Software Engineering 332:452

Traffic Monitoring Project

Group 13 – Feb 22, 201

http://tingogame.me/TrafficMonitoring/

Maxine Dienes

Charu Jain

Mehul Salhotra

Akshay Sardana

Justin Silang

Jason Yang

Table of Contents

Responsibility Matrix

Responsibilities	Maxine	Charu	Mehul	Akshay	Justin	Jason
Cover Page (0)		100%				
Customer Statement of Requirements (9)			40%	40%		10%
System Requirements (6)	40%				20%	40%
Functional Requirements Specification (30)	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%
User Interface Specification (15)	30%	25%	10%	10%	10%	15%
Domain Analysis (25)	20%		20%	20%	20%	20%
Project Management (10)		55%			45%	
Plan of Work (5)		50%	25%	25%		
References (0)	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%
Total	**16.9%**	**16.75%**	**16.35%**	**16.35%**	**17.2%**	**15.55%**

Customer Statement of Requirements

Problem Statement

The Problem

If there is one daily aspect of the modern lifestyle that people all around the world express annoyances over, it is undoubtedly being stuck in traffic. Perhaps you are a university student running late to your class, and every traffic light seems to change to red before you. Or maybe an emergency forces you to drive to the hospital, but every red light delays your treatment minute by minute. Whatever the case may be, everyone who has ridden a motor vehicle has experienced this frustrating and tedious phenomenon before. Traffic congestion poses problems not only in wasted time but also monetary costs in wasted fuel when your car is idle, to the tune of upwards of $755 annually per driver (Morgan). If traffic is such a wasteful and universal problem, shouldn't there be a way to reduce the frequency of it, or at least avoid it yourself?

The Solution

Traffic congestion will always be inevitable, but there can be methods to reduce its severity. Currently, services exist for individuals to avoid areas of heavy traffic. Popular map services such as Google Maps and Mapquest Traffic give information about traffic only based on current accidents, construction, road closures, and other incidents. Most of these services collect data in intervals and releases them to show current traffic congestion. While it is essential to provide current traffic information for users, it often becomes unreliable because these services frequently fail to provide up-to-date and accurate traffic information. Instead, it is often more helpful to incorporate an archive of historical traffic conditions for where the user requests.

Thus, the reliability of historical traffic data becomes the fulcrum of our proposed traffic application. Our project will stray away from only using live traffic reports to give information about traffic congestion. It will use this traffic data to utilize past traffic patterns in a particular area. The purpose of obtaining past traffic data is to enable us to locate areas that will most likely form traffic again and anticipate alternative routes early on, to create a faster and more efficient system. The traffic information can be further used to collect weather along routes and time of day for each data. We aim to give a complete presentation of recent and past traffic data to determine traffic behavior.

Main Specifications

There are three main factors that influence the probability of a driver encountering traffic along a certain route, and consequently will be incorporated into our application:

1. **Historic Traffic Patterns**

   As previously mentioned, a route which historically experiences heavy congestion may exhibit patterns of similar traffic activity. Our service will take this into consideration, past accident reports on a route which should increase the probability of traffic within its respective incident locations. These reports will be retrieved from traffic information sites freely available on the internet.

2. **Inclement Weather Conditions**

   During periods of inclement weather, a driver's behavior changes in response. He or she will likely slow down because of the reduced vision, which inevitably leads to higher congestion. Our service will look mainly into three weather effects: fog, heavy rain, and snow, all of which increase the probability of traffic in affected areas. We can easily access and retrieve relevant weather data from online weather services.

3. **Time of Day**

   The time of day will heavily affect how many cars will likely be on the road. We will adjust the levels of predicted traffic congestion depending on the time of day, which becomes especially relevant during busy hours, like rush hour in the mornings and evenings.

Utilizing all these factors together, we can then obtain a clear representation of how traffic should act in a given area.

Other Features

After the core functionality of displaying traffic info based on past traffic, weather, and time of day is fully implemented, we will look at several more desirable features, some of which are commonplace competitor products, but others which are unique to ours. Our overall goal of the finished product is to integrate many different functionalities of already existing services and combine them into one highly useful application. Here are some of the potential additions:

- Users of our service may report areas where there have been police monitoring and activity. This information will then be passed to other users, notifying them so they can either avoid highlighted areas or slow down to prevent getting ticketed by the authorities. This will promote wary and safe driving. Features will include an audio notification if the user is speeding in an area where police are known to frequently ticket.

- A sleek GUI to truly provide a user-friendly experience. The interface should be intuitive to the user as we are providing a service. Also, new editions will present the user with the fastest alternative route when dealing with traffic.

- Options for social media integration with popular apps such as Facebook and Instagram. Users can “check in” on Facebook and receive recommendations from friends for restaurants. If there is a popular restaurant being highly mentioned on Yelp, Twitter, or shared on Instagram users can see pop ups or notifications on the GPS app.

- Voice Activation for improved usability, allowing consumers to not get distracted while driving and thus reduce the amount of casualties from cellular distraction. Personalized for the consumer, giving preventative warnings to allow consumers to prematurely divert from their route. The voice activation can also be used to choose a song from your connected iPod or iPhone using a voice command. We will also give the user the ability to pair the music with different stations such as Pandora, Milk Music, Spotify, Soundcloud, Google Music, and various other mobile applications.

- Personalized music based upon location, weather, traffic, and preference. In order to increase consumer satisfaction, we customize the music in the car along with the user's mood, weather, traffic congestion, and several other factors.

- The addition of scenic route availability, this will lessen the amount of frustration received from "rush hour" traffic. Most consumers respond to visual stimulus, thus we want to increase consumer satisfaction by introducing the ability for users to rate the scenic value of a specific route, eventually giving users the option to choose either the fastest route or the scenic route.

- Monthly updates for newly added roads to increase accuracy of directions, avoid traffic, and reduce travel time.

- An optional "Do Not Disturb" feature that will prevent the notifications of text messages while the app is open. Along with our voice activation, this will help keep drivers engaged and safe on the road while using our application.

Example Outline of User Activities

The desired traffic services will view traffic predictions within the state. First, the user selects his zip-code to view traffic information in his vincinity. This process is outlined below:

"Traffic Summary within Zip Code" [The user is presented additional choices]

1. "Select Target by Zip Code"
    - user enters zip code of target traffic
2. "Time of Day"
    - user selects intervals of one hour

3. “Weather Conditions”
    - user selects which weather was present during their target traffic location
4. “Police Monitoring”
    - user selects police monitoring feature to be on or off along their target traffic location
5. “Voice Activation”
    - user toggles the voice activation feature

The user will be able to look up a traffic summary based on their selection of zip code, the time of day, and weather conditions. These basic selections that users will find intuitive and easy to understand will help provide an accurate match to current traffic conditions when analyzing past data.

Another way of viewing traffic predictions will be along a route. The user will input a starting and destination location. The service will consult with Google Maps to find the fastest route disregarding traffic information and display expected traffic along that route. The service will now suggest routes that avoid traffic congested areas to the user. The process shown below:

“Traffic Summary along a route” [The user is presented a minimal traffic route]

“Starting Address”, “Destination Address”

- The user will input these two boxes then the service will present a route between these two locations via GUI

“Time of Day”

- user selects intervals of one hour

“Police Monitoring”

- user selects police monitoring feature to be on or off along their target traffic location

“Voice Activation”

- user toggles the voice activation feature

Mobile Application

After the web application is fully functional, we are looking to port our product onto cellular devices, namely smartphones. This project will eventually be extended to mobile applications, where the user can access the traffic monitoring application on the road. It should retain all the core functions of the original web application, but still be simplistic and intuitive enough for people who are driving or on the go. This will increase the overall portability of the product, as well as providing users with a way to access traffic data in a situation where he or she most likely finds it the most relevant: while actually driving.

The mobile app should not trim away any of the main functions of the original application. Such features include presenting traffic data "within a zipcode" or "along a route", inclusion of current time, weather and police report data, and voice activation features. Furthermore, all of the original methods in calculating the traffic congestion will remain the same, along with the methods of taking a user's inputted data and display relevant traffic information based on data taken from traffic and weather services. The only changes from the web application will be in the form of the GUI, which will be optimized for a smartphone experience. The map interface will take up more than half of the screen, while the menus will be organized vertically to complement a smartphone.

One major consideration of a smartphone app will be in its ease and safety of use. Because users will most likely be multitasking between driving and navigating the app, it is crucial the app remains simple and intuitive, with the fewest number of key-presses as possible. Also, voice activation may greatly reduce the distractions upon a user, and can potentially mitigate safety risks while driving.

Glossary of Terms

- *Administrator* - An entity which has privileges to customize, alter and update all facets of the traffic monitoring system.
- *Algorithm* - Procedure or formula for solving a problem.
- *Application* – The main program that the user will be interacting with. This program is made to be user friendly and is located on the website.
- *Current Traffic*– Traffic that is currently stored on the Traffic Service websites at the time the user is utilizing the traffic monitoring service. This data is concrete, and describes the known conditions of traffic at the moment.
- *Database* - A storage device that can be used to store the weather data and traffic data being collected.
- *Developer* - Someone who is involved with creating the website's front-end and its back-end
- *Google Maps*- A service that contains maps. The application will display traffic information to the user using this service.
- *Google Traffic* - A service that contains traffic data that will be collected, parsed, and stored in a database.
- *Graphical User Interface (GUI)* - A type of interface that allows users to interact with the website using graphical components (buttons, checkboxes, etc.) and images instead of text-based commands.
- *Input* - Information that is entered by the user into the application in order to specify the output desired.
- *Mobile Application* - Software for an Android mobile device that is available to all users of the traffic system. Mimics the basic function of the web application.
- *Mobile Device* - A device that is meant primarily for mobile use, including tablets and smartphones

- *Module* - Any of a number of distinct but interrelated units from which a program may be built up or into which a complex activity may be analyzed.

- *Parsed* - Analyze (a string or text) into logical syntactic components, typically in order to test conformability to a logical grammar.

- *User* – A person who intends to access historical traffic data by using the traffic

System Requirements

Enumerated Functional Requirements

Identifier	PW	Requirement
REQ1	4	The system interface shall allow the user to specify traffic summary information by zip code, time of day, weather conditions, and road conditions.
REQ2	5	The system shall pull traffic data from inner Google Traffic API.
REQ3	5	The system shall analyze the traffic data.
REQ4	5	The system interface shall display the traffic summary.
REQ5	4	The system's interface shall include text boxes in which the user can input their starting location and destination for directions.
REQ6	5	The system shall pull weather data for every few hours.
REQ7	3	The system shall allow the user to zoom in/out of a map to different areas.
REQ8	2	The system shall allow user non-essential features i.e. voice activation, music, police watch
REQ9	3	The system shall extend services to mobile applications.

Table 1: Enumerated Functional Requirements for a traffic monitoring system

Enumerated Nonfunctional Requirements

Identifier	PW	Requirement
REQ10	3	The system shall have an intuitive and user friendly web page.
REQ11	4	The system shall display Google Map's suggested route as well as the suggested route based on traffic history.
REQ12	2	The system shall have a mobile version of this system.
REQ13	5	The system shall use an algorithm which determines the status of traffic.
REQ14	2	The system shall have preferences for route such as toll roads.
REQ15	4	The system shall allow the administrator to control the frequency and time of data gathering scripts.
REQ16	3	The mobile application shall provide analyzed traffic information depending on the user's current location.

Table 2: Enumerated Nonfunctional Requirements for a traffic monitoring system

FURPS Table

Functionality	❖ Features Web and Mobile based Applications for ease of use for the Users. ❖ Features inputs such as Zip Code, Weather, and Time of Day in order for the User to get an accurate depiction of traffic history they wish to see. ❖ Features a police report system. ❖ Features hands free usage with voice activation. ❖ Features personalized music.
Usability	❖ The home page of the Website Application has an intuitive menu that provides links to all of the other pages on the site. ❖ All of the pages on the site are created with a similar interface, allowing the user to navigate easily. ❖ Input boxes are labeled and easy to see.
Reliability	❖ Evaluations are done to check the validity of the data the Users entered onto the Website Application as well as the Mobile Application. ❖ Many versions and configurations will be implemented in the Application to promise that no information will be lost (Version Control).
Performance	❖ Multiple Users can access the Application at the same time. ❖ Efficiency evaluations are done when designing the maps in order check that unnecessary information is not processed.
Supportability	❖ Website Application is compatible with many browsers. ❖ Mobile Application performs on all updated Android devices. ❖ Classes are documented and separated clearly in order to develop an easy and secure expansion with the Application.

Table 3: FURPS requirements for a traffic monitoring system

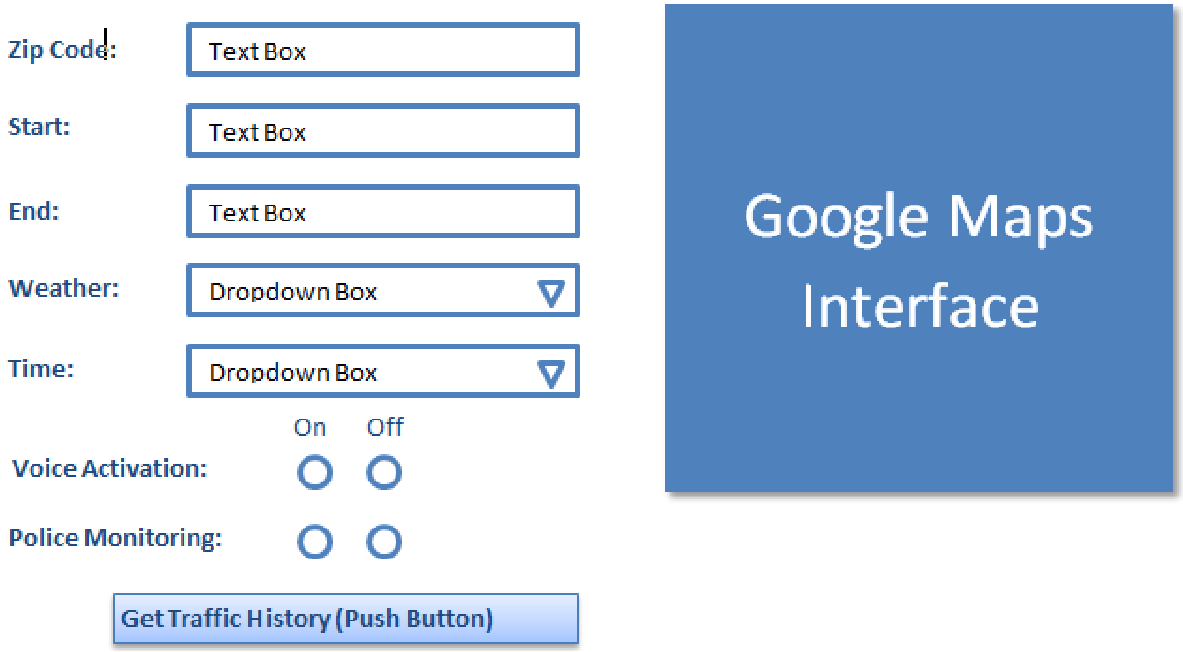
On-Screen Appearance Requirements

Identifier	PW	Requirement
REQ17	5	The system shall contain a Google Maps interface in its display.
REQ18	4	The system shall have a simple and concise mobile app GUI similar to the website version. (Traffic Monitoring System)

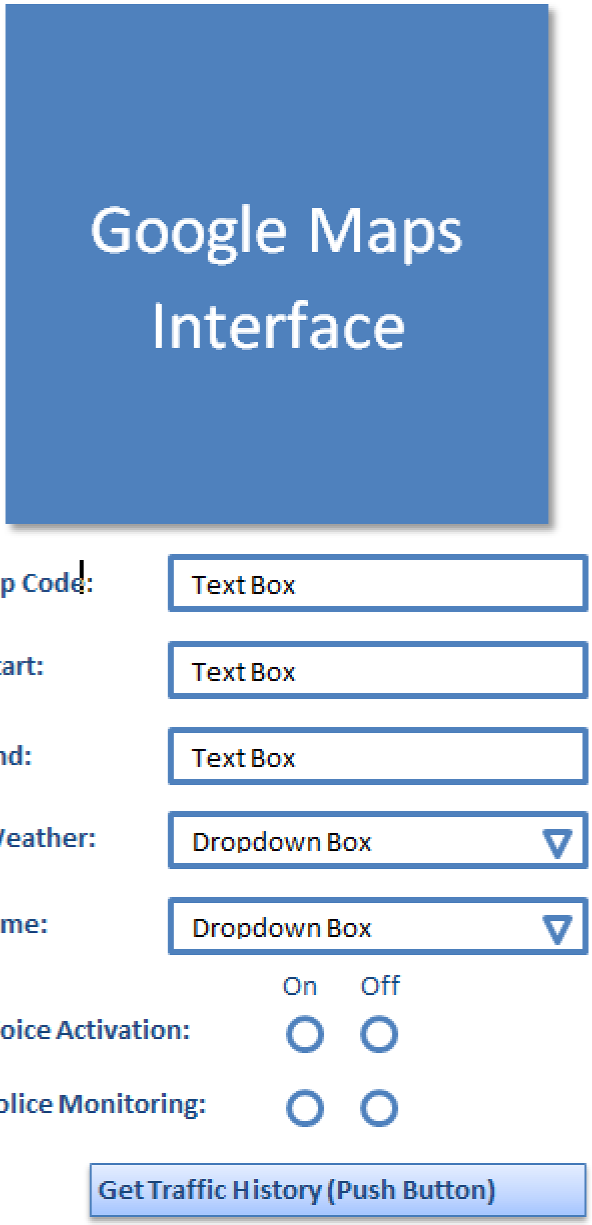
Table 4: On-screen Appearance Requirements for a traffic monitoring system

On-Screen Appearance Illustrations

- ❖ REQ17



❖ REQ18


Google Maps Interface
Zip Code: Text Box
Start: Text Box
End: Text Box
Weather: Dropdown Box ▽
Time: Dropdown Box ▽
On Off
Voice Activation: ○ ○
Police Monitoring: ○ ○
Get Traffic History (Push Button)

Functional Requirements Specification

- Computer Users
- Administrator
- Mobile Users

Actors & Goals

- *User*
  - Initiating Actor
  - The user accesses the main application to receive traffic history reports and directions of their choosing

- *Administrator*
  - Initiating Actor
  - The administrator's goal is to run and maintain the traffic and weather collection services

- *Application*
  - Participating Actor
  - The goal of the Application to provide the user accurate results of their traffic and directions queries in the form of text or images. The Application platform supports many web browsers and will run on a website.

- *Database*
  - Participating Actor
  - The goal of the Database bookkeep all the data to be used. It will store the weather and traffic data sent by the Weather and Traffic Services. Police report data will be received through forms on our website. It will also store the parsed traffic data that is used to display the traffic history. The database is scalable.

- *Mapping Service*
  - Participating Actor
  - The goal of the Mapping Service is to plot traffic history data on a map that will be intuitive for the user to understand.

- *Geocoding Service*
  - Participating Actor
  - The goal of the Geocoding Service is to obtain the longitude and latitude coordinates of the address.

- *Directions Service*
  - Participating Actor
  - The goal of the Directions Service is to get a route from starting to end destination locations. A further goal is to implement the route based on traffic history and real time traffic.

- *Police Report Service*
  - Participating Actor
  - The goal of the Police Report Service is to store user submitted reports into the database. It parses the reports by region and time.

- *Weather Service*
  - Participating Actor
  - The goal of the Weather Service is to provide weather data relating to the roads that the Application finds in the route. This data will be stored in the Database.

- *Traffic Service*
  - Participating Actor
  - The goal of the Traffic Service is to provide traffic data relating to the roads used that the Application finds in the route. This data will be stored in the Database.

- *Social Media/Music Service*
  - Participating Actor
  - The goal of the Social Media/Music Service is to provide social media and music options while using the Application. The goal is to improve user experience.

Use Cases

- Casual Description

User Case	Name	Description
UC1	TrafficAccount	The web interface will ask the user for the zipcode, time, and weather. After the user inputs the data for a specific zipcode, it will display which roads have a history of traffic and how congested they are.
UC2	GetDirections	The web interface will ask the user for the time, Zipcode, and weather. The User will then input their starting location and destination. The interface will display on the map the suggested route based on routing information from Mapquest and traffic history in the route area.
UC3	MapService	The User's request will be displayed using the Mapping Service. The Mapping Service will be using the Google Maps API to provide the bulk of its use. This service will also display real time accidents and areas of poor road conditions. The map provided will ideally be a non-static map which the Users will have the ability to change to fit their needs.
UC4	GetTrafficInfo	Provide traffic data requested by the Traffic Service. The Traffic Service and its specifications are controlled by the Administrator. This service includes police whereabouts and a scenic route availability.
UC5	GetWeatherInfo	Provide weather data requested by the Weather Service. The Weather Service and its specifications are controlled by the Administrator.
UC6	GetMobileInfo	

		Provides the user with a traffic report in a given location via the mobile application on an Android smartphone
UC7	SocialMedia&Music	Provides the user to access relevant social media recommendations and music applications via the Social Media/Music Service. The Social Media/Music Service and its specifications are controlled by the Administrator.
UC8	VoiceActivation	Provides the user to allow voice activation while using the application, as well as providing the optional “Do Not Disturb” feature.
UC9	ReportTraffic	Allows the user to report the traffic in their area via the mobile application on an Android smartphone. The mobile reports will be added to the database.
UC10	ReportPoliceActivity	Allows the user to report police activity in the area via the mobile application on an Android smartphone. The mobile reports will be added to the database.

Table 5: Casual Descriptions of use cases for a traffic monitoring system

♦ Use Case Diagram

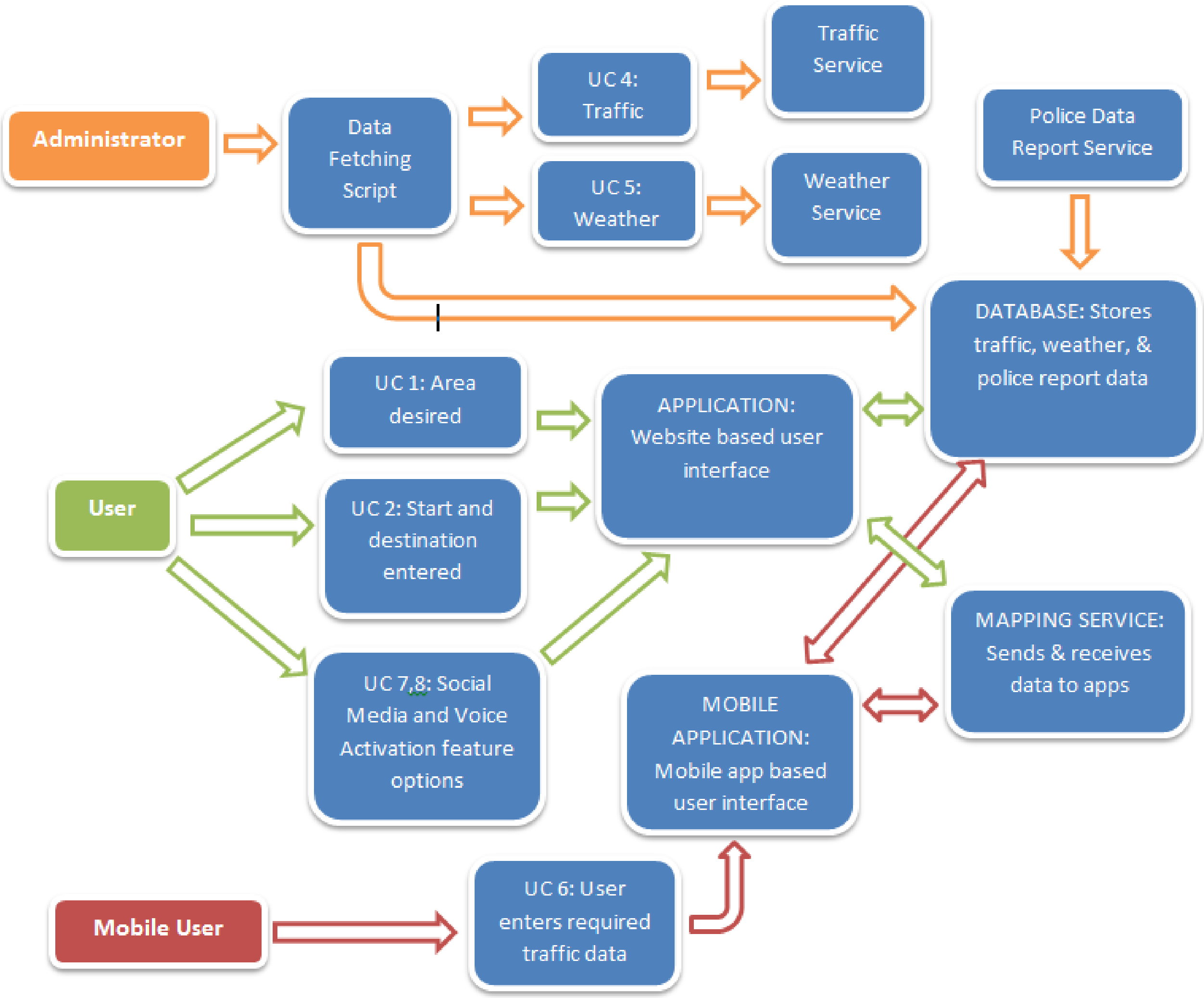

Administrator
Data Fetching Script
UC 4: Traffic
UC 5: Weather
Traffic Service
Weather Service
Police Data Report Service
DATABASE: Stores traffic, weather, & police report data
User
UC 1: Area desired
UC 2: Start and destination entered
UC 7,8: Social Media and Voice Activation feature options
APPLICATION: Website based user interface
MAPPING SERVICE: Sends & receives data to apps
MOBILE APPLICATION: Mobile app based user interface
Mobile User
UC 6: User enters required traffic data

Figure 1: Use Case Diagram for a traffic monitoring system

Traceability Matrix

	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 8
REQ 1	X	X							
REQ 2	X		X	X					
REQ 3	X	X	X	X					
REQ 4			X	X					X
REQ 5		X							
REQ 6					X				
REQ 7				X					
REQ 8							X	X	
REQ 9						X			X
REQ 10	X	X							
REQ 11	X								
REQ 12						X			X
REQ 13	X	X	X	X					X
REQ 14		X							
REQ 15				X	X		X		
REQ 16									X
REQ 17	X	X							
REQ 18	X	X	X	X	X	X	X	X	X

TOTAL PW	**35**	**34**	**24**	**31**	**13**	**9**	**10**	**6**	**22**

Table 6: Traceability Matrix of use cases for a traffic monitoring system

- Fully-Dressed Description

UC1: Traffic Account

Initiating actor: User

Goal: To view the traffic history based upon the user's current zipcode, time, and weather.

Participating Actors: Application, Database, MappingService, WeatherService

Preconditions:

- Application is available
- Database is not empty
- Services needed are available – time and weather

Postconditions:

- Traffic history is displayed on the map

Main Success Scenario:

1) The User inputs a zipcode and time of day. User must select weather conditions from a drop down box.

2) The Application will send a query to the Database based on the zipode, weather, and time of day.

3) The Database will return all the information it has that matches with those specifications.

4) The Application will create a new array that will give the best indication of the traffic history that meets those specifications.

5) The Mapping Service is called and it returns the map image to be displayed.

6) The Application displays the map image for the user to view.

Extensions:

2) Invalid zipcode

I. Application will detect incorrect zipcode.

II. Application will suggest similar zipcodes.

III. Application will request user to re-enter zipcode.

5) An empty array is returned

I. Application will return only the zipcode.

II. Mapping service will be called and only history based on the zipcode and any other available specifications will be requested.

III. The image will be displayed along with a message indicating the missing variables.

UC2: Get Directions

Initiating actor: User

Goal: To obtain driving directions based on traffic conditions along with an image of the route on a map.

Participating Actors: Application, Database, MappingService, Geocoding Service, and DirectionsService

Preconditions:

- Application is available
- Database is not empty

- All services are available

Postconditions:

- Directions and route images are displayed for the user on the Application.

Main Success Scenario:

1) The User inputs a zipcode, a time of day, and weather.

2) The User clicks the “Submit” button on the Application.

3) The Application takes the inputted addresses and calls the Geocoding Service.

4) Geocoding Service returns the longitude and latitude of the addresses.

5) The Application uses the longitude and latitude to create a call to the Database to find all the traffic points between the starting and end location.

6) The Database returns all the traffic points and the severity.

7) The Application takes the traffic points and creates the call to the Directions Service.

8) The Directions Service returns a suggested route, which has been optimized with the traffic severity points.

9) The Application calls the Mapping Service with the route information.

10) The Mapping Service returns an image of the map.

11) The Applications displays the map of the route for the User along with the directions.

Extensions:

4) The user input an invalid or not found address.

I. The Application will stop processing and display on the page that the address was invalid or not found.

7) The Application received no traffic data from the database.

I. The Application continues processing. The route will not be influenced by the traffic data and will be a simple quickest route.

UC4: Get Traffic Info

Initiating actor: Application

Goal: To obtain traffic data.

Participating Actors: Database, Traffic Service

Preconditions:

- Database is available
- Traffic Service is available
- Thirty minutes passed since last call

Postconditions:

- Traffic data stores into the database
- Begin countdown to next call

Main Success Scenario:

1) Application seeks to obtain the traffic data.

2) The traffic is obtained using a traffic service and the data is reviewed.

3) A script parses the data to be used in the database.

UC5: Get Weather Info

Initiating actor: Application

Goal: To obtain weather data.

Participating Actors: Database, Weather Service

Preconditions:

- Database is available
- Weather Service is available
- Thirty minutes passed since last call

Postconditions:

- Weather data stores into the database
- Begin countdown to next call

Main Success Scenario:

1) The weather is obtained using a weather service and the data is reviewed.

2) A script parses the data to be used in the database.

Extensions:

1) The primary weather service is unavailable.

I. A secondary service will be used to obtain weather data.

UC6: Get Mobile Info

Initiating actor: User on smartphone

Goal: To obtain a traffic report on a mobile device.

Participating Actors: Mobile Application, Database, Mapping Service

Preconditions:

- Network communication available

Postconditions:

- Traffic report on mobile device

Main Success Scenario:

1) User enters destination.

2) Application collects location and time from phone.
3) Application sends data to web based system.
4) Web based system returns report to application.
5) Application displays the results.

Extensions:

1) The user enters an invalid destination.

I. The user is notified and asked to try again.

UC9: Report Traffic

Initiating actor: User on smartphone

Goal: To allow users to share and update traffic conditions using the smartphone app.

Participating Actors: Mobile Application, Database, Mapping Service

Preconditions:

- Network communication and GPS available

Postconditions:

- Give user redeemable points for personal updates

Main Success Scenario:

1) User selects traffic intensity of their completed route from a variety of choices.
2) Report is sent to the database.
3) User receives points that can be redeemed later.

Extensions:

2) The database fails to collect the report.

I. The user is notified and asked to try again.

System Sequence Diagrams

❖ *Use Case 1*

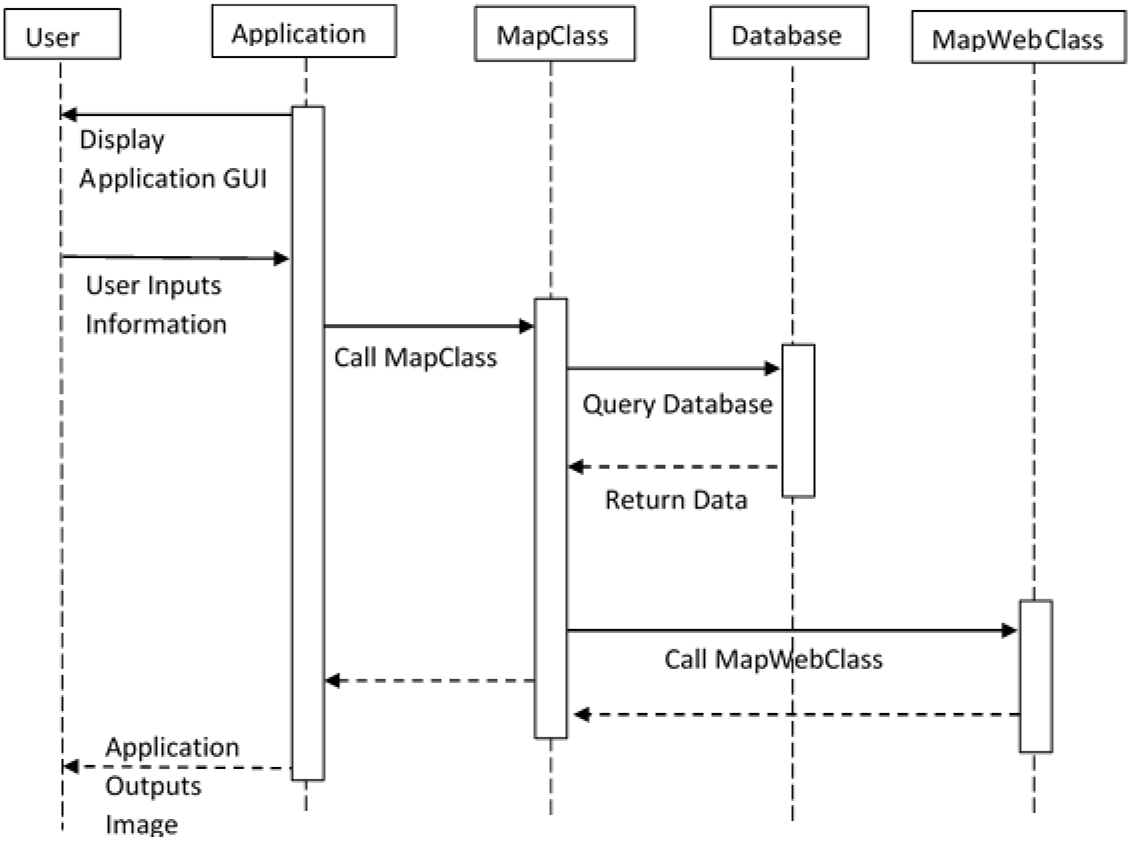


Figure 2: System sequence diagram for Use Case 1

Description:

First, the application presents the user with a GUI that allows him to enter in relevant information, including zipcode, weather conditions, and time of day. User then inputs said information. The application calls the class “MapClass”, which then queries the database for the data pertaining to the user’s inputs of selected weather, zipcode, and time of day. Database returns the information, which is then parsed by MapClass. Next, MapClass calls class “MapWebClass”, returning a map image, able to be displayed on the GUI. MapClass sends this image to the app, which then displays the image on the screen for the user.

❖ *Use Case 2*

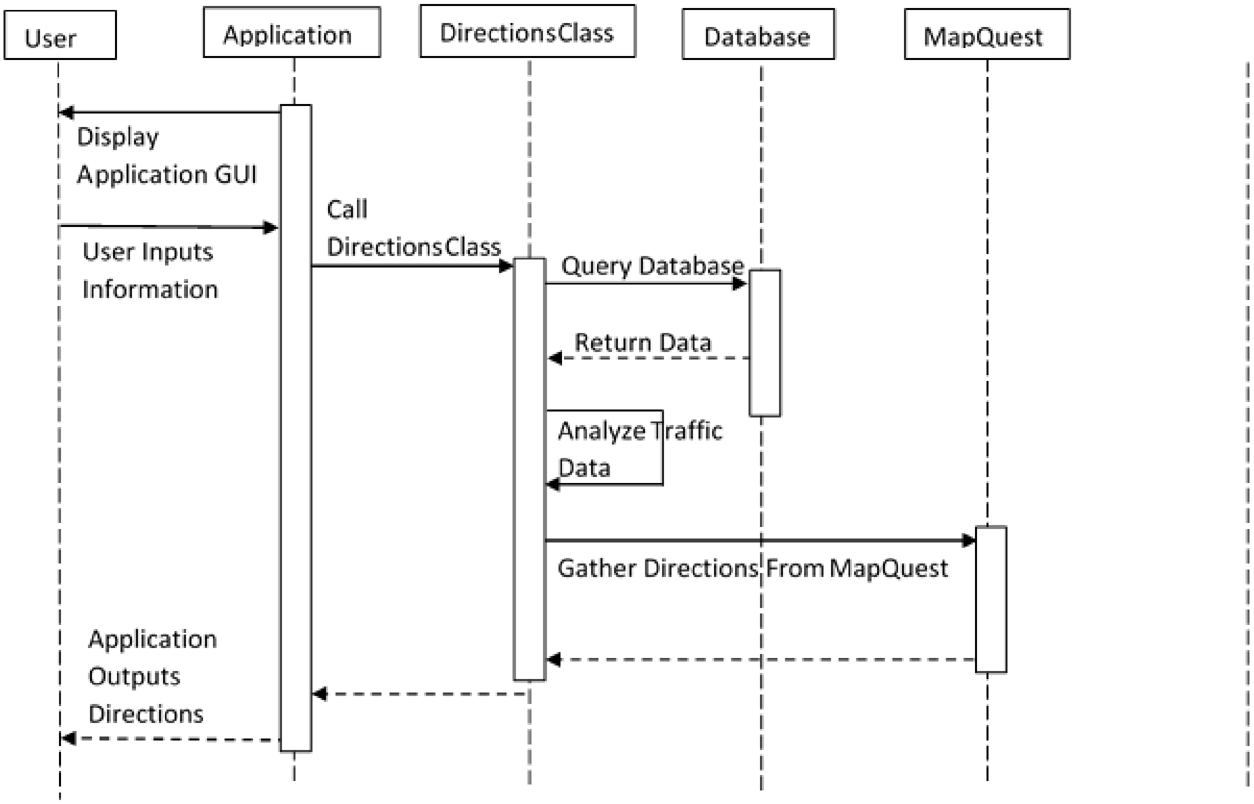


Figure 3: System sequence diagram for Use Case 2

Description:

Application starts by displaying the GUI to the user, who then inputs information relating to his start and end destinations to the system. It then calls class “DirectionsClass”, which calls the database with the proper start and end locations. The database returns all the relevant data regarding weather conditions, traffic, etc. It is the analyzed to find an optimal route. Then, DirectionsClass gathers directions from MapQuest, which then sends the directions back to itself. Finally, the directions are routed back to the user shown on the screen.

- *Use Case 4*

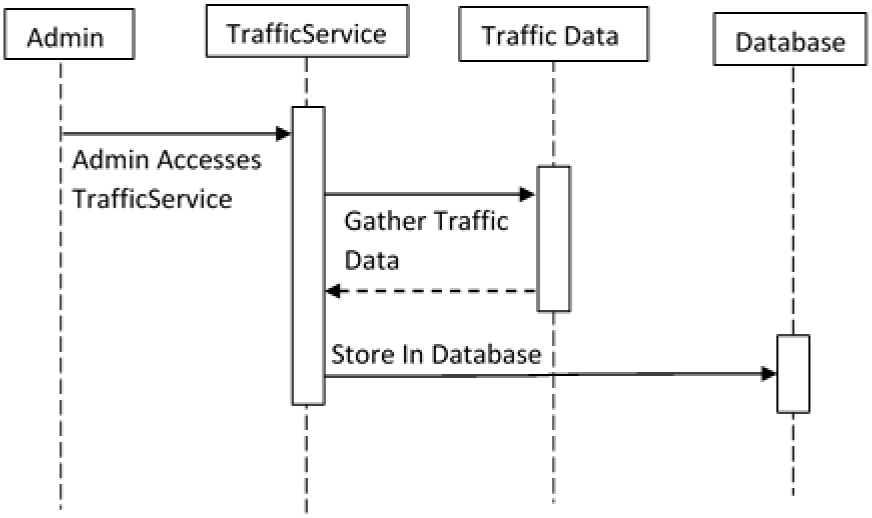


Figure 4: System sequence diagram for Use Case 4

Description:

The admin either accesses the TrafficService or sets it to be updated periodically. TrafficService class then gathers traffic data from the web, which is sent back to the class. The class then stores the relevant traffic data into the database. This use case can either be done manually by the admin or programmed to be updated every X amount of time.

❖ *Use Case 5*

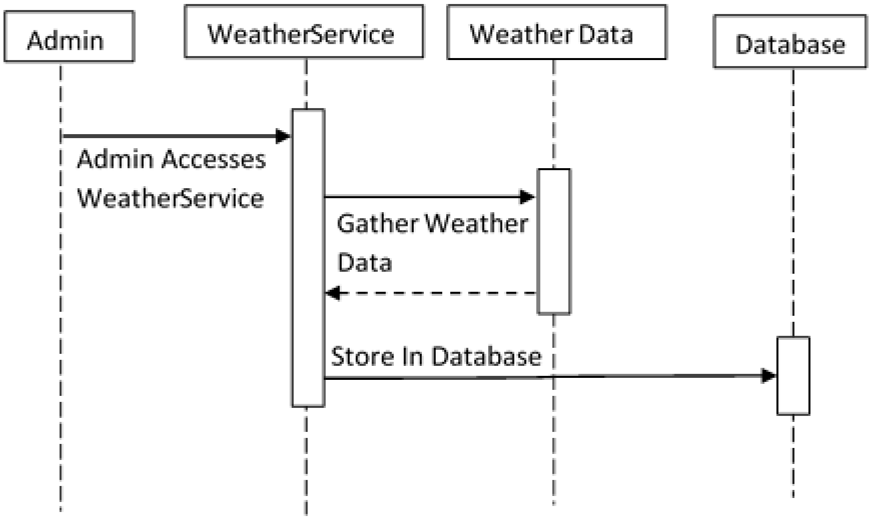


Figure 5: System sequence diagram for Use Case 5

Description:

This is very similar to the use case 4, except instead of traffic data, the Admin uses the WeatherService to gather weather data. Similarly, this process can either be performed manually or automated.

❖ *Use Case 6*

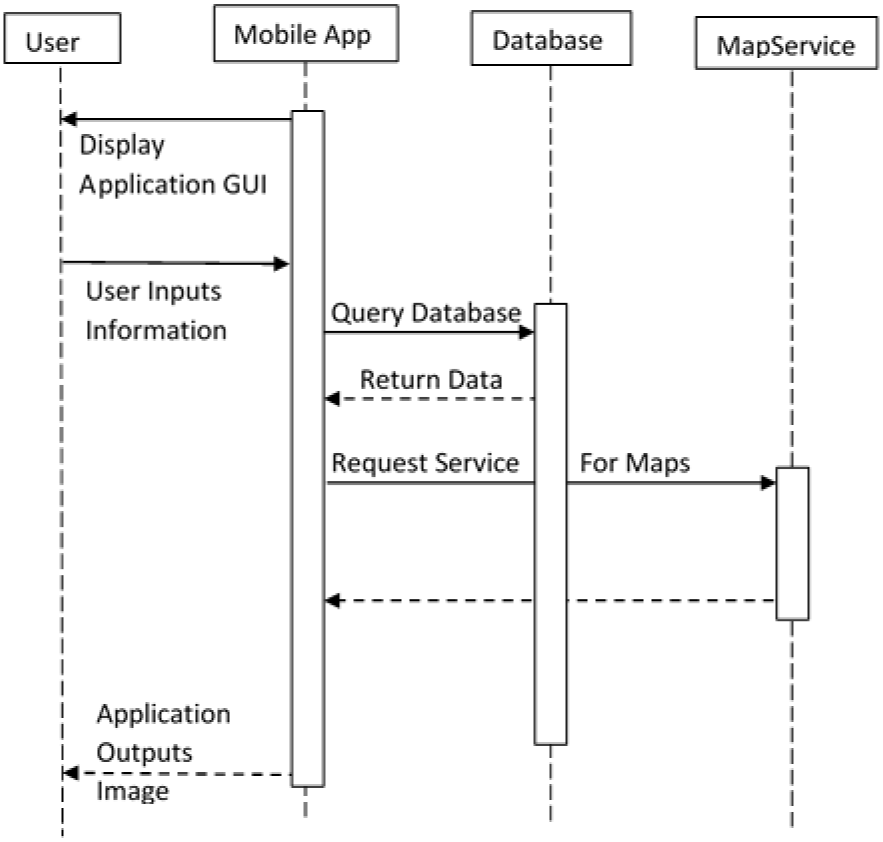


Figure 6: System sequence diagram for Use Case 6

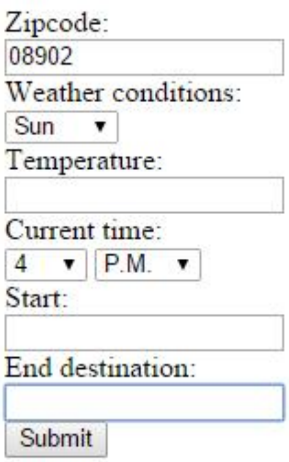
Description:

The mobile app presents a GUI to the user's smartphone screen. The user then inputs relevant information regarding his weather, zipcode, and time of day to the system. The mobile app queries the database, seeking the relevant information and returns it. Then, the app requests an image of a relevant map from MapService class, which sends it back to the app. Finally, the app presents the map image to the user through the screen.

User Interface Specification

Preliminary Design

- Website Interface Specifications



Zipcode:
08902
Weather conditions:
Sun ▾
Temperature:

Current time:
4 ▾ P.M. ▾
Start:

End destination:

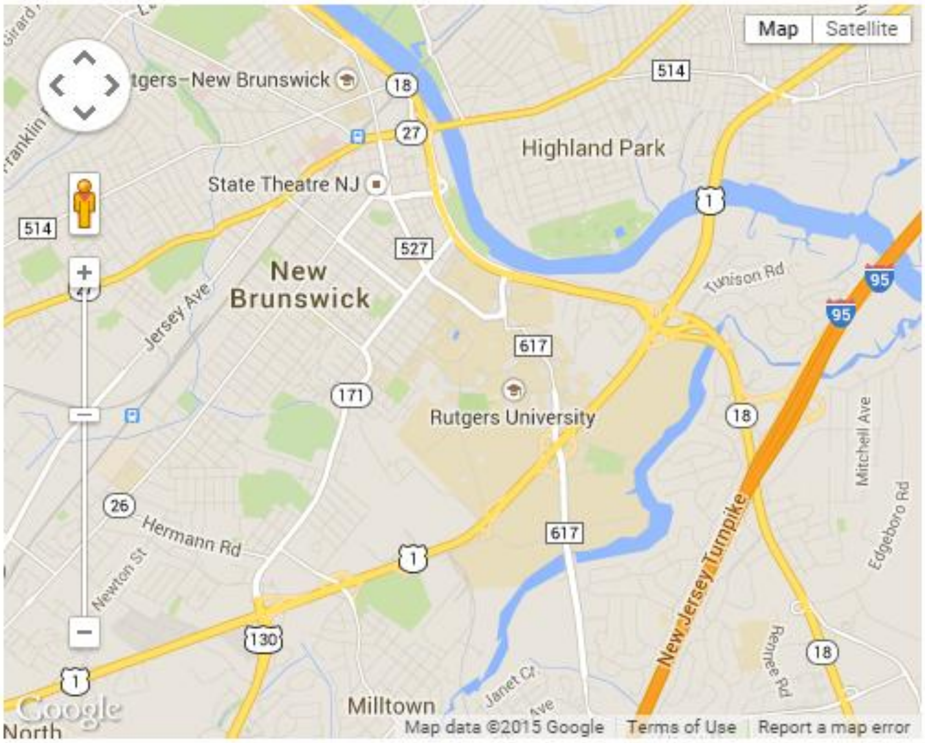
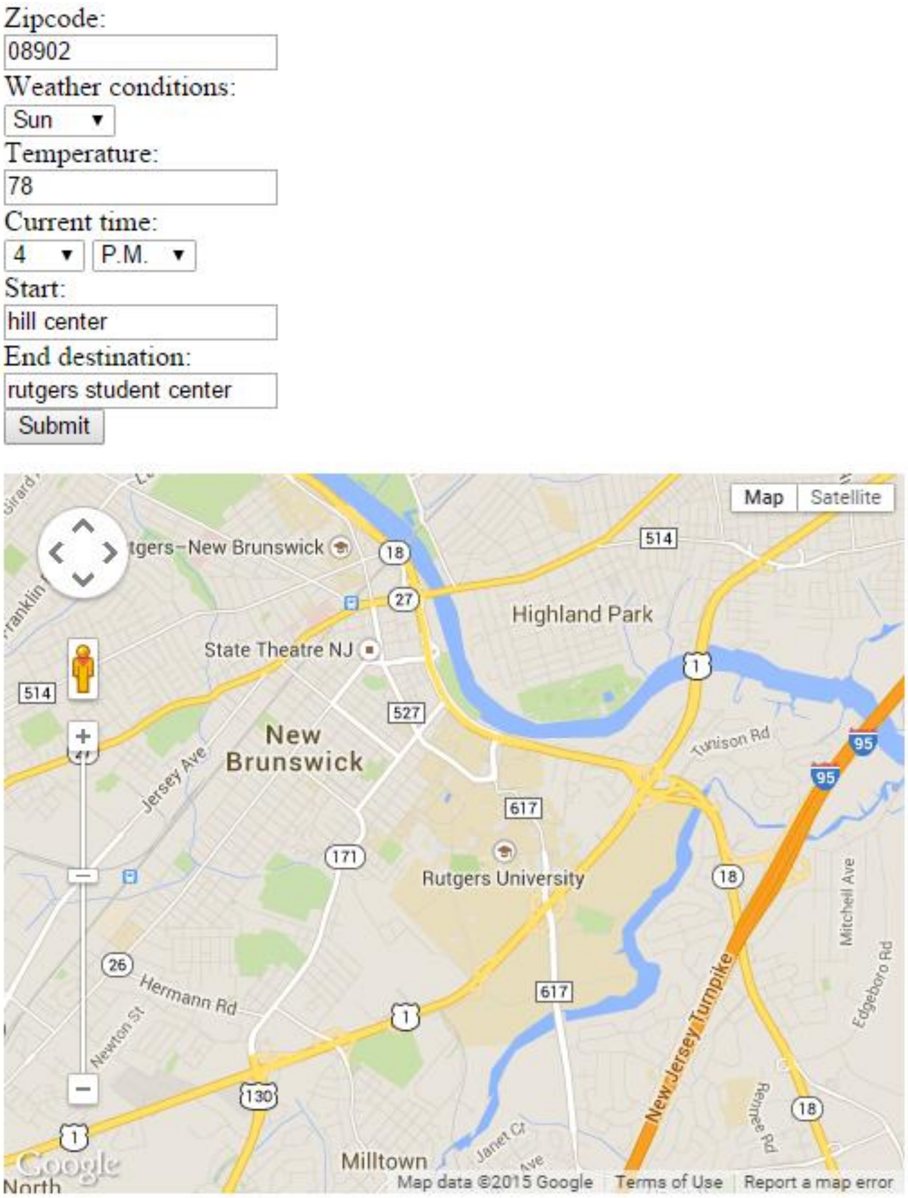


Figure 7: User Interface Specifications for Website



Zipcode:
08902
Weather conditions:
Sun ▼
Temperature:
78
Current time:
4 ▼ P.M. ▼
Start:
hill center
End destination:
rutgers student center
Submit

Figure 8: User Interface Specifications for Website Example

Description:

Once the user enters all their information, they will click submit. The map will update to show the best directions possible, along with a set of instructions as to how to get there.

- ❖ Android Interface Specifications

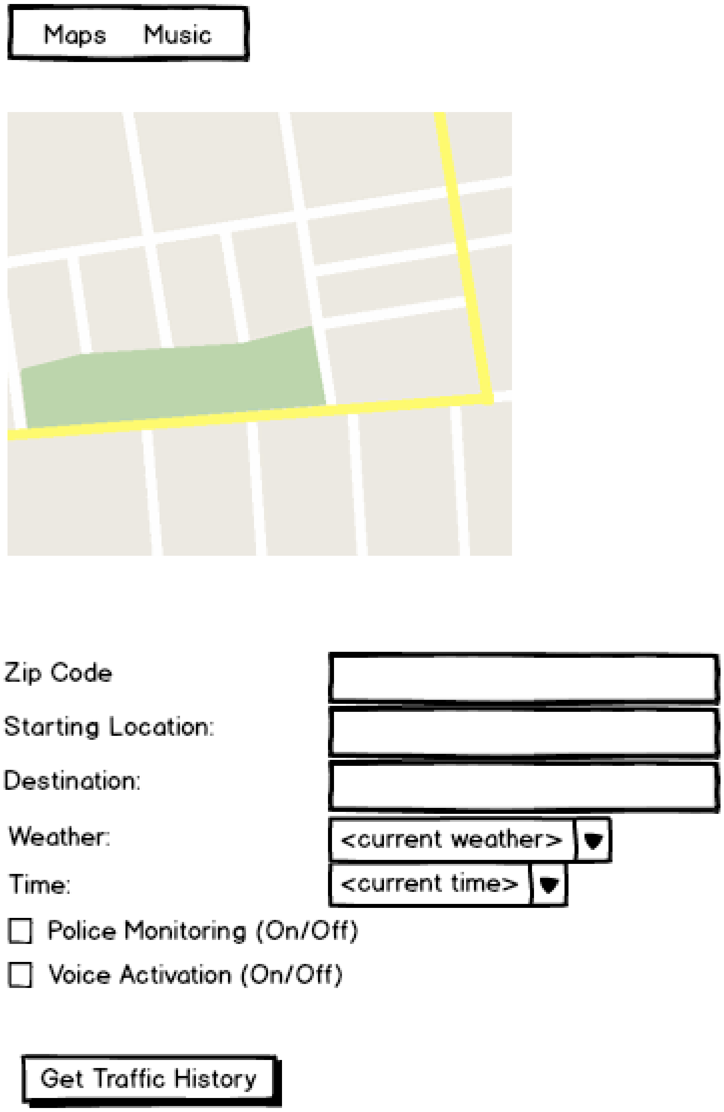


Figure 9: User Interface Specifications for Mobile Application

Description:

Once the user enters all their information, they will press “Get Traffic History”. The map will update to show the best directions for them to take, along with a set of instructions for how to get there.

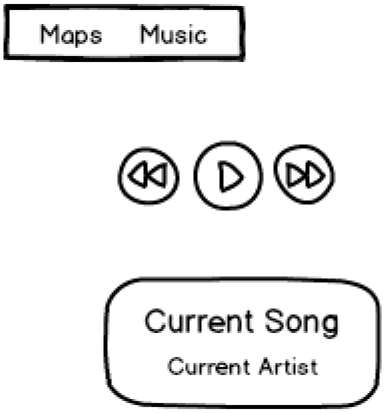


Figure 9: User Interface Specifications with Music Feature for Mobile Application

Description:

When the user navigates to the Music tab, there will be a playlist of songs based on the traffic search, and the user will be able to pause/play the song, and navigate between songs. The song information will be displayed below these controls.

User Effort Estimation

- UCP: Use Case Point
- UUCP: Unadjusted Use Case Points
- TCF: Technical Complexity Factor
- ECF: Environment Complexity Factor

UCP = UUCP x TCF x ECF

Actor Type	Description of how to recognize the actor type	Weight
Simple	The actor is another system which interacts with our system through a defined application programming interface (API)	1
Average	The actor is a person interacting through a text-based user interface, or another system interacting through a protocol, such as a network communication protocol.	2
Complex	The actor is a person interacting via a graphical user interface.	3

Table 7: Definition and weight of different actor types

- UUCP = unadjusted actor weight (UAW) + Unadjusted use case weight (UUCW)

Finding UAW:

Actor Name	Description of relevant characteristics	Complexity	Weight
Administrator	Administrator interacts with system through command lines and/or interface.	Complex	3
User	User interacts with system through graphical user interface.	Complex	3

Mobile User	Mobile user interacts with system through graphical user interface.	Complex	3
Application	The Application is a system interacting with our system.	Average	2
Database	The Database is a system interacting with our system.	Simple	1
Mapping Service	The Mapping Service is a system that our system interacts with.	Simple	1
Directions Service	The Directions Service is a system that our system interacts with.	Simple	1
Police Report Service	The Police Report Service is a system that our system interacts with.	Simple	1
Weather Service	The Weather Service is a system that our system interacts with.	Simple	1
Traffic Service	The Traffic Service is a system that our system interacts with.	Simple	1
Social Media/Music Service	The Social Media/Music System is a system that our system interacts with.	Simple	1

Table 8: Unadjusted Actor Weight for a traffic monitoring system

- UAW = (5 x Simple) + (1 x Average) + (3 x Complex) = 5x1 + 1x2 + 3x3 = 16

Finding UUCW:

Actor Type	Description of how to recognize the actor type	Weight
Simple	Simple user interface. Up to one participating actor (plus initiating actor). Number of steps for the success scenario: <= 3. If presently available, its domain model includes <=3 concepts.	5
Average	Moderate interface design. Two or more participating actors. Number of steps for the success scenario: 4 to 7/ If presently available, its domain model includes between 5 and 10 concepts.	10

Complex	Complex user interface or processing. Three or more participating actors. Number of steps for the success scenario: >= 7. If available, its domain model includes >= 10 concepts.	15

Table 9: Definition and weight of different actor types for unadjusted use case weight

Use Case	Description	Category	Weight
Traffic Account (UC 1)	Complex user interface. Three participating actors. 7 steps for success scenario.	Complex	15
Get Directions (UC 2)	Complex user interface. Four participating actors. 12 steps for success scenario.	Complex	15
Get Traffic Info (UC 4)	Simple user interface. Two participating actors. 3 steps for success scenario.	Simple	5
Get Weather Info (UC 5)	Simple user interface. Two participating actors. 3 steps for success scenario.	Simple	5
Get Mobile Info (UC 6)	Moderate interface design. Three participating actors. 5 steps for success scenario.	Average	10
Social Media/Music (UC 7)	Simple user interface. Two participating actors. 3 steps for success scenario.	Simple	5
Report Traffic (UC 9)	Moderate interface design. Three participating actors. 5 steps for success scenario.	Average	10

Table 10: Unadjusted Use Case Weight for a traffic monitoring system

❖ UUCW = (3 x Simple) + (2 x Average) + (2 x Complex) = 3x5 + 2x10 + 2x15 = 65

❖ **UUCP = UAW + UUCW = 16 + 65 = 81**

Finding TCF:

Technical Factor	Description	Weight
T1	Distributed system (running on multiple machines)	2
T2	Performance objectives (are response time and throughput performance critical?)	1
T3	End-user efficiency	1
T4	Complex internal processing	1
T5	Reusable design or code	1
T6	Easy to install (are automated conversion and installation included in the system?)	0.5
T7	Easy to use (including operations such as backup, startup, and recovery)	0.5
T8	Portable	2
T9	Easy to change (to add new features or modify existing ones)	1
T10	Concurrent use (by multiple users)	1
T11	Special security features	1

T12	Provides direct access for third parties (the system will be used from multiple sites in different organizations)	1
T13	Special user training facilities are required	1

Table 11: Technical Complexity Factors weights

Technical Factor	Description	Weight	Perceived complexity	Calculated factor
T1	Distributed system	2	2	4
T2	Expected fast loading pages & good performance on scripts	1	4	4
T3	End-user efficiency has no expectations/is not predictable	1	3	3
T4	Complex internal processing	1	4	4
T5	Reusable web pages and database tables	1	3	3
T6	Installation not required	0.5	1	0.5
T7	Easy to use	0.5	5	2.5
T8	Portability is not a concern	2	1	2
T9	Easy to change by developers	1	3	3
T10	Concurrent use available	1	2	2
T11	Security with user passwords	1	2	2

T12	No direct access for third parties	1	0	0
T13	No special user training required	1	0	0

Table 12: Technical Complexity Factors for a traffic monitoring system

- TCF Total: 30
- C1 = 0.6
- C2 = 0.01
- **TCF = 0.6 + (0.01 x 30) = 0.9**

Finding ECF:

Environmental Factor	Description	Weight
E1	Familiar with the development process	1.5
E2	Application problem experience	0.5
E3	Paradigm experience	1
E4	Lead analyst capability	0.5
E5	Motivation	1
E6	Stable requirements	2
E7	Part-time staff	-1
E8	Difficult programming language	-1

Table 13: Environmental Complexity Factors weights

Environmental Factor	Description	Weight	Perceived Impact	Calculated Factor
E1	Beginner familiarity with the UML-based development	1.5	1	1.5
E2	Some familiarity with application problem	0.5	2	1
E3	Object-oriented knowledge needed	1	2	2
E4	Average lead analyst	0.5	1	0.5
E5	Highly motivated, some members occasionally slacking	1	4	4
E6	Stable requirements expected	2	3	6
E7	No part-time staff involved	-1	0	0
E8	Programming language is of average difficulty	-1	3	-3

Table 14: Environmental Complexity Factors for a traffic monitoring system

- ECF Total = 12
- C1 = 1.4
- C2 = -0.03
- **ECF = 1.4 + (-0.03x12) = 1.04**

- **UCP = UUCP x TCF x ECF = 81 x 0.9 x 1.04 = 75.82 = 76 use case points**

Domain Analysis

Domain Model

- Application Users:

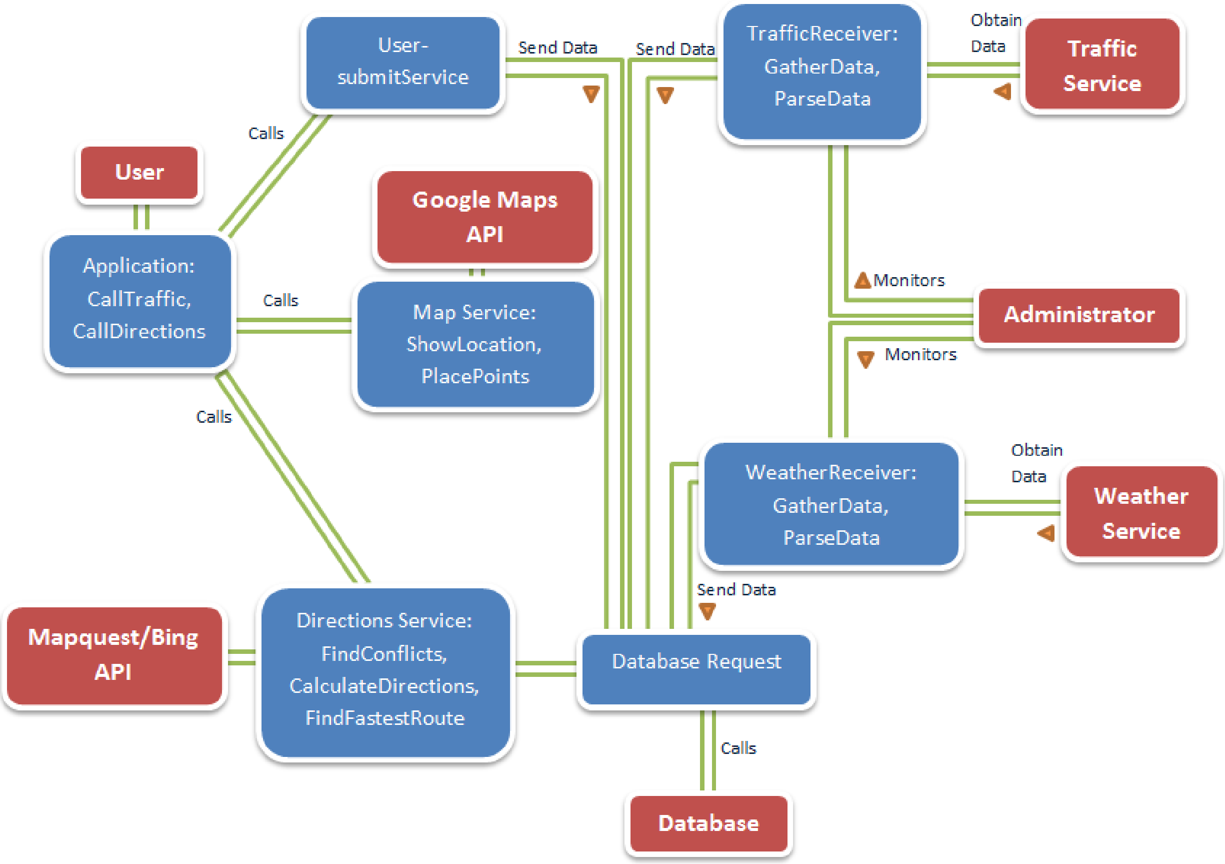

User-submitService
Send Data
Send Data
TrafficReceiver: GatherData, ParseData
Obtain Data
Traffic Service
Calls
User
Google Maps API
Application: CallTraffic, CallDirections
Calls
Map Service: ShowLocation, PlacePoints
Monitors
Administrator
Monitors
Calls
WeatherReceiver: GatherData, ParseData
Obtain Data
Weather Service
Send Data
Mapquest/Bing API
Directions Service: FindConflicts, CalculateDirections, FindFastestRoute
Database Request
Calls
Database

Figure 10: Domain Model diagram

Description:

The Domain Analysis shows that the application is accessible to the User. These bounds are defined as so to create easy user experience. The Application can call the Maps or Directions Service. Although these services are similar, the procedures needed to complete each action require different APIs to be used for each function. The Google Maps API is more suited for placing unique points on the map, while the Mapquest/Bing API is preferable for finding routes while avoiding locations that have high traffic severities. These functions call the Database to receive the required information about the traffic points they wish to analyze or display.

The Administrator has control of the two scripts that can receive data from Web Services. The traffic and weather scripts gather and parse the data received from the Web Services, and they store that data in the database to be used by the main Application. The police data script will gather police data from user submitted service and store that data into the database.

♦ Mobile Users:

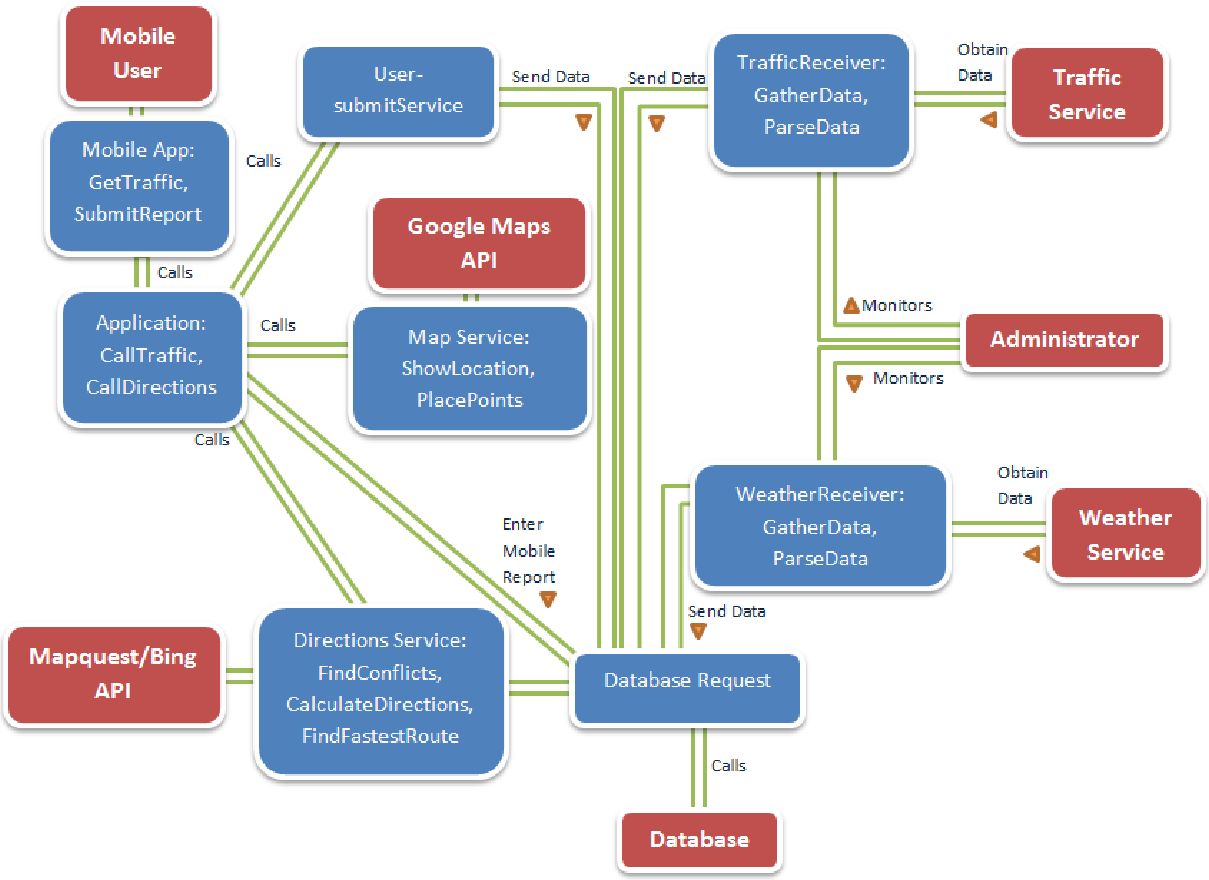

Mobile User
Mobile App: GetTraffic, SubmitReport
Calls
User-submitService
Send Data
Send Data
TrafficReceiver: GatherData, ParseData
Obtain Data
Traffic Service
Calls
Google Maps API
Application: CallTraffic, CallDirections
Calls
Map Service: ShowLocation, PlacePoints
Monitors
Administrator
Monitors
Calls
Obtain Data
WeatherReceiver: GatherData, ParseData
Weather Service
Enter Mobile Report
Send Data
Mapquest/Bing API
Directions Service: FindConflicts, CalculateDirections, FindFastestRoute
Database Request
Calls
Database

Figure 11: Domain Model diagram for Mobile Users

Description:

The Mobile Application is made to mimic the functionalities of the website Application. Because of this, the best way to replicate the functions of the website would be to use the code of the website itself. The Mobile Application gets a series of inputs from the Mobile User, and the Mobile Application sends that data to the main Application. The Application then parses the data from the Mobile Application. At this point, the Application behaves in the exact same way as the previous Domain Diagram. The Mobile Application receives a series of inputs for a mobile traffic report. These inputs are again sent to the main Application. This time, the Application parses the data received, and it enters it straight into the database, bypassing the MapService and DirectionsService.

Traceability Matrix

	Application	Database	Mapping Service	Directions Service	Police Report Service	Weather Service	Traffic Service	Social Media/ MusicService	Mobile Application
UC1		X	X			X	X		
UC2	X	X	X	X		X	X		
UC3	X	X	X			X	X		
UC4		X					X		
UC5						X			
UC6							X		X
UC7								X	X
UC8	X							X	
UC9		X				X	X		X
UC10		x			x				x

Table 15: Traceability Matrix for Domain Analysis

- Definitions:

- Use Case 1: Traffic Account

Concept definitions

Responsibility Description	Type	Concept Name
Takes in user input to find which areas the user is requesting traffic history.	Doing	Application
Database returns the information matching user input and application return output that displays most accurate estimation of traffic history.	Doing	Application
Mapping service is called and displays the map image for user.	Communicating & Doing	MapReturn
Should previously have the traffic, time, and weather that is necessary for user operation.	Knowing	Database

Table 16: Concept Definitions for Use Case 1

Association Definitions

Concept Pair Name	Associated Definition	Association Name
Application ↔ Map Service	The application will send the current user location to the Map Service. The Map service then returns the map area to the user.The Map service returns the map area to the user	Provides Data
MapReturn ↔ Database	The Map Service uses the Database information to display the proper direction, time, weather, traffic congestion, location, etc. through the internal algorithm, and displays it for the user.	Provides Data

Table 17: Association Definitions for Use Case 1

Attribute Definitions

Concept	Attributes	Attribute Description
Application	Time	Present Time desired by user
	Weather	Weather desired by user
	Date	Date and day of week desired by user
	Location	Geographical location desired by user
Database	Traffic Data	Traffic congestion associated with the time, date, and location
	Weather Data	Previous weather history associated with certain areas

Table 18: Attribute Definitions for Use Case 1

❖ Use Case 2: Get Directions

Concept Definitions

Responsibility Description	Type	Concept Name
Receives input from user based on where (zipcode) he would like to receive traffic information.	D	Application
Stores all relevant data from police reports, traffic reports, and weather data.	K	Database
Reads starting and end locations from the user. Estimates the fastest route from start to end, taking into account weather, time of day and historic traffic data.	D	DirectionsService

Table 19: Concept Definitions for Use Case 2

Association Definitions

Concept Pair Name	Associated Definition	Association Name
Application <-> DirectionsService	DirectionsService receives the user's starting and end locations from the Application, and sends back directions and map image of the route.	Provides Data
DirectionsService <-> Database	DirectionsService queries the database and retrieves the relevant information regarding weather, time, traffic, and police data if option was selected.	Provides Data

Table 20: Association Definitions for Use Case 2

Attribute Definitions

Concept	Attributes	Attribute Description
Application	Weather	Type of weather inputted
	Time	Time inputted
	IsPolice	If police monitoring option is turned on
	ZipCode	Zipcode inputted by user
	StartLocation	Starting address location
	EndLocation	Destination address location
Database	Traffic Data	Data of traffic reports based on time of day and location
	Weather Data	Data of weather conditions of current input location
	User Police Data	User submitted police data reports
DirectionsService	Directions	List of directions from address of StartLocation to EndLocation

Table 21: Attribute Definitions for Use Case 2

- Use Case 4: Get Traffic Info

Concept Definitions

Responsibility Description	Type	Concept Name
Obtain the traffic history from traffic service website	D	TrafficIntake
Obtain the weather history from a weather service website	D	WeatherIntake

Table 22: Concept Definitions for Use Case 4

Association Definitions

Concept Pair Name	Associated Definition	Association Name
TrafficIntake → Database	TrafficIntake uses the information gained from TrafficReturn and keeps it in the data base	Provides Data

Table 23: Association Definitions for Use Case 4

Attribute Definitions

Concept	Attributes	Attribute Description
TrafficIntake	Location	Location where traffic is happening
	TrafficType	Congestions level of traffic
	Time	Time of Traffic
Database	TrafficData	Traffic intensity associated with location and time

Table 24: Attribute Definitions for Use Case 4

❖ Use Case 5: Get Weather Info

Concept Definitions

Responsibility Description	Type	Concept Name
Keeps archive of weather history	K	Database
Obtain the weather history from a weather service website	D	WeatherIntake

Table 25: Concept Definitions for Use Case 5

Association Definitions

Concept Pair Name	Associated Definition	Association Name
WeatherIntake → Database	WeatherIntake uses the information gained from WeatherReturn and keeps it in the data base	Provides Data

Table 26: Association Definitions for Use Case 5

Attribute Definitions

Concept	Attributes	Attribute Description
WeatherIntake	Location	Location where weather was recorded
	WeatherType	Type of weather recorded
	Time	Time of weather recorded
Database	WeatherData	Weather associated with location and time

Table 27: Attribute Definitions for Use Case 5

❖ Use Case 6: Get Mobile Info

Concept Definitions

Responsibility Description	Type	Concept Name
Receives input from user and display traffic information based on which areas the user inputs into the mobile application.	D	Mobile Application
Mobile Application sends user inputs to main web application	D	Application
Contains relevant weather, traffic and police report data	K	Database
Mapping service is called and displays the map image for user.	D	MapReturn

Table 28: Concept Definitions for Use Case 6

Association Definitions

Concept Pair Name	Associated Definition	Association Name
Mobile Application → Application	Mobile app has same functionality as main application.	Software Port
MapReturn → Database	The Map Service accesses the database to display the proper direction, time, weather, traffic congestion, location, etc. through the internal algorithm, and displays it for the user on the mobile app.	Provides Data

Table 29: Association Definitions for Use Case 6

Attribute Definitions

Concept	Attributes	Attribute Description
Mobile Application	Weather	Type of weather inputted
	Time	Time inputted
	IsPolice	If police monitoring option is turned on
	ZipCode	Zipcode inputted by user
	StartLocation	Starting address location
	EndLocation	Destination address location
Database	Traffic Data	Data of traffic reports based on time of day and location
	Weather Data	Data of weather conditions of current input location
	User Police Data	User submitted police data reports

Table 30: Attribute Definitions for Use Case 6

- Use Case 9: Report Traffic

Concept Definitions

Responsibility Description	Type	Concept Name
User inputs traffic intensity of the area they are in based upon automatic location services (GPS).	D	Mobile Application
Receives the location and traffic intensity inputted by user.	K	Database
Analyzes this data and starts to create patterns based upon time and location.	D	Database
Database sends this updated information to the application.	D	Application
Gives user points for their contributions.	D	Mobile Application

Table 31: Concept Definitions for Use Case 9

Association Definitions

Concept Pair Name	Associated Definition	Association Name
MobileApplication -> Database	Mobile Application receives user input and sends this data to the database.	Provides data
Database -> Application	After storing and analyzing this data, the application sends this new information to the application.	Provides data

Table 32: Association Definitions for Use Case 9

Attribute Definitions

Concept	Attributes	Attribute Description
MobileApplication	Time	Time taken from mobile device
	TrafficIntensity	User inputted traffic intensity
	Location	Location taken from mobile device
Database	TrafficData	Traffic intensities with their location and times, creating a pattern along the way.

Table 33: Attribute Definitions for Use Case 9

Plan of Work

2/27	• Interaction and Class Diagrams
3/5	• System Architecture and System Design • Data Collection System: Weather & Traffic
3/6	• Advanced template for website • Database Server to keep data (as long as data collection system was successful)
3/9	• Product Brochure for demo • GUI Interface
3/20	• Actual traffic calulating model • Traffic Mapping display
3/22	• Mobile traffic display • Implement weather conditions and road conditions into traffic
3/26	• Police watch, music, and voice activation features • Mobile & Website testing

References

511 Traffic Incidents:
http://www.511nj.org/

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

***"Software Engineering*" by Ivan Marsic:**
http://www.ece.rutgers.edu/~marsic/books/SE/

***"The Costs of Highway Congestion"* by David Morgan**
http://abcnews.go.com/US/story?id=94064

Traffic.com:
https://www.here.com/traffic

Voice activation for android application:
http://developer.android.com/reference/android/speech/RecognizerIntent.html

Weather.com RSS Feed:
http://www.weather.com/