the time in which the user's trip will commence in order to properly display predicted traffic trends in a map of the area. Additionally, as seen in the bottom right corner of the figure, there will also be an information section that will showcase any auxiliary information about the user's route. This information includes, but is not limited to, graphs of traffic severity vs. time, graphs of traffic severity vs. weather, and information about gas costs.

Functional Requirements Specifications

Stakeholders

- **Commercial Users** - The regular day-to-day user of our application. They use it to plan their trip and commutes in the future.

- **Business Users** - Delivery and transportation services who want to use our application to plan gas expenses and to minimize traffic for their drivers.

- **Administrators** - The individuals responsible for developing and maintaining the application.

Actors and Goals

Table 4: Actor and goal information

Actor	Role	Goal
User	Initiating	To obtain some information about traffic either in a general area or along a specific route.
Database	Participating	To hold historical records of traffic and weather in the tri-state area and to provide those records upon request.
Mapping Service	Participating	To return information about areas and routes connecting two points, and to provide a map image which corresponds to a set of coordinates.
Weather Service	Participating	To return information about the forecasted weather in a certain zip code area and along routes.
Gas Feed Service	Participating	To return information about the current gas prices for cost estimation.

Use Cases

Casual Descriptions

For the following use cases, we derive from the functional requirements.

- ***UC-1: ViewHeatMap*** - Allows a user to enter a city/zip code and day of the week/time and weather condition to allow the user to view a heat map of traffic for a specific day of the week and time. The user can also view statistical data about traffic in the area of interest.
  - Associated Actors: User, Mapping Service, Database
  - *Extension:* If the date/time of travel is within the next 10 days, the weather can be forecasted.
  - Derived from requirement REQ1, REQ2, REQ4 - REQ6

- ***UC-2: ViewRoute*** - Allows a user to enter the details of their route on a day of week and time, and allows the user to view route recommendations using the day of the week/time and weather condition provided by the user. The user can also view statistical data about traffic on the route of interest.
  - Associated Actors: User, Mapping Service, Database
  - *Extension*: The user can view alternate (detour) routes.
  - *Extension*: The user can view gas prices along their routes.
  - *Extension:* If the date/time of travel is within the next 10 days, the weather can be forecasted.
  - *Extension*: The user can view current traffic conditions along route.
  - Derived from requirement REQ7 - REQ9, REQ11, REQ12, REQ14, REQ16

- ***UC-3: CreateAgenda*** - Allows a user to enter three or more destinations and a time frame to suggest the best time/order in which to visit each destination using the day of week/time and weather condition provided by the user.
  - Associated Actors: User, Mapping Service, Database
  - *Extension*: The user can view gas prices along their routes.
  - *Extension:* If the date/time of travel is within the next 10 days, the weather can be forecasted.
  - Derived from requirement REQ7, REQ9, REQ13-REQ16

- ***UC-4: ViewDetourInfo*** - Allows a user to view a suggested alternate detour route. (optional sub use case, «extends» UC-2: ViewRoute).
  - Associated Actors: User, Mapping Service
  - Derived from requirement REQ10

- ***UC-5: ViewGasCost*** - Allows a user to view the projected gas cost of his/her trip . (optional sub use case, «extends» UC-2: ViewRoute, «extends» UC-4: *CreateAgenda*).
  - Associated Actors: User, Gas Feed Service
  - Derived from requirement REQ17

- ***UC-6: ViewForecastInfo*** - Allows a user to take into account the forecasted weather over the next ten days instead of the user inputting these conditions. (optional sub use case, «extends» UC-1: ViewHeatMap, «extends» UC-2: ViewRoute, «extends» UC-3: *CreateAgenda*).
  - Associated Actors: User, Weather Service
  - Derived from requirement REQ3

- ***UC-7: ViewCurrentRoute*** - Allows a user to view the current traffic conditions along their specified route for immediate travel (optional sub use case, «extends» UC-2: ViewRoute).
  - Associated Actors: User, Mapping Service
  - Derived from requirement REQ18

Use Case Diagram

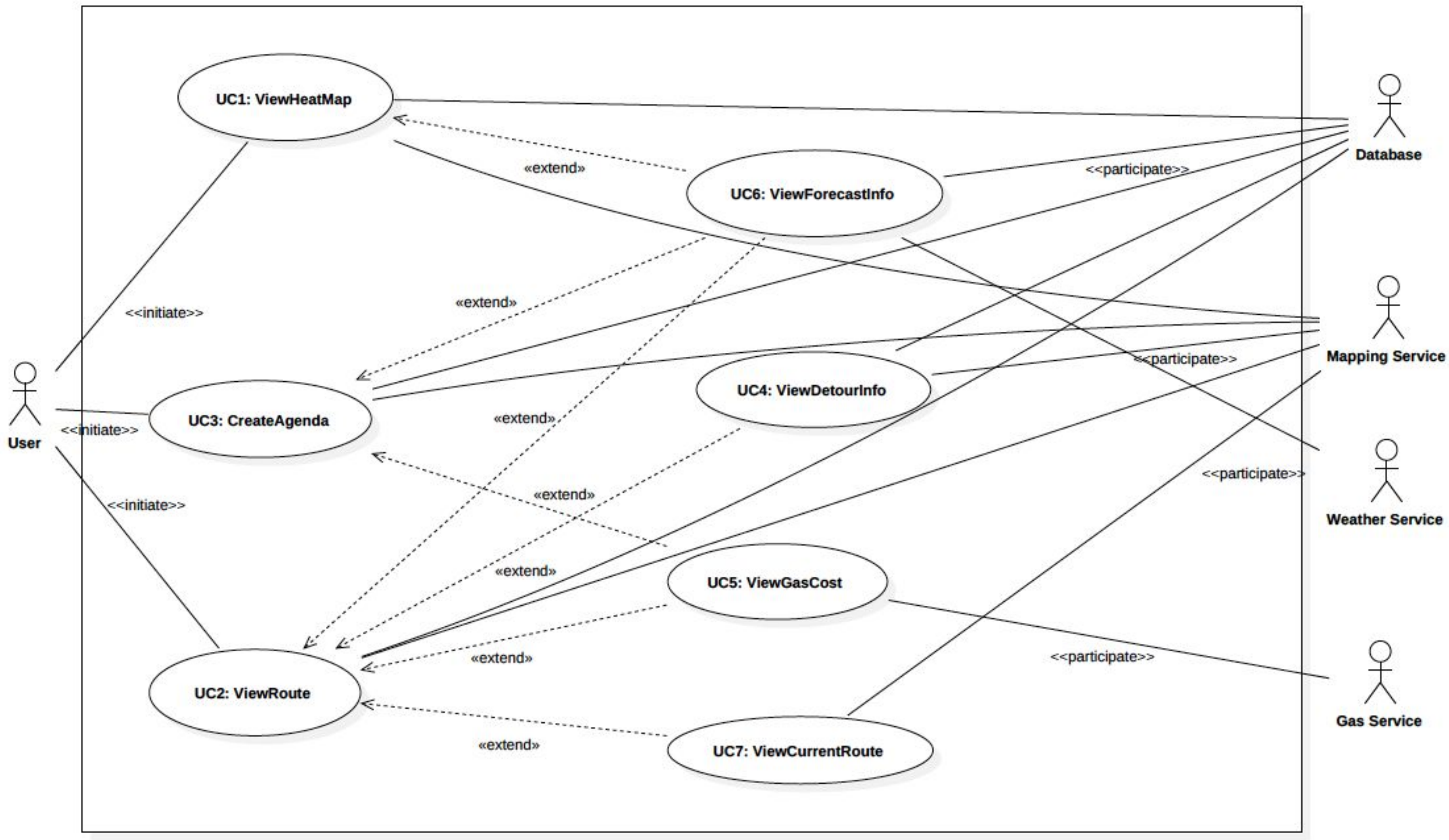

Figure 2: Use Case Diagram

The use case diagram in Figure 2 depicts the interaction of the user with the use-cases as well as the supporting actors: Database, Mapping Service, Weather Service, and Gas Service. UC1, UC2 and UC3 have multiple extensions, UC4-UC7. At first, we considered making a use case for every possible combination of inputs and scenarios. For example, a use case could have been viewing the heat map without forecasted weather information, and another use case could have been viewing the heat map with forecasted weather information. This approach however seemed to be clunky and redundant. Instead we made certain features extended use cases of the base cases for greater simplicity and understanding.

Traceability Matrix

REQ	PW	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7
REQ1	5	X						
REQ2	3	X						
REQ3	3						X	
REQ4	2	X						
REQ5	5	X						
REQ6	2	X						
REQ7	5		X	X				
REQ8	2		X					
REQ9	5		X	X				
REQ10	3				X			
REQ11	2		X					
REQ12	2		X					
REQ13	3			X				
REQ14	3		X	X				
REQ15	3			X				
REQ16	2		X	X				
REQ17	4					X		
REQ18	2							X
Max PW		5	5	5	3	4	3	2
Total PW		17	21	21	3	4	3	2

Figure 3: Use Case Traceability Matrix

Note: Nonfunctional requirements are not included as they apply to all use cases and we did not want to complicate the traceability matrix with extra information. UC-3 - UC-7 are extended use cases. They extended upon base cases UC-1 - UC-3, which is why the PW points of UC-4 -UC-7 are lower.

Use Cases Fully-Dressed Descriptions

Use Case UC-1:		**ViewHeatMap**
Related Requirements:		REQ1, REQ2, REQ4, REQ5, and REQ6
Initiating Actor:		User
Actor's Goal:		To view a heat map and statistical data about traffic around a specific location and specific time.
Participating Actors:		Database, Mapping Service
Preconditions:		- The system displays a menu of available functions: Heat Map, Route, and Agenda. - The traffic and weather data in the database is not empty for the user-specified location.
Postconditions:		- The system displays a map with traffic highlighted with selected severities. - The system displays a graph of statistical traffic data.
Flow of Events for Main Success Scenario		
→	1	**User** arrives on landing page and selects the “Heat Map” option.
←	2	**System** displays a map and input fields for zip code or city, range of miles, day of the week, time, weather conditions, and severity levels.
→	3	**User** inputs zip code or city, range of miles, day of week, time and a general weather condition and severity level.
←	4	**System** signals (a) **mapping service** to get mapping data and (b) **database** to fetch necessary traffic and weather data to analyze conditions of the area for the heat map.
←	5	**System** displays a map with highlighted traffic according to severity and a graph of statistical traffic data.
Flow of Events for Extensions (Alternate Scenarios):		
3a.**User** enters invalid information		
←	1	**System** resets the input field and asks for appropriate input.
3.b **User** inputs date/time within 10 days of the current date. Extends UC-6 ViewForecastInfo.		
←	1.	**System** displays map with highlighted traffic according to the severity and a graph of statistical traffic data based on the forecasted weather data for the user’s specified date/time.

Note: The alternate scenario 3.a is for invalid information input; this can only apply to an invalid zip code/city entered. All other user inputs will be limited, for example with dropdowns and checkboxes, so that the user cannot enter an invalid input. Scenario 3.b is of the extended use case where the user chooses an option to use forecasted weather if their travel is in the next 10 days. This scenario is further described in UC-6 description.

Use Case UC-2:		ViewRoute
Related Requirements:		REQ7, REQ8, REQ9, REQ11, REQ12, REQ14, REQ16
Initiating Actor:		User
Actor's Goal:		To view a route from starting point to destination with traffic information and statistical data along the route.
Participating Actors:		Database, Mapping Service
Preconditions:		- The system displays a menu of available functions: Heat Map, Route, and Agenda. - The traffic and weather data in the database is not empty for the user-specified location.
Postconditions:		- The system suggests a route on a map with traffic highlighted with selected severities along the route. - The system displays a graph of statistical traffic data along the route.
Flow of Events for Main Success Scenario		
→	1	**User** arrives on landing page and selects the “Route” option.
←	2	**System** displays a map and input fields for starting point, destination, day of the week, time, weather conditions, and severity levels.
→	3	**User** inputs starting point, destination, day of week, time and a general weather condition and severity level.
←	4	**System** signals (a) **mapping service** to get mapping and route data from starting point to destination and (b) **database** to fetch necessary traffic and weather data to analyze conditions of the route..
←	5	**System** displays a map with highlighted route and highlighted traffic according to checked severity levels and a graph of statistical traffic data.
Flow of Events for Extensions (Alternate Scenarios):		
3a.**User** enters invalid information		
←	1	**System** resets the input field and asks for appropriate input.
3.b **User** inputs date/time within 10 days of the current date. Extends UC-6 ViewForecastInfo.		
←	1	**System** displays map with highlighted traffic according to the severity and a graph of statistical traffic data based on the forecasted weather data for the user’s specified date/time.
6. **User** chooses to view a detour route. Extends UC-4 ViewDetourInfo.		
→	1	**System** suggests an alternative route to take that has less traffic.
7. **User** chooses to show gas information along the suggested route. Extends UC-5 ViewGasCost.		

←		1	**System** displays projected gas costs associated with the route (and detour routes).

Note: The alternate scenario 3.a is for invalid information input; this can only apply to an invalid addresses entered. All other user inputs will be limited, for example with dropdowns and checkboxes, so that the user cannot enter an invalid input. Scenario 3.b,6 and 7 is of the extended use cases that are further explained in their own use case descriptions. Use cases UC-4 and UC-5 are not fully dressed in this report.

Use Case UC-6:			**ViewForecastInfo** (sub-use case)
Related Requirements:			REQ3
Initiating Actor:			User
Actor's Goal:			To use forecasted weather for their location/route for a date/time in the next 10 days as a parameter for the heatmap, route, and agenda features.
Participating Actors:			Database, Weather Service
Preconditions:			- A user is at either “heat map”, “route” or “agenda” page. - The weather service has a forecast available for a date within the next 10 days for the location specified.
Postconditions:			- The system uses forecasted weather for the traffic analysis.
Flow of Events for Main Success Scenario			
→		1	**User** selects option to use weather that is forecasted for next 10 days.
→		2	**User** inputs date and time that is the next 10 day along with other input parameters.
←		3	**System** fetches the weather data from **weather service** for the date/time in the location specified.
←		4	**System** goes back to Step 4 of UC-1,UC-2, or UC-3* and uses this information for traffic analysis.

*UC-3 is not fully-dressed in this report, as it will not be part of demo 1, but will be part of demo 2.

Note: This use case extends the three base cases UC-1, UC-2 and UC-3. This option exists for all 3 base cases.

System Sequences Diagrams

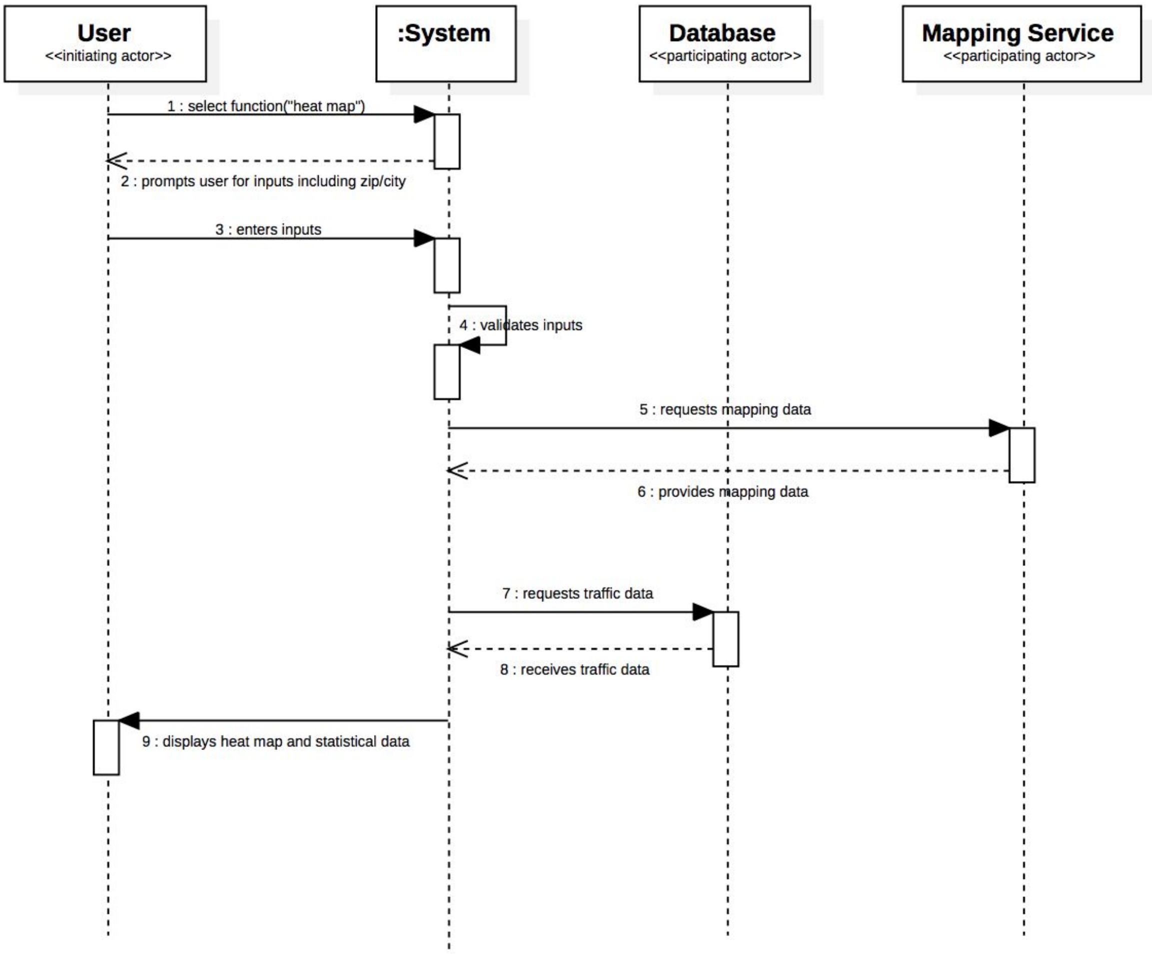

Figure 4: UC-1 Main Success Scenario Sequence Diagram

Figure 3 is a sequence diagram of the main success scenario involving the heat map feature base case, UC-1. The inputs prompted to user and inputted in steps 2 and 3 include zip code or city, range of miles, day of the week, time, weather conditions, and severity levels. This sequence diagram illustrates when the user does not use forecasted weather. The numbering of the sequence diagram is not intended to match the fully dressed description; the numbering here was done by the UML tool used.

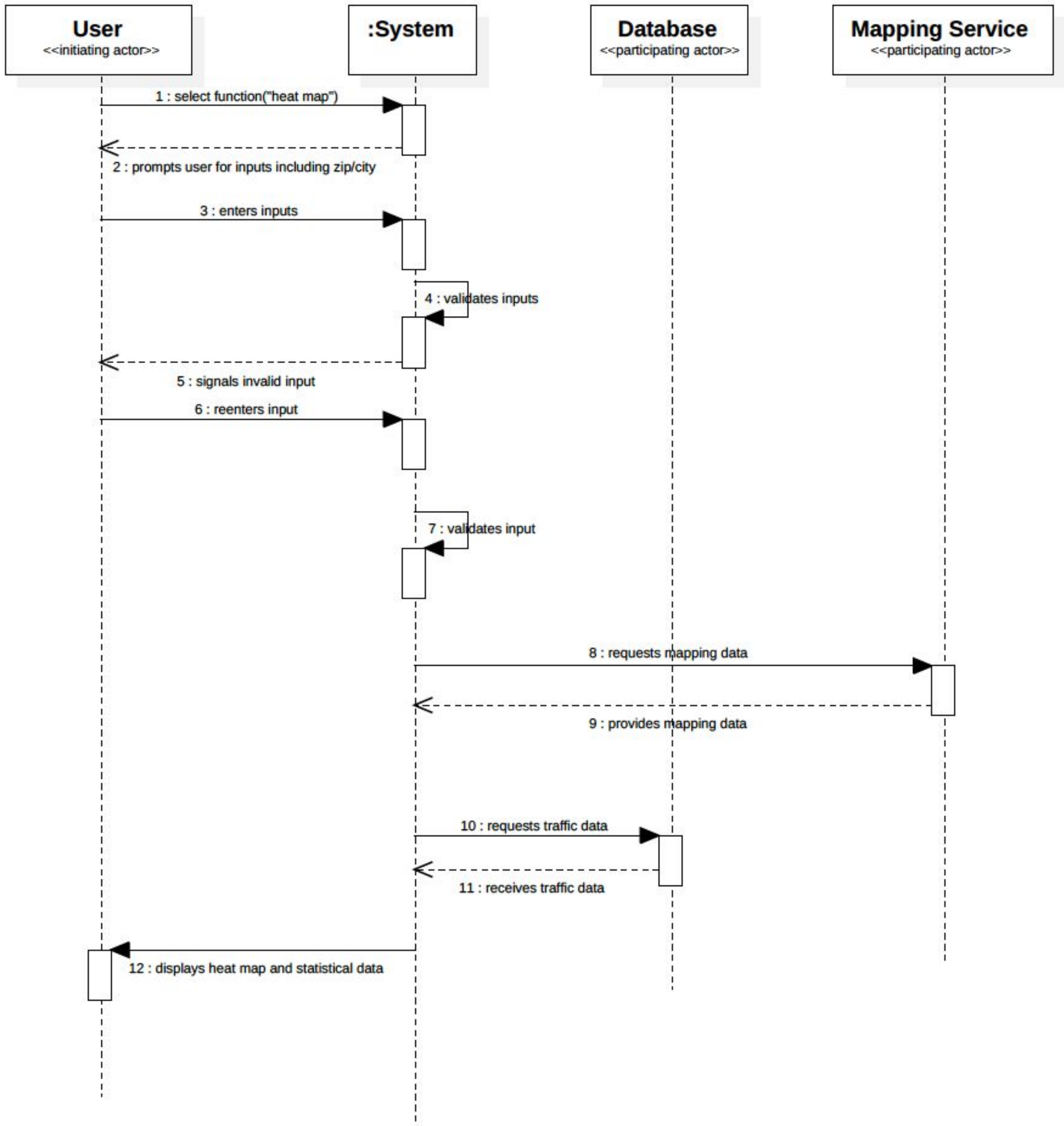

Figure 5: UC-1 Alternate Scenario Sequence Diagram

Figure 4 is a sequence diagram of the alternate scenario involving the heat map feature base case, UC-1, where the user enters an invalid input. The only invalid input the user could enter is city or zip code because the remainder of the inputs will be limited by the user interface. The inputs prompted to the user and inputted in steps 2, 3 and 6 include zip code or city, range of miles, day of the week, time, weather conditions, and severity levels.

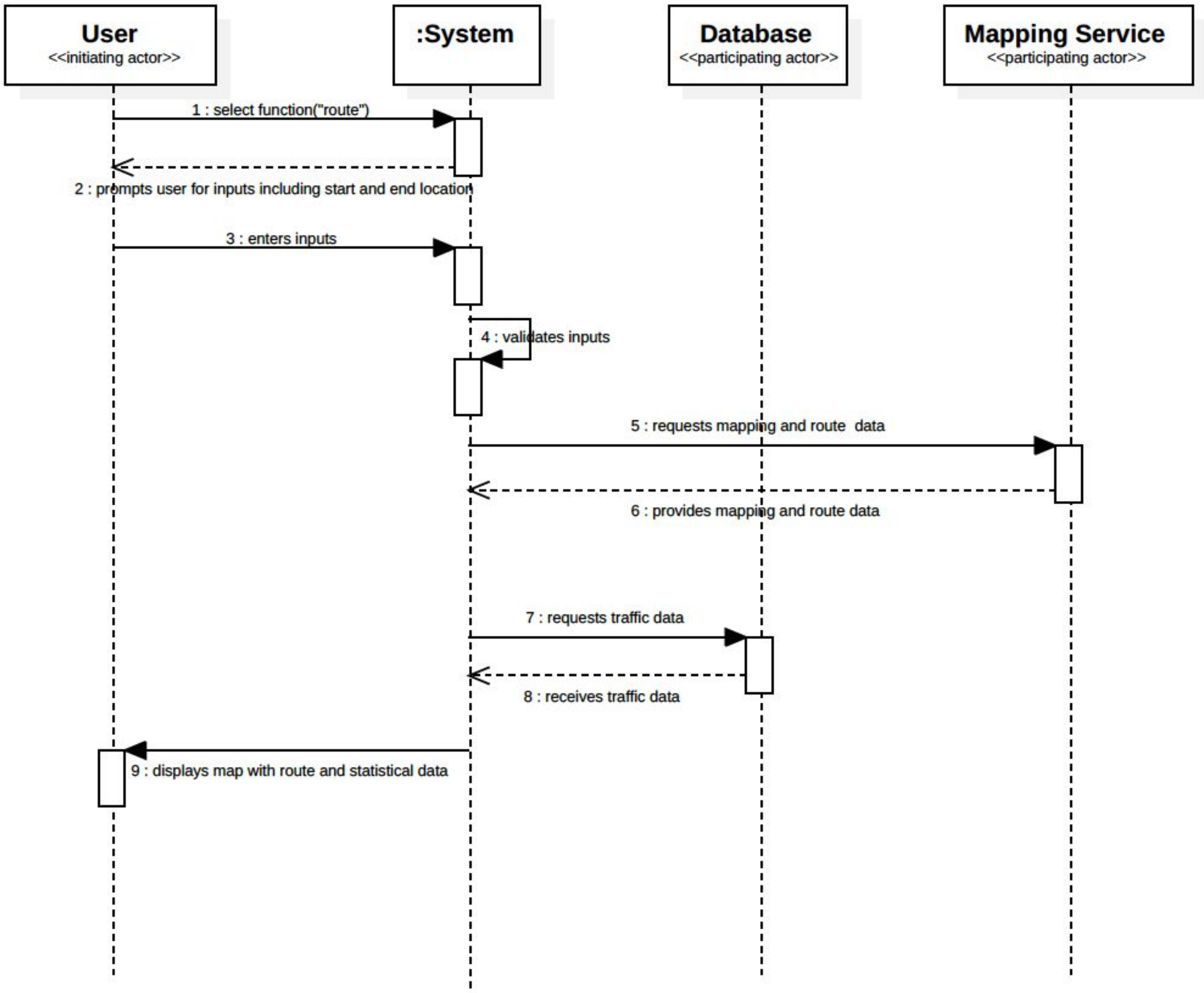

Figure 6: UC-2 Main Success Scenario Sequence Diagram

Figure 5 is a sequence diagram of the main success scenario involving the route feature base case, UC-2. The inputs prompted to user and inputted in steps 2 and 3 include starting point, destination, day of the week, time, weather conditions, and severity levels. This sequence diagram illustrates when the user does not use forecasted weather.

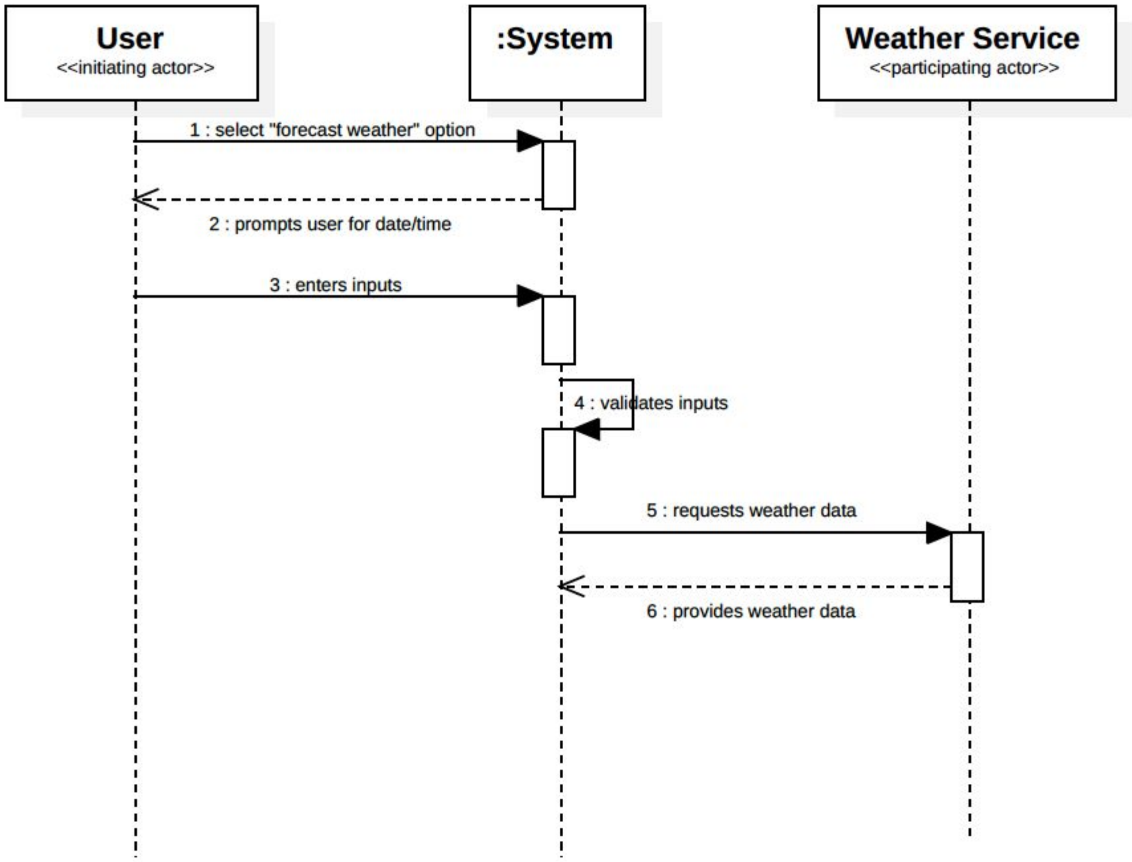

Figure 7: UC-6 Main Success Scenario Sequence Diagram

Figure 6 is the main success scenario sequence diagram for UC-6, where forecasted weather is used. This sequence diagram can be used for UC-1, UC-2 or UC-3 where the user may want to use forecasted weather instead of the inputting weather as this is an extension or alternative scenario of all the base cases. The inputs in steps 2 and 3 are of a date that is in the next 10 days and a time. The option “forecast weather” is a generic phrasing of the option. It may be phrased differently in actual UI.

User Interface Specification

Preliminary User Interface Design

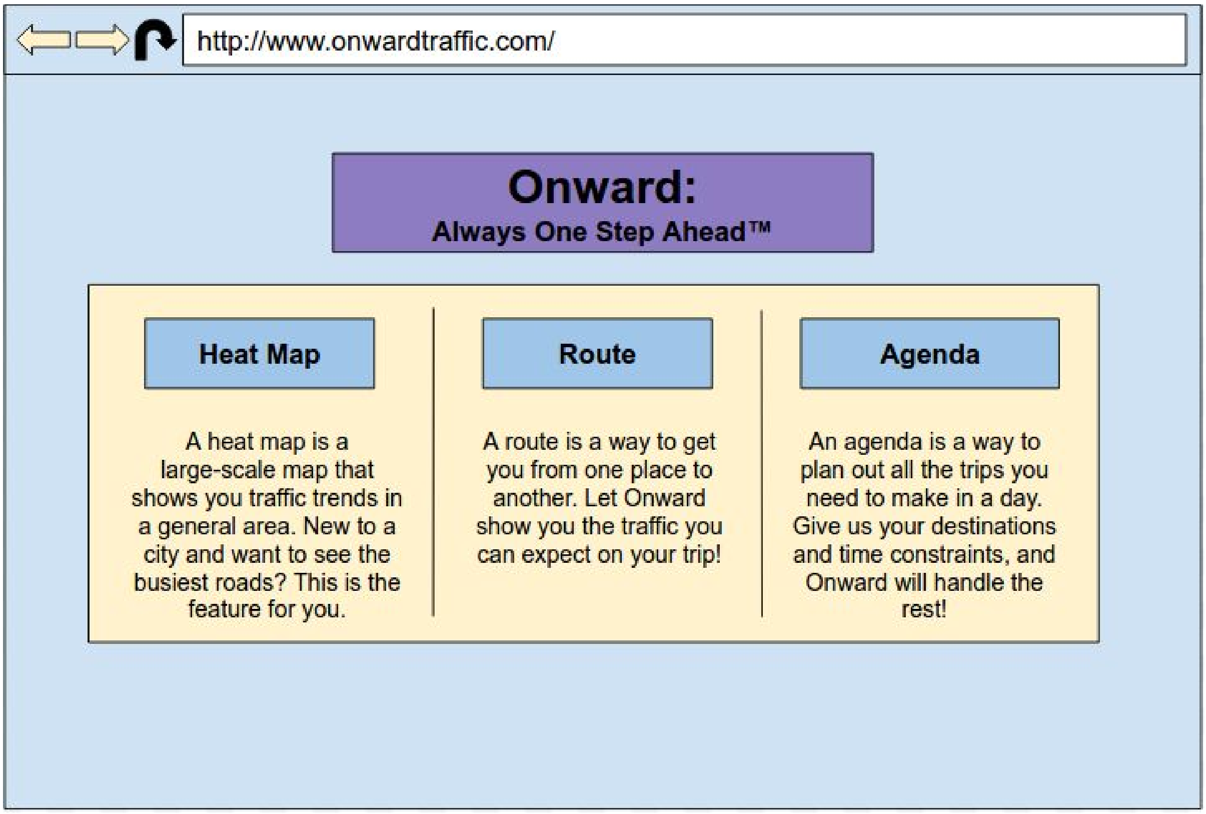

Figure 8: Landing Page

The above user interface shows the homepage of the Onward website. From here, the user can choose one of the features on the site, the heat map, the route or the agenda feature, based on the descriptions provided.

UC-1: ViewHeatMap

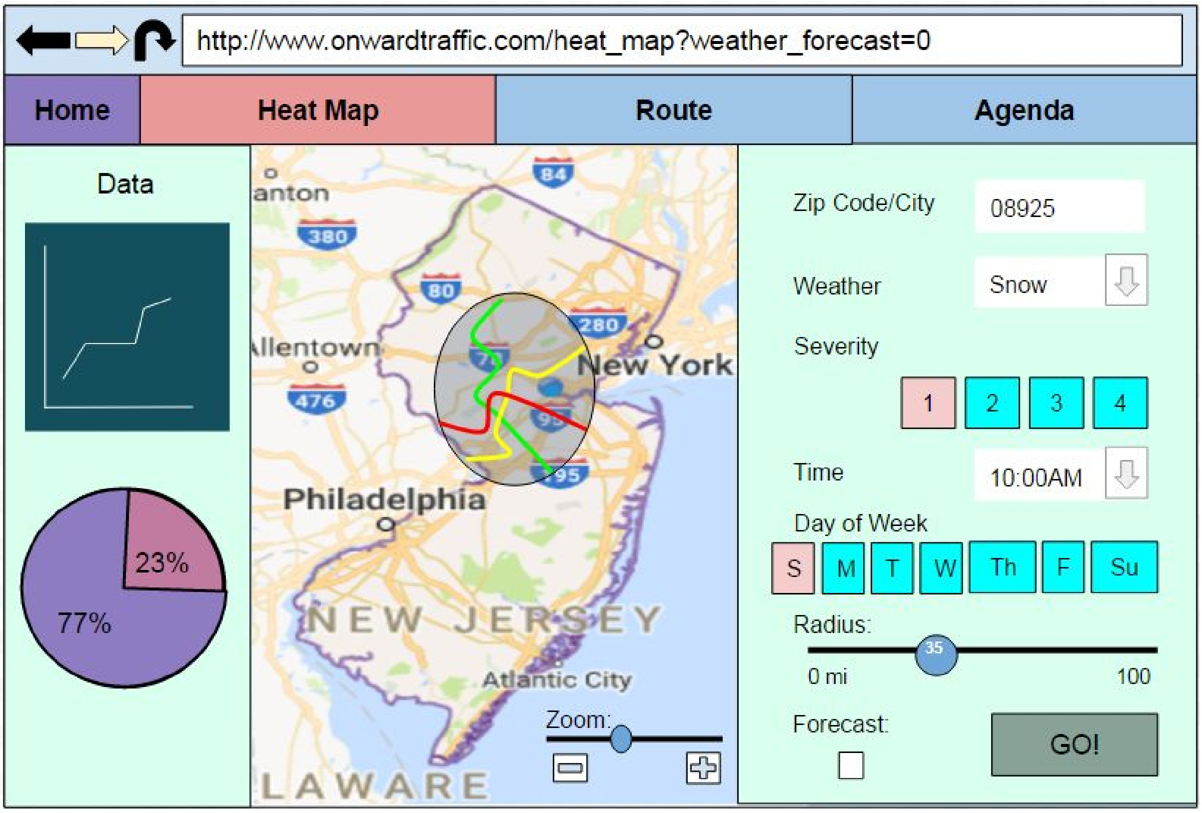

Figure 9: Heat Map

After moving past the landing page and answering the previous prompt, the heat map user will come to this page where they input all pertinent information regarding the area whose trends they want to see. They will have to enter the zip code, the weather (forecasted or non forecasted weather), the traffic severity to display, the time of day, the day of the week, and a slider for the radius around the zip code. The application will then display a circle around the selected area on the map in the middle of the page. The user can manipulate the map through either scrolling or zooming. All major roads in this circle will be highlighted, with color segments indicating the severity of the traffic along those segments. On the left side of the page, the user has access to some visualizations of data related to the traffic trends in the area shown. This data will be visible on the desktop version of the application, but removed on the mobile version in order to remove visual clutter for the mobile user.

UC-6: ViewForecastInfo

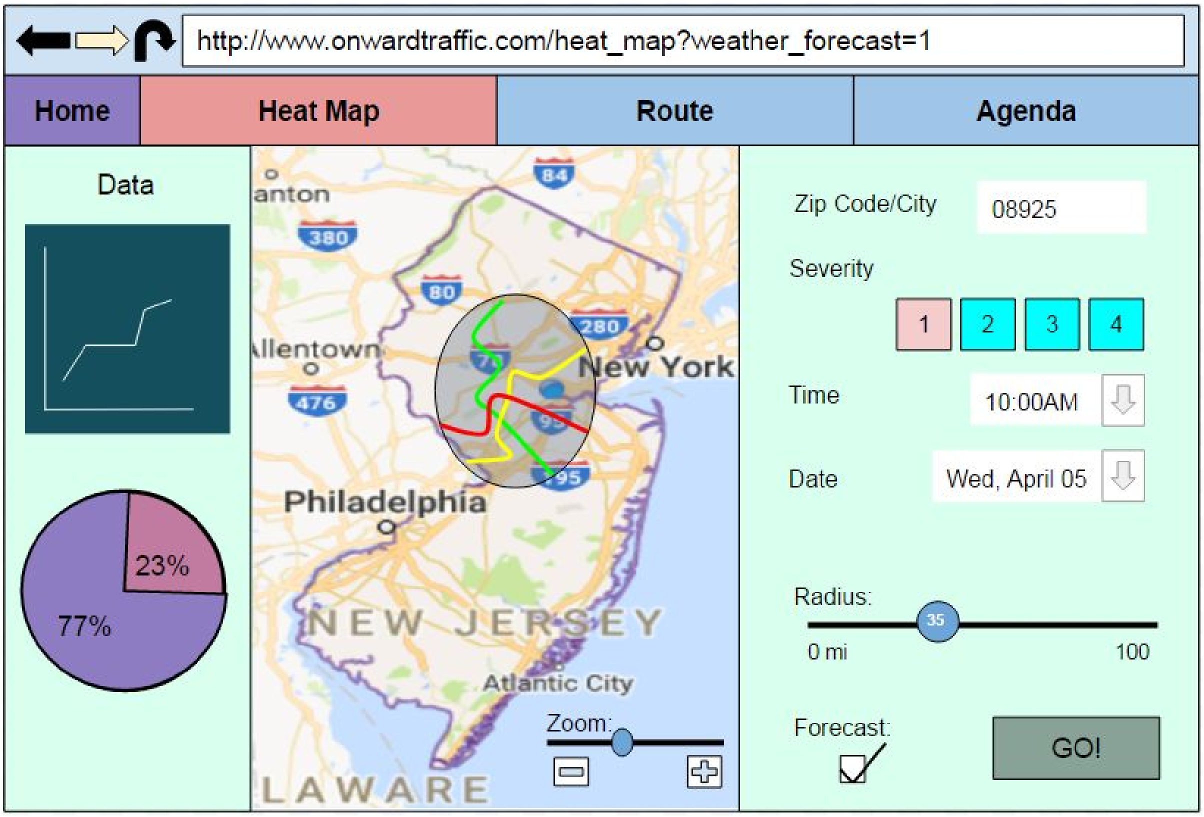

Figure 10: Heat Map with Weather Forecast

This use case is very similar to the previous one. This user, however, wants to see traffic predicted for a specific day in the future, as opposed to general traffic trends. This extension use case will take advantage of weather forecasts to free the user from entering weather themselves. It also replaces the 'day of the week' selector with an exact date drop-down box that will have an option of 10 days, which allows the user to view traffic severity exactly as our service predicts it to be on that day. Other than this, the UI behaves in the same way as the previous Heat Map use case. We have used the Heat Map use case to illustrate this feature, however this also extends to the "Route" and "Agenda" features.

UC-2: ViewRoute

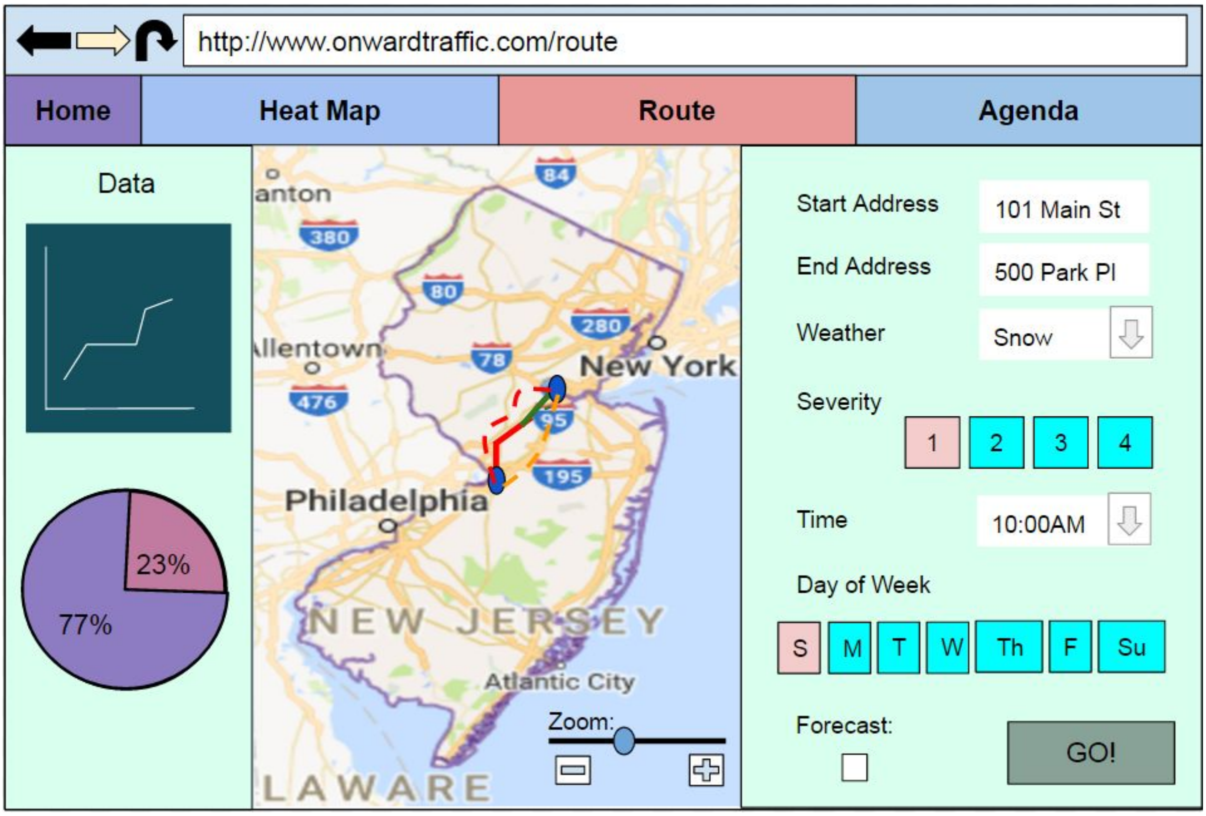

Figure 11: Route

After moving beyond the landing page, route feature users will come to this page. The right side of the page contains forms where the user will input pertinent parameters such as their start and end address, the weather (either a direct option ie. snow, rain, etc. or following a forecast), and the traffic severity to display. The middle of the application will be the focal point, showing the user several calculated routes. Users can manipulate the map display through both zooming and scrolling. The solid route is the suggested route. The alternative routes are displayed as dashed lines. Segments of color on the routes indicate the severity of traffic along that segment. Selecting one of these routes will cause pertinent data about traffic trends along that route to display on the left. This data only displays on the desktop version, so as to avoid visual clutter for the mobile user.

User Effort Estimation

UC-1 (Assuming that the user has successfully navigated to the home page of the website)

1. Navigation: Total 2 mouse clicks as follows:
    a. Click "Heat Map"
    --after completing data entry as shown below--
    b. Click "Go"
2. Data Entry: Total 9 mouse clicks and 5 keystrokes as follows:
    a. Click cursor to "Zip Code/City" text field
    b. Press five keys which correspond to the zip code i.e. "08925"
    c. Click an option to enter in the user's predicted weather
    d. Click the dropdown button next to weather to display weather options
    e. Click the desired weather option i.e. "Snow"
    f. Click on the severity of the traffic incidents displayed i.e. "1"
    g. Click the dropdown menu next to Time
    h. Click the desired time frame in which to display the Heat Map
    i. Click on the day of the week to show i.e. "Sunday"
    j. Click and drag the button to show the radius of traffic incidents to be displayed

Note: If a user inputs a city, then depending on the city name (ie. the number of letters in the city's name) the keystroke count will be different.

UC-6 (Assuming that the user has successfully navigated to the home page of the website)

1. Navigation: Total 2 mouse clicks as follows:
    a. Click "Heat Map"
    --after completing data entry as shown below--
    b. Click "Go"
2. Data Entry: Total 7 mouse clicks and 5 keystrokes as follows:
    a. Click cursor to "Zip Code/City" text field
    b. Press five keys which correspond to the zip code i.e. "08925"
    c. Click on the severity of the traffic incidents displayed i.e. "1"
    d. Click an option to use the forecasted weather from the next 10 days
    e. Click the dropdown menu next to Time
    f. Click the desired time frame in which to display the Heat Map
    g. Click the dropdown menu next to Date
    h. Click the desired date to display i.e. "Wed, April 5"
    i. Click and drag the button to show the radius of traffic incidents to be displayed

Note: For the purpose of illustrating the UC-6 we showed the usage with the UC-1, Heat Map. But it could also be used with the "Route" and "Agenda" features, UC-2 and UC-3 in a similar manner. If a user inputs a city, then depending on the city name the keystroke count will be different.

UC-2 (Assuming that the user has successfully navigated to the home page of the website)

1. Navigation: Total 2 mouse clicks as follows:
    a. Click “Route”
    --after completing data entry as shown below--
    b. Click “Go”
2. Data Entry: Total 8 mouse clicks and 21 keystrokes as follows:
    a. Click cursor to “Start address” text field
    b. Press the keys which correspond to the address i.e. 101 Main St
    c. Click cursor to “End address” text field
    d. Press the keys which correspond to the address i.e. 500 Park Pl
    e. Click an option to enter in the user’s predicted weather
    f. Click the dropdown button next to weather to display weather options
    g. Click the desired weather option i.e. “Snow”
    h. Click the dropdown menu next to Time
    i. Click the desired time in which to display the Route
    j. Click on the day of the week to show i.e. “Sunday”

Note: Keystroke count depends on the address inputted.

For the above effort estimations, the worst case scenario is that the user would re-enter their zip code/city or addresses which would increase keystrokes.

Domain Analysis

Our domain analysis for the use cases UC-1,UC-2, and UC-6. We will provide the domain analysis for these as they are our fully dressed use cases.

Domain Model

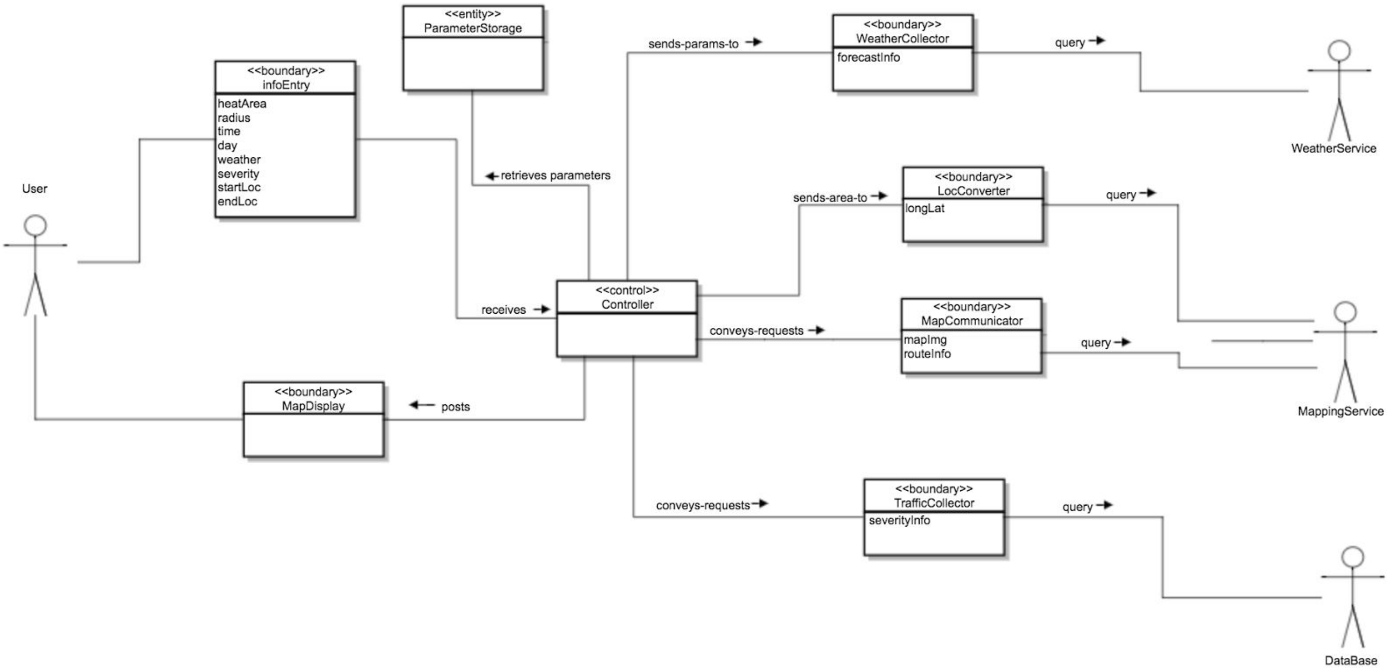

Figure 12: Domain Model for UC-1, UC-2, UC-6

To construct our domain model, we began by adding the Controller and the InfoEntry. We started here because the user's inputs are the natural starting point of all the use cases. From there, we developed the other concepts as they were needed to fulfill the use cases. The Controller receives these inputs from InfoEntry and stores them in the entity ParameterStorage. We added this entity because we knew we should be storing these parameters for the duration of the user's actions. Originally, we considered having the Controller directly interact with the actors. For the sake of cohesion, however, we decided to assign the job of actor-interaction to separate concepts. The Controller then passes the address-formatted location to the LocationConverter, which uses the Mapping Service to convert the location to a longitude and latitude. This longitude and latitude is used to query our Database and gather traffic incidents in the desired area. This traffic severity info is then passed along with the latitude and longitude info to the MapCommunicator. The MapCommunicator then uses this info to acquire a map image from the Mapping Service. Then the MapDisplay displays that image to the user.

Concept Definitions

Table 5: Concept Definitions for UC-1, UC-2, and UC-6

Responsibility	Type	Concept
R1: Coordinate the receiving and sending of info to and from the different concepts.	D	Controller
R2: Accept user input for all of the various parameters.	D	Info Entry
R3: Keep track of the user-inputted parameters.	K	Parameter Storage
R3: Take locations in address format and convert them to latitudinal-longitudinal format using a Mapping Service.	D	Location Converter
R4: Take latitudinal-longitudinal format locations and query the Database for all traffic severities that match the locations.	D	Traffic Collector
R5: Take traffic severities and latitudinal-longitudinal locations and use those to obtain a map image from the Mapping Service.	D	Map Communicator
R6: Take date, time, and location, and use those to obtain a weather forecast from the Weather Service.	D	Weather Collector
R7: Take a map image and display it to the user.	D	Map Display

Association Definitions

Table 6: Association Definitions for UC-1, UC-2, and UC-6

Concept Pair	Association Description	Association Name
Info Entry ↔ Controller	User enters data relevant to their needs. The data is sent to the controller.	Conveys Requests
Location Converter ↔ Controller	The controller sends location request from the user to the Location Converter. The Location Converter returns either the bounding box for a heat map or the route.	Provides Data
Map Communicator ↔ Controller	Controller sends time/location data needed to Map Communicator. Map Communicator returns map data from the Mapping service to the Controller.	Provides Data
Traffic Collector ↔	Controller sends time/date/location	Provides Data

Controller	information needed from database and Traffic Collector queries the data. Traffic Collector returns the subset of data needed from the database.	
Map Display ↔ Controller	Controller sends data compiled from the the other concepts and Map Display uses the data to display a map with the data to the user.	Displays Data
Controller ↔ Parameter Storage	The Controller sends information received from other concepts to Parameter Storage for storage.	Stores Data
Extension: Weather Collector ↔ Controller	If the user chooses to use forecasted weather data, the Controller sends the information needed to the Weather Collector. The Weather Collector returns the weather forecast from the Weather Service.	Provides Data

Attribute Definitions

Table 7: Attribute Definitions for UC-1, UC-2, and UC-6

Responsibility	Attribute	Concept
R8: User inputted location of the area that the heat map will be centered around	heatArea	infoEntry
R9: User inputted radius of the area that the heat map will display	radius	
R10: User inputted time of day that heat map statistics will be based off of	time	
R11: User inputted day of week that heat map statistics will be based off of	day	
R12: User inputted weather type that heat map statistics will be based off of (snow, clear, etc.)	weather	
R13: User inputted severity of traffic that should be displayed on the heat map	severity	
R14: User's starting location for route feature	startLoc	
R15: User's ending location for route feature	endLoc	
R16: Weather forecast for date and time specified	forecastInfo	WeatherCollector

by user		
R17: Acquired longitude and latitude converted from the inputted zip code	longLat	LocConverter
R18: Map with highlighted routes and traffic information to be displayed to the user	mapImg	MapCommunicator
R19: Suggested route to get from starting point to destination specified by user	routeInfo	
R20: Traffic information acquired from the database to be displayed on the map	severityInfo	TrafficCollector

Traceability Matrix

<table>
<tr><td rowspan="11">DOMAIN CONCEPTS</td><td>Use Case</td><td>UC-1</td><td>UC-2</td><td>UC-6</td></tr>
<tr><td>PW</td><td>17</td><td>21</td><td>3</td></tr>
<tr><td>Controller</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Info Entry</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Parameter Storage</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Location Converter</td><td>X</td><td>X</td><td></td></tr>
<tr><td>Traffic Collector</td><td>X</td><td>X</td><td></td></tr>
<tr><td>Map Communicator</td><td>X</td><td>X</td><td></td></tr>
<tr><td>Weather Collector</td><td></td><td></td><td>X</td></tr>
<tr><td>Map Display</td><td>X</td><td>X</td><td></td></tr>
</table>

Figure 13: Traceability Matrix of Domain Model

In Figure 13 we have shown the traceability matrix for our fully dressed use cases.

System Operation Contracts

Table 8: Operation Contract for UC-1

Operation	viewHeatMap
Preconditions	● *heatArea* must be specified by the user. Must be a valid input. ● *radius* of interest must be specified by the user. ● *severity* must be specified by the user. ● Route information for the specified *heatArea* must exist in the database.
Postconditions	● *mapImg* must be displayed for the user. ● *severityInfo* must be displayed on the map on the *heatArea*. ● *severityInfo* analysis must be displayed on a graph.

Table 9: Operation Contract for UC-2

Operation	viewRoute
Preconditions	● *time* must be specified by the user. ● *day* of travel must be specified by the user. ● *weather* must be specified by the user. ● *startLoc* must be specified by the user. ● *endLoc* must be specified by the user. ● *severity* must be specified by the user. ● Route data must exist in the database.
Postconditions	● *routeInfo* must be displayed for the user. ● *severityInfo* must be displayed on the map along the route. ● *severityInfo* analysis must be displayed on a graph.

Table 10: Operation Contract for UC-6

Operation	viewForecastInfo
Preconditions	● Weather API must have the weather forecast for the next ten days available. ● Weather info must exist in the database.
Postconditions	● *forecastInfo* must be incorporated with the user info to display the proper output.

Mathematical Model

The most pressing question we had to deal with regarding mathematical models was: how do we predict the severity of a road based on historical data? In order to do this, we look at all the incidents recorded along a given road, time, and set of weather conditions. In order to scale our results, we take the sum of the severities of these incidents and divide it by the total amount of possible incidents. Below is a formalized representation of the algorithm that we will be employing to calculate the average severity for a road at a certain datetime.

$$A_{x,t} = \frac{\sum_{i=0}^{N} \tau(x,t)}{M} \qquad (1)$$

Where:

- $A_{x,t}$ is the average severity along a given road *x* at datetime *t* (a specific hour on a specific day of the week).
- N is the total number of traffic incidents for a given road *x* at datetime *t* (a specific hour on a specific day of the week).
- M is the total number of times the cron job has ran since data collection has commenced for a given road *x* at datetime *t* (a specific hour on a specific day of the week).
- $\tau(x,t)$ is a function that fetches the severity for each traffic incident for a given road *x* at datetime *t* (a specific hour on a specific day of the week).

All traffic incidents in our database are assigned a corresponding severity level from 1-4. In equation (1), all the severities of traffic incidents that occurred on road *x*, at a specific time *t* are summed. This total is then divided by the number of times the cron job has ran for that specific scenario (road and datetime) since data collection commenced. This will ensure that the result is scaled with consideration for the null case where no traffic is recorded at a given time. Ultimately, this equation now calculates the average severity level for road *x* at datetime *t*, and accounts for the null case. The average severity is expected to be very small using this formula, less than 1.0. To make these numbers easier to understand, we will scale them up so that the averages reported to the user are either 1,2,3, or 4.

Plan of Work

After the submission of report #1 we will develop our basic infrastructure and the setup of our database and preliminary design of our web application within the next week. We will then begin the development of the features associated with the use cases UC-1, UC-2, and UC-6 for our demo #1 on March 24th. After demo #2 we will then work on development of features associated with UC-3,UC-4,UC-5 and UC-7. This timeline is better illustrated in the Gantt chart below.

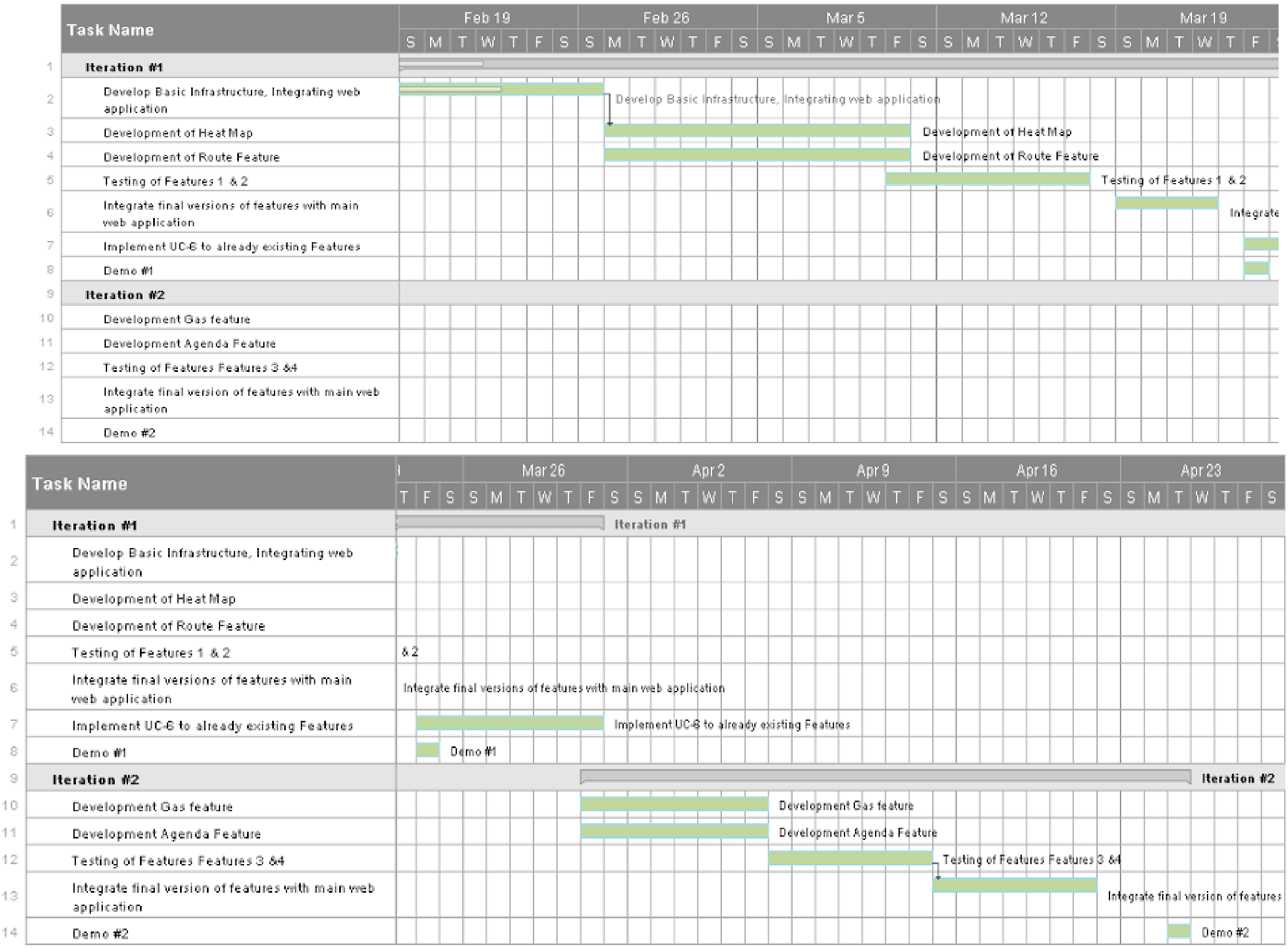

Figure 14. Gantt Chart

The team will be working in 2 subteams that will deliver the following features according to the Gantt chart.

Subteam 1:

Lauren Williams
Shubhra Paradkar
Sean Olejar

This subteam will be initially responsible for constructing the route suggestion feature of Onward along with the use of forecasted weather for demo #1. We believe this team has the ability to develop this feature based on the team members' experience implementing shortest-path algorithms. The team also has the experience required to both implement a visual interface for the feature and to effectively interact with the database. After demo #1, this team is tentatively scheduled to develop the agenda feature.

Subteam 2:

Mhammed Alhayek
Ridwan Khan
Brian Monticello

This subteam will be initially responsible for the traffic severity heat-map feature along with the use of forecasted weather for demo #1. We believe this team has the ability to effectively develop this feature based on the team members' experience in visualising data. The team members also have the experience required to implement a visual interface for their feature and interact with the database, much like Subteam 1. After demo #1, this team is then tentatively schedule to develop the least cost of travel feature.

Project Management

All team members are evenly responsible for project management. Version control is being established through the use of Git. Though we are using a managed workflow, all team members are responsible for unanimously reviewing and accepting pull requests. All team members are working equally in scheduling and participating in meetings, distributing and coordinating work and developing the system.

References

- Bing Traffic API - *Bing Developer Network.* 2017. <https://msdn.microsoft.com/en-us/library/hh441725.aspx>
- Current Weather Data - *Weather Underground API*. 2017. <https://www.wunderground.com/weather/api/>
- myGasFeed API - *JGSolutions.* 2010. <http://www.mygasfeed.com/keys/api>

The above references are the current API's being used to collect data and that will be used for future implementation.

Appendix

At the very start of this project we had created a Google Form to get feedback on what users would like to better improve mapping and traffic related applications, http://bit.ly/2l9YX2s. We received 58 responses and used these responses to shape our customer statement of requirements and helped us shape the requirements of our project. Some of the most popular responses about how to improve mapping apps were the ability to add multiple stops to the route feature and suggestions on when to leave to avoid rush hour (i.e. when will rush hour start/end). We took into account these responses when we created the agenda feature.