**iTraffic**

**A Traffic Simulator that will deliver dynamic traffic flow through dynamic traffic lights and proper time allocation**

A Project Study Presented to the Faculty of the College of Information,

Computer and Communications Technology

University of San Jose – Recoletos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In Partial Fulfillment of the

Final Requirements for Thesis 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mel Raymond G. Salma

*Proponent*

Mr. John Leeroy Gadiane

*Adviser*

**CHAPTER I**

**INTRODUCTION**

**Rationale of the Study**

Nowadays, the streets of Cebu are witnessing an unprecedented increase in traffic congestion due to road repairs, road accidents, and most especially inadequate traffic light systems. The two factors namely: road repairs and road accidents are events that can’t be controlled or handled completely by traffic personnel, but unlike those two the traffic light system is. The problem with the current traffic light systems of today is that the traffic agent handling the traffic lights at a certain traffic location can’t discern properly the sequence of the timer and light allocation. The reason to it being the lack of vision or understanding towards the ongoing events at the congested road, and thus causing the inefficiency of the traffic personnel.

In one of the major cities of Cebu, named Mandaue City, this city is known for its advanced traffic system with large numbers of traffic cameras at various road intersection points and has a data centre to boot. Mandaue’s main force for monitoring the entire traffic and road situation of the city is the organization called Traffic Enforcement Agency of Mandaue or TEAM. Recently, TEAM has been using an application that gives them the capability to control the traffic lights of any given road intersection found in the Mandaue. Currently, the application has performed suitably with regards to handling the traffic flow, but it doesn’t aid the traffic agent enough to understanding and solving the congested traffic event. Other information about the app is that it works in such; that the traffic agent is capable of pre-setting the traffic timers on a given road intersection and would also be able to manually configure them to whichever would benefit the road. Now at the occurrence of congested traffic and by observation of the progress to being uncongested is becoming stagnant; a traffic road agent would then take into action to contact a traffic agent at the data centre to configure the existing traffic timers and lights in order to lessen the congested traffic. Here, the traffic agent would then be making use of the training the he underwent through. A part of that said training would be able to disseminate how much time to allocate in order to lessen the congestion of traffic and to provide better flow of vehicles.

The current traffic system in Cebu, or anywhere in the Philippines, is mainly composed of a traffic enforcer, traffic lights with a traffic counter, and a data centre. The traffic enforcer is usually seen on the road if either the traffic light/s are malfunctioning or the congested traffic is becoming uncontrollable and becoming lengthy. The traffic enforcer is considered to be the front-face of the traffic management system of the city, and at the back-end of it would be the data centre with its respective agents who are pulling the strings. Here, they will be manipulating the traffic light system, in order to fix or create a solution to the congested traffic that is occurring by using the traffic lights for better traffic flow. Thus the traffic system, so far, comes with certain limitations that indefinitely shows and needs improvement. These limitations being:

1. The traffic system only considers the current situation that is occurring on the spot. It only focuses on what’s in front of the traffic and not take into consideration those who are following.
2. The traffic enforcers can’t balance the traffic altogether or remove the biases that they have discerned through normal vision. Along with the fact that the traffic enforcer and traffic agent have so little vision and are unable to weigh in on the best solution that need to take place, and
3. The issue with regards to the traffic light system being not automated as it only possesses a set amount of time. The rare occasion also to when the current traffic isn’t that heavy, and the traffic light timer remains the same because of it being set at a predefined amount. This rare occasion is sometimes disregarded and left unnoticed by the agents at the data centre.

The goal of this research would be to develop an application that will serve to providing insight or extra insight to the traffic agent on what to do at the presence of a congested traffic situation. That insight would be beneficial, so as to remove any biases that the traffic agent might be afflicted upon. By analysing the entirety of the situation to it becoming congested and along with the corresponding data to lessen or eliminate the congested traffic; it would then prove to be useful for the traffic agent. Now, to what application that could assist the traffic agent in such a way would be the development of a **traffic simulator.** By using a traffic simulator the user would then be able to duplicate a congested traffic event and is free to add multiple inputs like vehicles of different speed and type and also road regulations. During or after a traffic run simulation, the agent would then be presented with analysed data that would be released by the simulator. The simulator would be capable of presenting such data by analysing both the speed and quantity of the vehicles during the simulation. And with the data collected, the traffic agent would process all the data, and thus deliver an unbiased and processed judgment to delivering an adequate or proper allocation or transition of traffic light along with its timer allocation.

**THEORETICAL BACKGROUND**

There has been an ever-growing issue with regard to properly managing congested traffic here in Cebu. Noticeably, one of the factors that have been causing the congestion of traffic would be the inadequate transition of traffic lights. Such an event is caused due to the traffic agent’s lack of information and vision. Lack of information and vision is due to the fact that the road cameras or traffic enforcers are not enough to provide a proper status or outlook of the situation to the traffic agents. This lack then poses an amount of pressure for the traffic agent on how to properly assess the situation. And because of the lack of understanding with what is transpiring at the congested traffic situation, the traffic agent would be forced to creating decisions that would be filled with flaws, ultimately worsening the situation even more.

**Using a Computer Simulator for Traffic Duplication**

There are many variables and principles that a traffic agent must take into account whilst making decisions with how to manage the traffic, most especially the congested ones. These events are known for being unexpected and all too sudden which will, of course, lead to the traffic agent to becoming perplexed with how to resolve the situation. But as the traffic agent remains calm and by then proceeding to follow the proper procedures and protocols then an organized fix to the situation may be developed. There is the possibility of such actions, however, would lie within it the irrationality or even the carelessness of overlooking details that would prove to be highly useful to form the desired output. Thus, in order to subtract the incidence of human error, the traffic agent would be in need of a tool capable of such data analysis and presentation. All which can be made done through the fields of Computer Science.

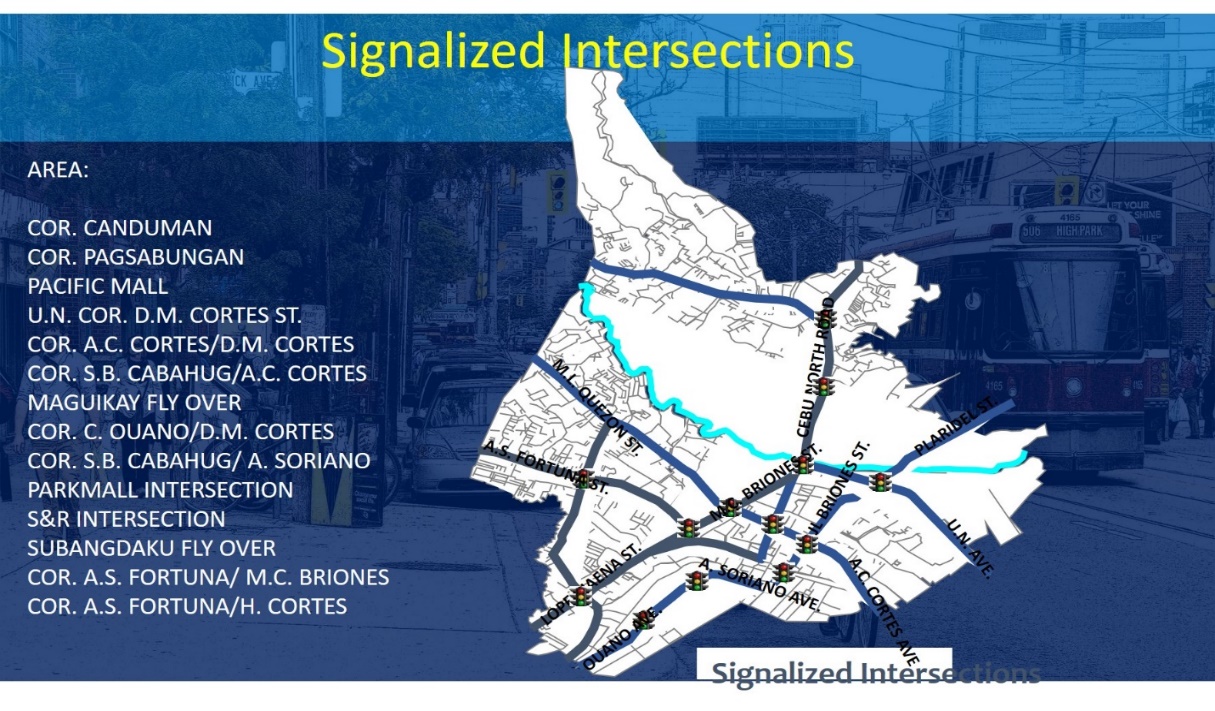
Computer Science is a field with various branches of study that is meant for handling and analysing data, and as such, any problem can be made solvable with the right parameters and the right algorithm to go along with it. Computer Science is all about problems and problem solving, and it is up to the researcher to create an answer to the problem [1]. With Computer Science, the development of a tool that would be able to assist the traffic agents in providing insights on what to do for a certain congested traffic situation would become possible. To create that insight it would require a tool capable of reduplicating that situation so that the agent can witness beforehand the flow of the traffic as how it is currently and how it will be.

This insight tool is what many engineers would call a ***simulator.*** A simulator is a very useful tool that is used to create an imitation or a reconstruction of any certain events in order to create an assessment or analysis [2]. These assessments or analysed data would be then used by data scientists of respective fields to create probable measures to improve their current system flow. Computer simulations, especially, are being used as of late for modelling many natural systems in physics, chemistry, biology, economics and social science. Computer simulations are gradually being used to explore and gain new insights into new technology and to estimate the performance of systems too complex for analytical solutions [3]. This can be then understood that the use of simulation software for handling traffic is a huge possibility. By creating his own flow of events, the traffic agent may use the simulation software to actually reduplicate either the current congested traffic situation or the previous traffic situations from before.

Furthermore, Krajzewicz, Hertkorn, Rössel, and Wagner are known for conducting their study which lead to the development of SUMO (Simulation of Urban MObility). SUMO is a traffic simulation system that is used to predict the traffic and at which location the traffic would then become congested [4]. Their study goes to show that the development of tools for modelling especially for traffic is completely doable. Their developed app is even capable of duplicating the actual layout of the roads of an actual location. However, as the state of the art for traffic simulators, it presented some lacking features that the traffic agent can’t work without. Although SUMO is capable of representing the data of the traffic, it can’t integrate algorithms or machine learning algorithms to which could provide the insights needed for the traffic agent to handle the congested traffic. But nonetheless, the development of SUMO has paved the future for traffic simulators, and as such, it will encourage the future development of more advanced and smarter traffic simulators.

Traffic simulators can be used for various situations may it be at long tunnels or at a small suburban area. Both of which share their own share of congested situations. Traffic congestion may occur at various locations at unprecedented time or events. To such, the development and usage of such simulation software to handle these situations are being done frequently. Traffic simulations are usually tackling events located at the intersection which is where constant events of congestion are usually located. It is such that the researcher is tackling the congested traffic situations of intersections located in Mandaue City. Location of intersections with the highest chances of congestion because of the abundance of the population or renowned record of congestion are being tackled by the research.

Below is an image of the signalized intersections located in Mandaue City. These locations have been recorded and reported to having a frequent history of traffic congestion. This was provided by the TEAM.



*Figure 1. Image of Mandaue City’s Signalized Intersections*

**Traffic Light System**

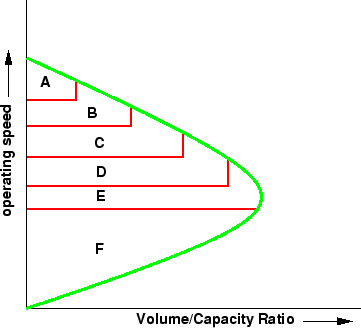
The simulation software that is being developed by the researcher is one that will manipulate the behaviour of the vehicles, traffic timer and traffic lights. These three are the attributes that the researcher will manipulate to creating the output of data that will be used to providing insights for the traffic agent. According to the head of TEAM, the main objective of the traffic light system is for providing safe and orderly movement of traffic. The reason to it is being able to avoid personal danger and property damage. As is, the traffic agent handling the traffic light transition will be the one responsible for creating that stable and ordered traffic. And by doing so, the traffic agent’s decisions will heavily influence the outcome on the current and upcoming traffic.

The study being conducted by the researcher will not only aim to create a traffic simulator, but also a traffic light system. This traffic light system will handle most of the data processing and will produce an output of a flowing decongested traffic. The variables that are to be processed will mostly involve the speed and volume of the vehicles and the delay– refers to nature of movement of the vehicles. Mainly, the traffic light system will be adhering with the traffic rules and light system principles that is stated by the Highway Capacity Manual (HCM, 2010) and the Signal Timing Manual. Additionally, the Signal Timing Manual has been used by the U.S. Department of Transportation, for the sole purpose of providing direction and guidance to managers, supervisor, and practitioners based on sound practice to proactively and comprehensively improve better road stimulation and signal timing.

The researcher will be basing most of the concepts of the traffic light system from the theories present from the *Transportation Engineering: Online Lab Manual*. The online lab manual is a product of investment made by the United States Department of Transportation. This was used to supplement undergraduates, practicing engineers and educational entities with important theories and concepts about what makes traffic and what happens in before and after the beginning of the green signal of the traffic light. It introduces concepts such as computing for the capacity of an intersection or even the lane. It also discusses the queuing theory of the traffic lights, as well with determining the expected duration of a traffic cycle. By knowing the duration of a particular traffic cycle, then a traffic agent would be able to make use of that information to properly handle the transition of vehicles from the traffic using the traffic lights.

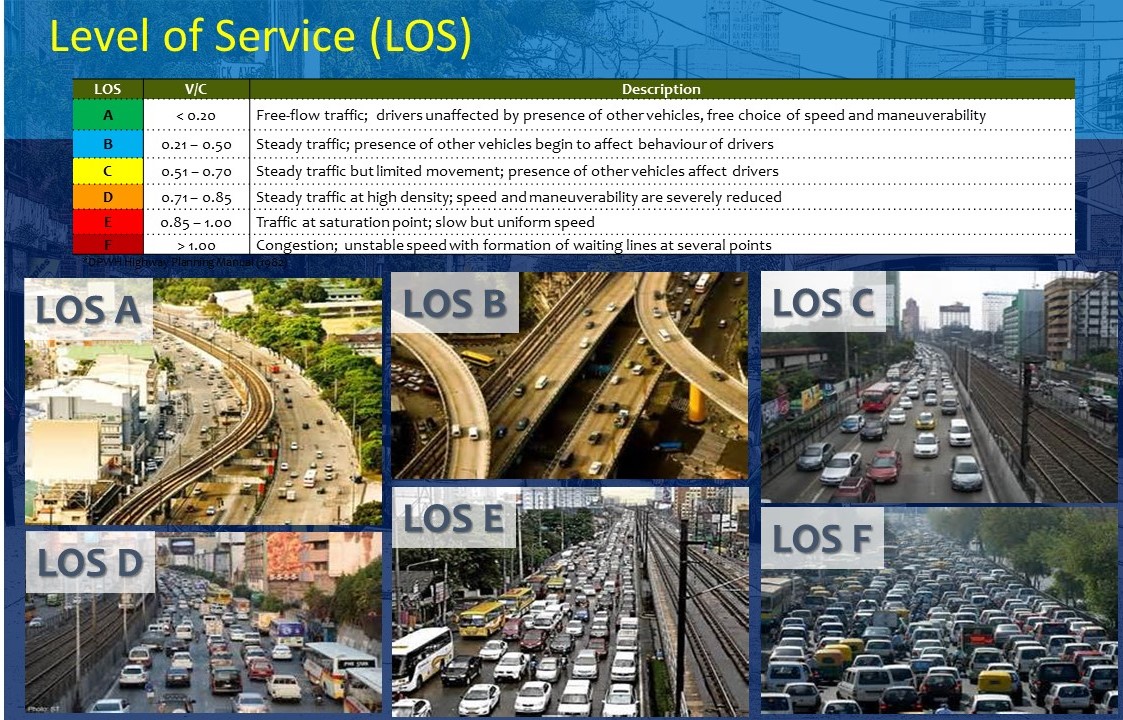
There are many other theories and concepts about how to perceive traffic. There is one major concept, however, that the researcher has been able to distinguish as very important. This concept being the labelling of the congested traffic present. The identification of what level of congestion is currently present will greatly influence the decision-making of the traffic agent. This is what the traffic engineers identify as LOS or Level of Service. The traffic light system that is to be developed by the research will be establishing most of its decisions from this LOS. The LOS is a qualitative measure which is used to relate the quality of motor vehicle traffic service which is present during the traffic phase [6]. The LOS is used to analyse roadways and intersections by categorizing traffic flow and assigning quality levels of traffic based on the vehicle speed and density.

Below is an image showcasing how to interpret the LOS. This was taken from the Highway Capacity Manual 2010 edition.



*Figure 2. Level of Service (LOS) chart*

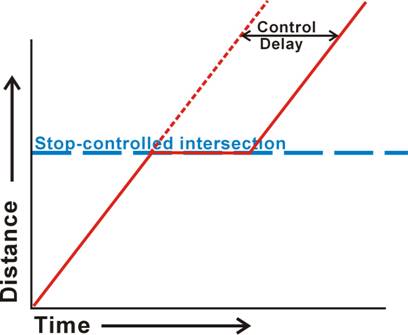
Determining the LOS means to interpret the average speed and total amount of vehicles that are present at the intersection. Below is another image that will showcase the different levels of LOS and its image correspondence.



*Figure 3. Level of Service*

The calculated LOS will be used by the entire system throughout to developing judgements or rule-based decisions. Another important factor that has been taking into account by the researcher is the capacity of the intersection. The capacity will base the knowledge of how much can the intersection accommodate, and this can be determine by upcoming vehicles to the intersection or lane.

Below is the formula for computing the capacity of an intersection which is according to the HCM 2010 edition.



*Figure 4. Control Delay chart*

**Traffic Road Rules**

**Random Forest Classifiers**

**REVIEW OF RELATED STUDIES**

There have been several studies conducted for the development of a system for solving traffic congestion. The researcher aims to tackle this problem and by doing so to provide a smart solution to it. So what the researcher is aiming to develop is a modified version of a traffic simulation system which will manipulate the traffic light system. By making use of the traffic light system, it may show what would become the favourable route to take in order to prevent or eliminate traffic congestion. Maybe perhaps it’ll be a step forward development of a better and faster solution to eliminating traffic congestion, as stated by the Head of TEAM.

**Improving Traffic Signal System**

The idea of improving the traffic light system was a subject that was discussed together with the head of TEAM. By improving the traffic light system and by therefore, making it dynamic. The probability of human error would lessen, and then the manipulation of the traffic lights would become more calculated and processed. The advantage of which would be the speed and reliability of the process of gathering the right output.

A research funded by the U.S. Department of Transportation about improving the operations of traffic signals at its city states was made because of the uprising of issues being experienced by motorists with regards to congested traffic. The issues were concerning about the time and money that were being spent and lost by the day-to-day motorists. The research was conducted by Juan M Rosales together with Associates P.C. The research would then be made into a primer which would be discussing how traffic signals would become essential tools for improving traffic flow. In this primer, it would discuss calculations and proper management of traffic signals, and at when would these traffic signal be proven to be most effective for traffic intersections. It also highlighted the fact of how the traffic agent personnel is an essential factor to the delivery of a traffic signal output, and that they require proper training to performing their job [7].

Another study that had sought to improve the traffic light system was one conducted by Qing He. K. Larry Head and Jun Ding. Their research discussed the processes of actuated-coordination and multi-modal priority control via simulation. These two have differed in terms of function and process, but they shared the same goal which was to improve the traffic signal system. Both made use of priority methods to lessen the delay of transportation vehicles like buses and pedestrian delay. Their study made use of a formulated request-based mixed-integer linear program that would accept multiple priority requests from vehicles inputted to the simulation. By using vehicle-to-infrastructure (v2i) communication, the priority requests from vehicles such as ambulances, commercial trucks and pedestrians would be accepted and then measured. The priority requests are only given to selected variables which were labelled upon input by the user. As the priority requests are being processed, the simulator would also be taking into account the coordination and actuation of the remaining vehicles. Now, as all the variable data are being processed, two outputs would be produced with the use of the two control systems. These outputs would be based on different objectives that were distinguished by each of the control systems. Each objective had respective penalties to how the performance or activity of the vehicles will progress. These penalties serve to manipulate the time given for the simulation to finish, so if the simulation failed to acquire the proper outputs then a penalty must be given. Along with this, the running signal coordination system will still be running to adjust itself as it will accept additional priority requests. And although the two control systems had displayed clear signs of difference with how they handled and processed the data. The research was able to highlight the strengths of the two and how they function suitably to the situation given [8].

**Smart Traffic Signal System**

To develop an improved traffic signal system would be to develop a ***smart traffic signal system***. It is labelled “smart” because it will perform procedures and processes all on itself with the use of machine learning algorithms. Machine learning is a famous study of algorithms that are used to create artificial intelligence. Intelligence that is able to eliminate certain human interaction because of how the system has been constantly learning to perform [13].

A study conducted by Bilal Ghazal, Khaled ElKhatib, Khaled Chahine and Mohamad Kherfan discussed the development of a smart traffic light control system that would fill the gaps of what the current traffic light system had. In their study, they expressed the various situations that the traffic agent may overlook during a traffic congested situation. One of which would be the passing of emergency vehicles; during this situation the traffic agent may need to calculate the right pattern to properly accommodate the situation and prevent the occurrence of congestion along the process. But then, the calculations may be either correct or lacking, for which it would be able to assist the emergency vehicle but what of that will follow. Their study expressed the use of sensors that will check the volume and speed of the vehicles present, and how this will be sued by the microprocessors to create a dynamic assignment of the traffic lights. The dynamic assignment of the traffic lights will be expressed via different levels which served to accommodate certain traffic situations. [9].

Another study also expressed the idea of the development of a traffic light signal system that is able to accept signals by vehicles on the road. This study was conducted by John Carl Mese, Nathan, J. Peterson, Rod David Waltermann and Arnold S. Weksler all of which are respected traffic engineers. Their study expressed the possibilities of traffic lights built with sensors to accept the data being sent by the vehicles on the road. By accepting the data, the traffic light would be able to assess the proper sequence of traffic lights and time allocation [10]. Their proposed system is another method of improving the traffic light system which intends to lessen human interaction, and at the same time it provides the user with the data. The presentation of the data accumulated is another advantage of the system.

**SUMO – Simulation of Urban MObility**

There are various simulation software up to date that are able to reduplicate a traffic scenario. It may be a one-way, two-way, a multilane highway, an intersection and even a roundabout. The difference with those simulators with SUMO is that they aren’t based on actual locations or even real schemes of places. That difference is grave with respect to improving the traffic flow; the state of the art simulators are those that will be able to reduplicate the scenarios of actual locations. Therefore, giving the real-life effect, for which is indefinitely present with advance traffic simulators.

The current traffic simulator that is currently being marked as the true “state of the art” by the researcher is SUMO or Simulation of Urban Mobility. SUMO is the current state of the art simulation software that is able to make use of Open StreetMap. The software is able to build the exported data from the map, or the XML, which would then showcase actual locations of real places present in the world. This then allows the possibility of creating traffic scenarios, and being able to control the output and basing it for any traffic situations.

The only downside to the simulator is the fact that it can’t integrate traffic light timer allocation or algorithms. The traffic light system present is not dynamic, and thus, it is only static in nature. The possibility of integrating SUMO with algorithms for purpose of Data Science is also made impossible because its limitations. SUMO is merely a static traffic simulator, but a decent one at best. There was an article that showcased the recent development and various applications to how SUMO can be used. It stated that SUMO is able to present data from test traffic cases that have been implemented. The said data is able to be tabulated and present in various forms of graphs and charts [14].

**PROJECT OBJECTIVES**

The study aims to develop a traffic simulator which will aid with providing the traffic agent with insights and analysis with regards to congested road traffics found on the streets of Mandaue City.

More specifically it aims to:

1. Simulate a traffic scenario before it became congested and will display how it leads to becoming congested
2. Make use of the Mandaue Road and Traffic Systems, to serve as a convenience for the traffic agent
3. Make use of machine learning algorithms to develop automated traffic lights and proper timer allocation
4. Develop a dynamic traffic light system that is capable of analysis and providing insights to traffic agent; to remove biases and flaw judgment

**PROJECT SCOPE AND LIMITATION**

The study is focused on developing a traffic simulator system that will be managing traffic intersections in Mandaue City, with traffic lights, and with the goal of improving the traffic flow. The simulator will make use of areas found only in Mandaue City and the locations that are to be used by the simulator would be those that are saved already in the database, and these locations will be based upon the city’s Traffic Code and Ordinances and the ICU LOS.

By making a traffic simulator system, the TEAM can witness how a certain traffic situation will take place and will also see how the traffic simulator system will handle the traffic with its time allocation for the traffic lights. The researcher will be basing the simulator system at a location where the current data center (a large group of networked computer servers that are typically used by organizations for the remote storage, processing, or distribution of large amounts of data, Google) is at the forefront of the improving traffic management system for Cebu. It is in Mandaue City, that more sophisticated management of traffic can be found being made done by TEAM. Their roads are amongst some of the many here in Cebu which possesses a large number of traffic lights and it is the ideal area for the conduction of the study because of its advanced road regulations, laws and ordinances. With the support of TEAM, the conduction of the research at this city location will become a smooth flowing one.

The traffic simulator system won’t be accepting inputs of the following:

1. Tragedies or accidents during the simulation because it is entirely beyond the scope of what the researcher intends to accomplish. It will only accept the inputs of traffic rules, vehicles, and traffic lights and then produce a time allocator that will make a more fluid traffic transition for the current traffic by analysing the volume of the inputted vehicles along with their time to reaching the point of junction and,
2. Pedestrian crossings during the simulation because the focus and purpose of what the researcher is aiming to develop will only be handling motor vehicle rules and traffic light regulations.

As major traffic congestion is recorded to be done at urban places/environments, the traffic simulator will be made based on intersections located at Mandaue City itself. The traffic simulator will follow the given rules of the road about when to change lanes for turning, one way, left turn and right turns. The traffic simulator system will not be made adjustable as the simulator is currently running, it won’t accept new inputs unless if it is being set as a new simulation because the simulator only makes use of the current presented elements and is not yet that intelligent to easily accept sudden or new alternations.

As the whole traffic predicament is brought to consideration into documents, by the Mandaue City Traffic Enforcement Agency, there are certain hours on the clock where the traffic is at its heaviest, and that the traffic light and traffic counter would be set a predefined amount. The simulator will not be adding the time situation at a particular intersection. Inputs such as “rush hours” or “working hours” will not be accepted by the simulation system because such factors although may seem beneficial for the whole concept of improving the overall traffic system, but as the traffic simulator is concerned it will not accept such events as it proves to be not a priority for the simulation project.

Finally, the simulator is aimed to be used by traffic agents or any member of the TEAM only because it is developed for the sole purpose of to be used by traffic personnel that will be making use of it to improve the traffic system. Users of other than the TEAM will not be able to access the simulator and you would need an existing account from the TEAM database if you were to use the simulation system. The simulator will be made use by officials of the TEAM, and it will be further discussed with along the simulator’s significance if it would be made use to the public as it is commended by the TEAM themselves.

**RESEARCH METHODOLOGY**

In order to get the actual feel of the Mandaue Road System to be made use as the field of simulation, the researcher is making use of the Open Streetmap (OSM) API for gathering the variable/s needed for the simulation. To make use of the OSM API, the researcher parsed it from its original XML file. By parsing the XML file that user would then be able to extract the nodes or the variables that would be used for the simulation, which is the development of the roads, locating traffic lights, and respected directions that are to be followed by the vehicle/s.

The nodes that are initially to be parsed and then extracted would be the RoadType and the NodeType of the OSM XML file. With these nodes, the researcher is able to build a road and is able to extract data from the OSM API such as the location of traffic lights and distinguishing main roads, one-way roads, junction and many more.

Now, the aim of the simulator is to develop dynamic traffic lights, or traffic lights that are not static or that aren’t reliant on human intervention, for various traffic situations that will undergo at different lanes. Its other output would also be the dynamic and proper time allocators that will be made use to deliver a better traffic flow for the traffic lanes.

The initial input that would be required for the simulator would be the location for the traffic intersection. The location needs to be already saved at the database in order for it to be accessed. These intersections will be saved using the coordinates of latitude and longitude, and the saving of intersection locations will be made do at the Admin’s view.

After inputting the location, the sidebar navigation menu will display options to add inputs for vehicle/s and traffic lights. The user is then able to drag n’ drop the new and available inputs to the traffic lane intersection. Once, the inputs are all settled the simulator would then be available to start and the following would be computed: LOS, capacity, Average Vehicle Speed per lane and Time Allocation.

The LOS will be calculated by calculating the volume capacity of a lane based upon the Highway Capacity Manual, 2000.

**Cap = Base Cap \* N \* fhv \* PHF\* fp \* fg**

Where:

Cap = Capacity in terms of vehicles per hour.

Base Cap = Base capacity in terms of passenger cars per hour per lane (varies

by facility type).

N = Number of through lanes.

fhv = Heavy vehicle adjustment factor (varies by facility type, vehicle mix, and

grade).

PHF = Peak-hour factor (the ratio of the peak 15-minute flow rate to the average

hourly flow rate).

fp = Driver population adjustment factor (used for freeway and multilane only).

fg = Adjustment factor for grades (used for two-lane highways only).

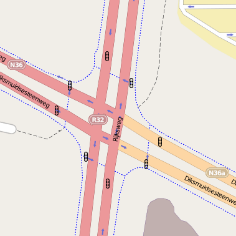
The LOS A is where the traffic is at its most free and there is no congestion and the LOS F is where the traffic is at its worst and there is severe congestion. The amount of second to give per LOS based upon the HCM thresholds, based on the HCM 2000, will be shown below.

|  |  |
| --- | --- |
| **LOS** | **Control Delay per vehicle (seconds per vehicle)** |
| **A** | ≤ 10 |
| **B** | > 10-20 |
| **C** | > 20-35 |
| **D** | > 35-55 |
| **E** | > 55-80 |
| **F** | > 80 |

*Table 1. Motor vehicle LOS thresholds at signalized intersections*

After the LOS is computed and after analysing the other lane situations. The amount of time given to the traffic whose LOS has been analysed will be given at an amount of time as stated above. For LOS A, a threshold of fewer than 10 seconds would be given whilst at the most congested of the other LOS, LOS F will be given a threshold of more than 80 seconds considering the number of vehicles on the lane.

After getting the LOS from each of the lanes from the intersection, the simulator will be using the machine learning algorithm, Random Forest**.** This algorithm will be the backend of the whole simulator, wherein it will be basing the proper time allocator for the traffic with the use of the LOS, average speed and average vehicle amount.

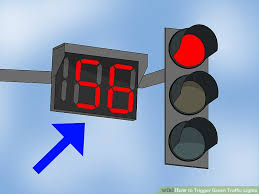
**INPUT PROCESS OUTPUT DATABASE**

**iTraffic Database**

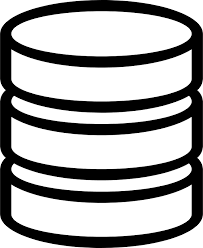
Location

**Simulation Map**

**Parser**



Vehicle



Road Rules

**Dynamic Traffic Light**

**and**

**Proper Time Allocation**

Vehicle Speed, Type, and Direction

As the simulator begins, a console will be displayed printing the data that is being processed by the simulator. The vehicle speed, average vehicle speed, LOS, capacity value and per lane time allocation are the data that will be shown throughout the whole simulation. This is to serve as an extra basis for understanding the overall data that has been accumulated and will be used for the traffic enforcers as knowledge and information basis to improve their road traffic management system.

The collection of data which serves as the basis for the whole simulator to use will be made done via the simulator itself. As the simulator is running, the data processed will be the very data that is being handled on the spot. This will be one of the functions that will make the whole simulator “dynamic”.

**CHAPTER II**

**SOFTWARE REQUIREMENTS AND DESIGN SPECIFICATIONS**

This chapter describes the requirements that the study intends to accomplish. It also presents the processes of the system to be able to achieve the said requirements. This chapter includes the Use Case Model, Use Case Narratives, Activity Diagrams, and Class Diagram to illustrate the design of the system.

**USE CASE DIAGRAM**

User

System

*Figure 4. Use Case Model*

The diagram above represents the use case diagram of the study. The user of the simulator will select a location for the simulation and after the user will input vehicles with their corresponding properties. These properties being one of them is the speed of the vehicles. After the user will start the simulation, the output will be the inputted car/s and traffic light/s that will re-enact an actual traffic situation.

**Use Case Narrative: UC 01 Location Selection and Build Map**

|  |  |
| --- | --- |
| Use Case | Select location for simulation |
| Actors: | Traffic personnel |
| Purpose: | Place for the simulation |
| Overview: | This method will allow the user to select the location for the whole simulation to take place, and would then be able to build the map for simulation |
| Type: | Essential |
| Precondition: | Access granted by the traffic officers |
| Postcondition: | A smooth traffic flow transition will be displayed |
| Flow of Events | |
| Actor Action | System Response |
| Input location for simulation  Build map | 1. The user will be asked for a location input for anywhere in Mandaue City at a input field at the sidebar navigation menu. |
| 1. It will display the area itself via OSM |
| 1. After the map interface is displayed, the action to add input/s of cars and/or traffic light/s will now be made accessible at the sidebar navigation menu. |

**Use Case Narrative: UC 02 Vehicle Attribute Input**

|  |  |
| --- | --- |
| Use Case | Input vehicle attributes |
| Actors: | Traffic personnel |
| Purpose: | For analyzation and traffic flow purposes |
| Overview: | This is where the user will input a vehicle/s attributes. The newly inputted vehicle will then be available and accessible to be drag n’ dropped to the maps. |
| Type: | Essential |
| Precondition: | User needs to input a location for simulation |
| Postcondition: | User will be able to add vehicles to the simulator |
| Flow of Events | |
| Actor Action | System Response |
| Add vehicle | 1. The system will ask for input regarding the vehicle |
| 1. After, the vehicle inputted will be displayed for use. |
|  |

**Use Case Narrative: UC 03 Starting Traffic Sequence**

|  |  |
| --- | --- |
| Use Case | Start Traffic Sequence |
| Actors: | Traffic personnel |
| Purpose: | To produce dynamic time allocation of traffic lights |
| Overview: | This will produce the output of the simulator, the dynamic traffic counter and traffic lights |
| Type: | Essential |
| Precondition: | Vehicles must be inputted to lane/s |
| Postcondition: | Output will be released |
| Flow of Events | |
| Actor Action | System Response |
| Starting the traffic sequence | 1. The user will be displayed with the traffic sequence |
| 1. System will display the output after the traffic sequence is finished |
|  |

**Use Case Narrative: UC 04 Inputting Street Rule/s to Vehicle**

|  |  |
| --- | --- |
| Use Case | Adding Street Rules to Simulator |
| Actors: | Traffic personnel |
| Purpose: | For training the simulator with actual rules from the streets |
| Overview: | This will add that “real-life” situation feel to the simulator which is by adding actual street rules, and by adjusting the simulator with the inputted rules given like “one-way”, “no left turn”. |
| Type: | Essential |
| Precondition: | Vehicles may or may not yet be inputted to lane/s |
| Postcondition: | Simulation will run with these rules in play |
| Flow of Events | |
| Actor Action | System Response |
| Adding a Street rule for a particular lane | 1. The vehicles will be notified about prescribed rule through event handlers or a flag handler through which will be triggered by new input/s from the user. |
| 1. System will take the sign/s into account for the output basis for display. |
|  |

**Use Case Narrative: UC 05 Stop/Pause Traffic Simulation**

|  |  |
| --- | --- |
| Use Case | Stop/Pausing the Traffic Simulation |
| Actors: | Traffic personnel |
| Purpose: | For adding changes to the current simulation or to terminate the entire simulation to start over |
| Overview: | This will give the user the opportunity to start over again or to add either more cars, traffic rules or traffic lights to the actual simulation |
| Type: | Optional |
| Precondition: | The simulation must have already ran |
| Postcondition: | The simulator will either start over again or will accept new inputs from the user |
| Flow of Events | |
| Actor Action | System Response |
| Pause Simulation/Stop Simulation | 1. If pause, then the system will accept additional inputs by the user |
| 1. If stop, then the system will show a pop-up window for confirming termination process and start a new simulation |

**ACTIVITY DIAGRAMS**

|  |  |  |  |
| --- | --- | --- | --- |
| Search location for Traffic Simulation | | | |
|  | **User** | **System** | **Output** |
| Build map  Inputs location | Extract parsed OSM nodes  Search in Database  Verify Inputted Location | The OSM Map interface will be built and ready to start the simulation  The name of location will be displayed at the input box |

*Figure 5. Search location and Build map for simulation*

Figure 5 illustrates how the user actor of the system will search for a location to start doing the traffic simulation. The initial requirement would be if that location is saved to the database since roads that are to be made use of need to belong to the Mandaue Traffic Road System, only those that belong may be visited for simulation. Then after inputting a valid location for simulation, the system will redirect from the default location which is Mandaue City Hall and to the traffic intersection.

The locations will be verified if whether they are found in the database for locations to be used in the simulation. The system will already be expecting specific locations for the simulation action, and if the inputted location/s are not saved in the database than a pop-up will be shown with the message informing the user that the inputted location is not saved in the database and it can’t be accessed.

After that, the user is then tasked to press on the “Build map” button to extract the nodes and then so build the map interface.

|  |  |  |  |
| --- | --- | --- | --- |
| Input Vehicles for Traffic Simulation | | | |
|  | **User** | **System** | **Output** |
| Input Vehicles Speed  Input Vehicles Type  Input Vehicles ID | Save to Database | Appear in sidebar menu as an available option |

*Figure 6. Input vehicles for simulation*

Figure 6 illustrates the occurrence when traffic enforcer actor will add an input of vehicle/s to the simulator system. As, the actor will add an input it will automatically be saved to database and then made accessible at the sidebar navigation menu for the drag n’ drop to start simulating.

|  |  |  |  |
| --- | --- | --- | --- |
| Drag N’ Drop Vehicles and Input Traffic Lights for Simulation | | | |
|  | **User** | **System** | **Output** |
| Drag n’ drop traffic lights  Drag n’ drop vehicles | Save to Database  Accept dragged input | Traffic Simulation button will be available to click |

*Figure 7. Drag n’ drop vehicles and traffic lights to Traffic Simulator*

Figure 7 illustrates how the system will accept the input into the traffic simulator. Once, a vehicle or traffic light is dragged into a lane it can be deleted/transferred as to what the user prefers. The user is able to do such before starting the simulation, but during the simulation the user can’t affect anymore the placement of the vehicles and traffic lights.

|  |  |  |  |
| --- | --- | --- | --- |
| Simulator System will Process the Traffic Flow | | | |
|  | **User** | **System** | **Output** |
| Start the Simulation | Compute average vehicle speed  Apply Traffic Signal Rules  Determine the best traffic time  Determine LOS  Compute ICU | Smooth Traffic Transition  through Dynamic Time  Allocation |

*Figure 8. Traffic Processing Module*

Figure 8 shows the overall process of the traffic simulator during the actual simulation. The process begins as soon as the traffic enforcer actor will press “Start Simulation”, then the vehicles will move and upon entry to the intersection the ICU LOS will be computed with the factors involving vehicle quantity, vehicle speed and arrival time to intersection. The determining of best traffic time will be based upon lane LOS and with the given basis of amount of time. This will then determine the lane to initially give the right of way. After, the simulator will abide by the traffic signal manual to the amount of time given to transition from current traffic light signal to the next and following to the next lane to be given right of way.

|  |  |  |  |
| --- | --- | --- | --- |
| Traffic Simulator will Display Traffic Report | | | |
|  |  | **System** | **Output** |
|  | Average vehicle speed  Traffic time given  Determined LOS  Computed ICU | Traffic Statistics Report |

*Figure 9. Traffic Simulation Statistics Output*

Figure 9 shows the data that will be taken from the simulator system to be shown to the traffic enforcer actor for analysis. The output will be used to improve traffic flow and traffic management.

|  |  |  |  |
| --- | --- | --- | --- |
| Enabling/Disabling Traffic Light/s to Simulator | | | |
|  | **User** | **System** | **Output** |
| Enable/Disable Traffic Light/s | OSM node Traffic Lights = on/off | Traffic Light/s are enabled/disabled |

*Figure 10. Enable/Disable Traffic Light/s*

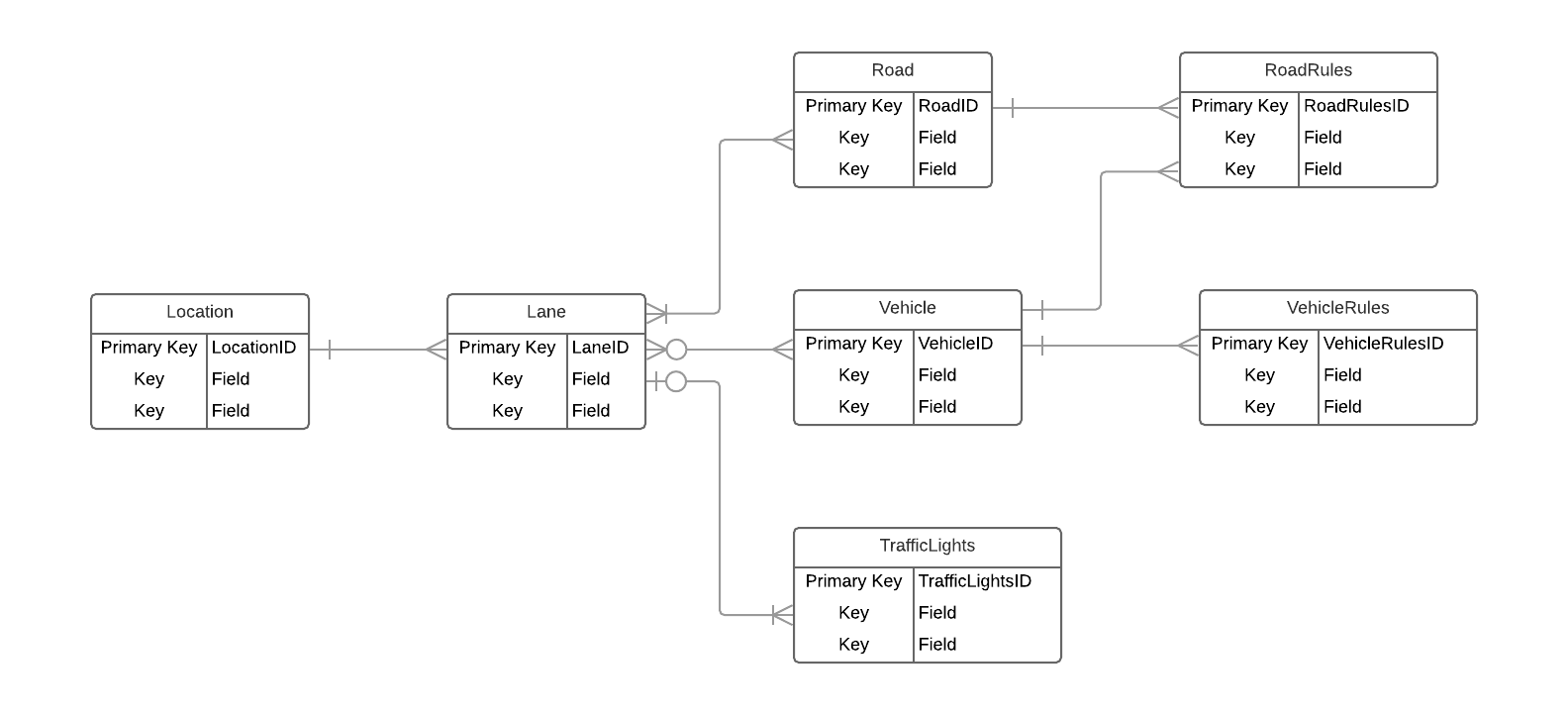
Figure 10 shows the process of enabling or disabling the traffic light/s of the simulator. These traffic light/s are already linked to the Open StreetMap API and as such its location found in the simulator will be the same as the exact placement of it in the actual road. The user is able to enable/disable the traffic light/s by simply just pressing on them.

|  |  |  |  |
| --- | --- | --- | --- |
| To Stop/Pause the Traffic Simulation | | | |
|  | **User** | **System** | **Output** |
| Stop the Simulation  Pause the Simulation | Display confirmation alert  Halt motor vehicle/s | An alert box will be shown asking for confirmation to stop the current simulation and to start a new simulation  The current traffic will display motor vehicle/s stop moving |

*Figure 11. Stop/Pause Simulation*

Figure 11 shows the process of when the user will attempt to stop/pause the simulation. At first, if the user wishes to pause the simulation, then the simulator will halt all vehicle/s and the user is allowed to input or add more vehicle/s to the simulator, and is also able to enable/disable traffic light/s. Secondly, if the user wishes to stop the simulation, then an alert window will pop-up asking for a confirmation to stop the simulation and will immediately process to start a new simulation from scratch.

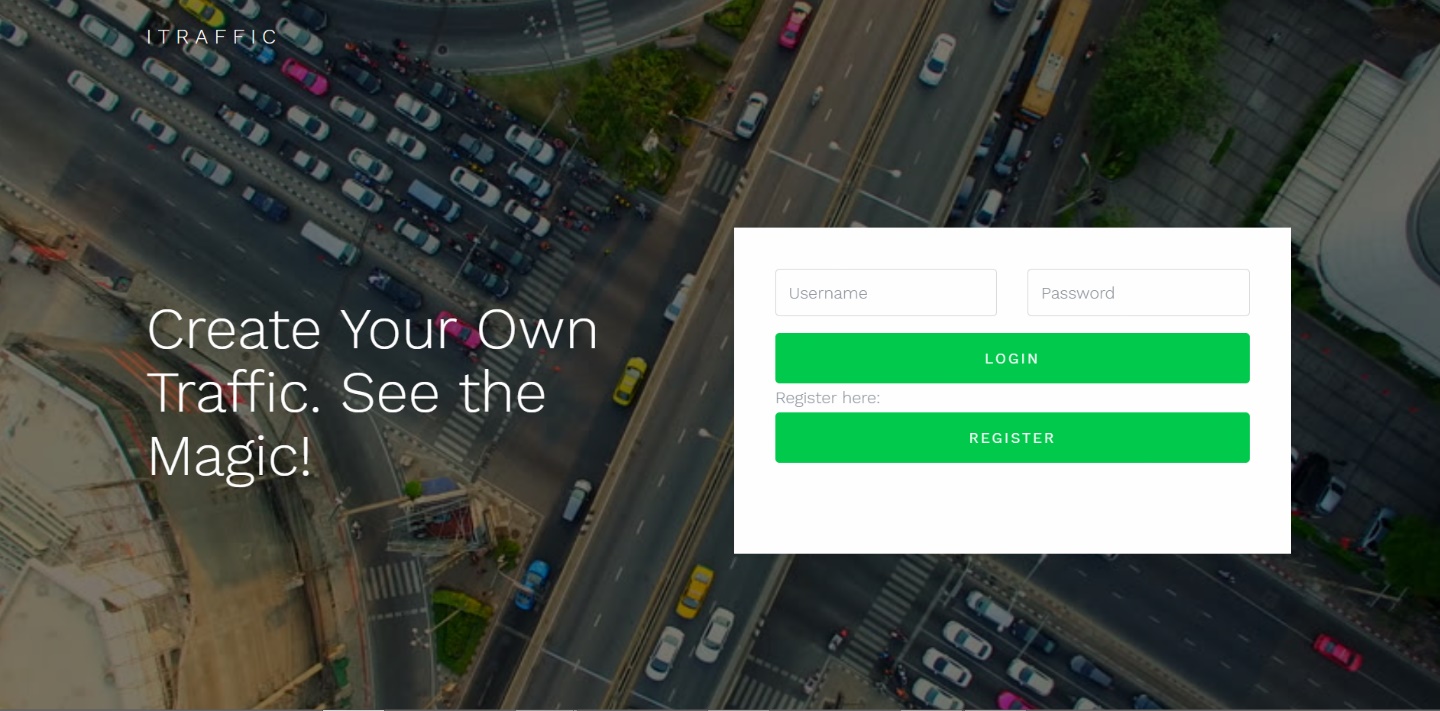
**ENTITY RELATIONSHIP DIAGRAM**



*Figure 12. Entity Relationship Diagram for the database*

Figure 12 presents the graphical representation of the database structure of the system before simulation. It states that the lane may have an input or zero or many car/s while a car may be directed to one or more lane/s. The traffic light may direct to one or more lane/s while a lane can only have one traffic light to be basing of to. Lastly, a car will follow the sequence of one traffic light and a traffic light may be used to control the sequence of one or more car/s.

**USER INTERFACE DESIGN**

****

*Figure 13. Loading GUI*

Figure 13 illustrates how the app will initially appear as it is being run at first. Here it will show a simple description of what the system is all about.

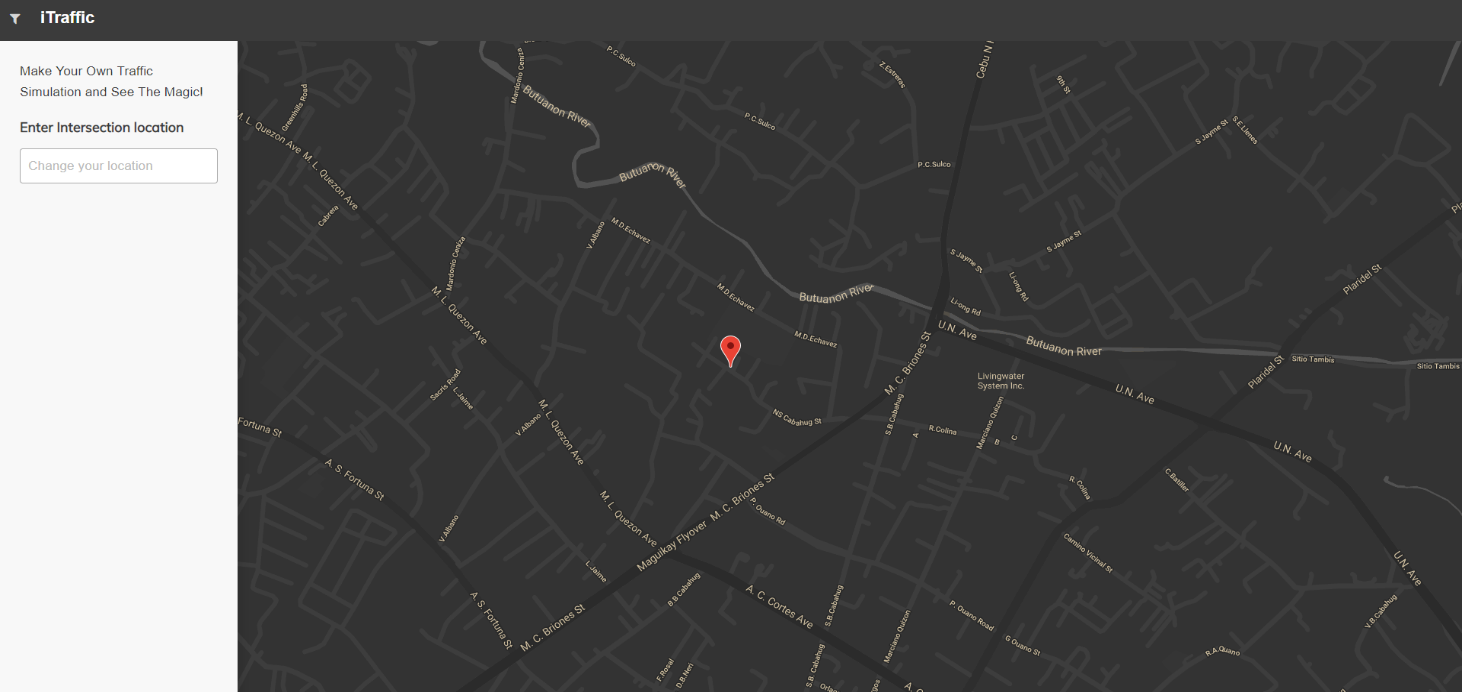
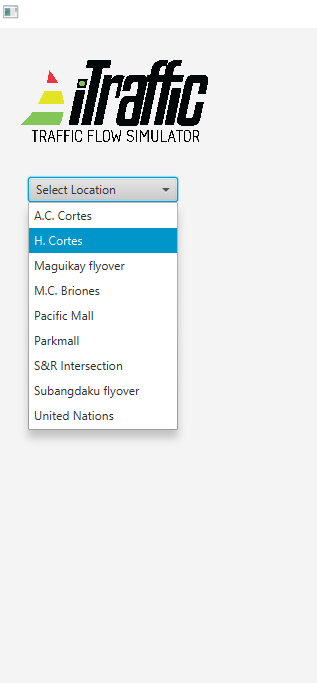
*Figure 14. Initial Simulator Dashboard GUI*

Figure 14 illustrates the landing of the simulator after its preloading phase. Here the maps will show the initial marker which is placed on top of Mandaue’s City Hall.



*Figure 15. Select Location*

Figure 15 illustrates the user interface of selecting a location for simulation. The locations displayed will be of ones which are saved to the database. If a location is inputted and it is not found at the database then the “map” will redirect to the default location and will display a message that the location inputted was not found at the database.

*Figure 16. Input Vehicles*

Figure 16 illustrates how to add attributes for a vehicle to be used by the simulator. In the simulator, the vehicle can be either added or deleted, so the user has full control of what to do and easy to use. The user is expected to input a vehicle’s type and speed for example: “SUV”, “5”, “Automobile”, the speed of the vehicles will be interpreted by k/h (kilometre per hour), if either of the expected attributes are missing than an error will be displayed since the system requires both attributes to be added to comprise a vehicle to be placed at the map simulator. The simulator would make use of the type and speed of the vehicle, in order to analyse them for the traffic light and time allocator algorithm.

*Figure 17. Drag n’ Drop Vehicles to Simulator*

Figure 17 illustrates how to perform drag n’ drop at the system. Drag n’ drop is easy to perform at the system, as such you simply need to click at the icon at the sidebar navigation menu and drop the icon at the lane found at the map interface.

*Figure 18. Drag n’ Drop Traffic Lights to Simulator*

Figure 18 illustrates how to add traffic light/s to the map interface. It basically functions the same as drag n’ dropping a vehicle to the map. The user would just have to click at the traffic light icon and drop it at the location where it is advised to be placed which is at the end of a certain lane because of the condition that vehicle/s are supposed to stop for a while as they await the system to display the traffic light and time allocator.

*Figure 19. Run Simulation*

Figure 19 illustrates the simulation running. The simulator will run as the user will press on the “Run” button at the top right of the frame.

*Figure 20. Stop/Pause Simulation*

Figure 20 illustrates how to stop or pause a simulation. The button for this function is the same one with running the simulator. It is indicated via colour green is “Run Simulation” and red is “Stop/Pause Simulation”.

*Figure 21. Time Allocator through Random Forest Classifier*

Figure 21 illustrates the expected output of the simulator which is the traffic light and time allocator. The simulator will display these via the traffic light and timer. The algorithm behind these outputs are made with the use of the Random Forest Classifier algorithm.

*Figure 22. Add More Vehicles to Simulation*

Figure 22 illustrates how to add more vehicle to the simulator as it is being paused. So, it is easy to perform because it is performed as how it is initially done which is to just press on the “Pause” button found on the top-right corner and then drag the icon of the vehicle you wish to add to the simulator. After you added a vehicle to the simulator, then it will be accepted to the simulator and saved to the database and after the simulator will re-adjust to the newly added data.

*Figure 23. Enabling/Disabling Traffic Light/s*

Figure 23 illustrates how to enable/disable the traffic light/s found at the simulator. It is easily made done by just simply clicking on the traffic light/s that you wish to be turned off/on.

*Figure 24. Presentation of Collected Data from Simulator*

Figure 24 illustrates the presentation of the data collected from the entire simulation which is the average vehicle speed, control delay, LOS, vehicles direction, throughput, time allocated per lane and traffic light respectfully, and lastly the status report of what is generally good or wrong of that certain traffic situation.

**CHAPTER III**

**SOFTWARE DEVELOPMENT AND TESTING**

**Development Software Platforms, Development Environment, and Tools**

The traffic simulation system has two components into it mainly: the simulation front-end and the back-end. The simulation front-end is where the user will interact with the system to create a traffic simulation. The simulation front-end is developed to be easy-to-use for the user, and this was developed with the use of the library JavaFX and the visual layout tool named Scene Builder. The back-end component is where the system will handle the data that is being produced from the front-end in order to create an output of an automated traffic light and time allocator. Simply, this is where the data will be collected for example the average speed of the vehicles, the LOS of the lane and directions of the vehicles. The system back-end will be processing all of this data and will perform the algorithm that was formulated by the researcher, that the right-of-way or the transition of traffic will be automated whereas, human interaction will still be needed and while the system’s back-end will be handling all of the actual processing, output, analysis and some pre-processing. The user will partake at the initial pre-processing which is at the beginning of running the simulation. For the back-end component, Java is used as the core programming language. The researcher made use of Java mainly because of how platform independent it is and how it can run on various operating systems. The development tool for the Java core is Eclipse IDE 2018-2019 with Java SE together also with JavaFX and Scene Builder. The database is created using MySQL as the Relational database management system. MySQL is handy and easy to coordinate with JavaSE and it is a simple and clean DBMS.

The simulator system’s back-end component, makes use of several external libraries for its development. The simulator is built with the use of OpenStreetMap API (API v0.6, 2009) and JavaFX libraries. The OpenStreetMap API enables the simulator to access actual visualization and location of the locations located in Mandaue City. Since the simulator is intended to automate the traffic road system of Mandaue City, it would need OpenStreetMaps to provide an exact and actual feel of the area since OpenStreetMap is based from live and actual data that was provided from the people from those areas. Also, JavaFX is on the verge of replacing Swing as Java’s default GUI library, and as such to cope with the trend, the researcher is making use of JavaFX because of how more and seemingly advanced it is becoming compared to Swing. JavaFX is used by the traffic simulator system as to be the GUI for the user and as the wrapper for the OpenStreetMap API. Alongside with JavaFX, it has its own separate frame or software for developing the GUI, which is Scene Builder. This software is made or designed as such to sync and work together with JavaFX as its personal yet separate JFrame. Scene Builder is a visual layout tool which is more advanced that Swing’s JFrame. This is a very handle tool which alike JFrame will allows users to drag and drop UI components to a work area. Scene Builder can also allow users to apply style sheets to their GUI so that they can bring out the beauty of their design.

**DEVELOPMENT AND TESTING PROCESS**

**DEVELOPMENT PROCESS**

**Data Retrieval**

The data to be retrieved that will be made used for the simulator would be the location name of the street that would be found at Mandaue City, the vehicle’s attributes mainly width, height, type and speed, vehicle’s lane and direction, and road regulations.

Since the developed system is a simulator, it will be retrieving data at a more unique manner. The data retrieval process of the simulator is made done during the ongoing run of the simulator by the user. The simulator will be retrieving the following data from the user which is: location, vehicle type, vehicle speed, lane of vehicle, and direction of vehicle. The simulator will be retrieving all these variables as soon as the user drops them over to the simulator.

Below are screenshots of data being collected during a simulation run.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Location | Bounds | | | |
|  | Minlat | Minlong | Maxlat | Maxlong |
| Pacific Mall | 10.3397800 | 123.9450600 | 10.3422100 | 123.9492700 |
| Maguikay Flyover | 10.3338600 | 123.9379900 | 10.3361600 | 123.9422000 |
| U.N. COR. D. M. Cortes St. | 10.3377700 | 123.9531300 | 10.3400700 | 123.9573500 |
| Cor. S.B. Cabahug/A.C. Cortes | 10.3280400 | 123.9438300 | 10.3326400 | 123.9522500 |
| Parkmall Intersection | 10.3237300 | 123.9336900 | 10.3260300 | 123.9379000 |
| S&R Intersection | 10.3179700 | 123.9294300 | 10.3202700 | 123.9336500 |
| Cor. A.S. Fortune/M.C. Briones | 10.3389200 | 123.9440500 | 10.3435200 | 123.9524700 |
| Cor. A.S. Fortune/H. Cortes | 10.3292500 | 123.9463100 | 10.3315500 | 123.9505200 |

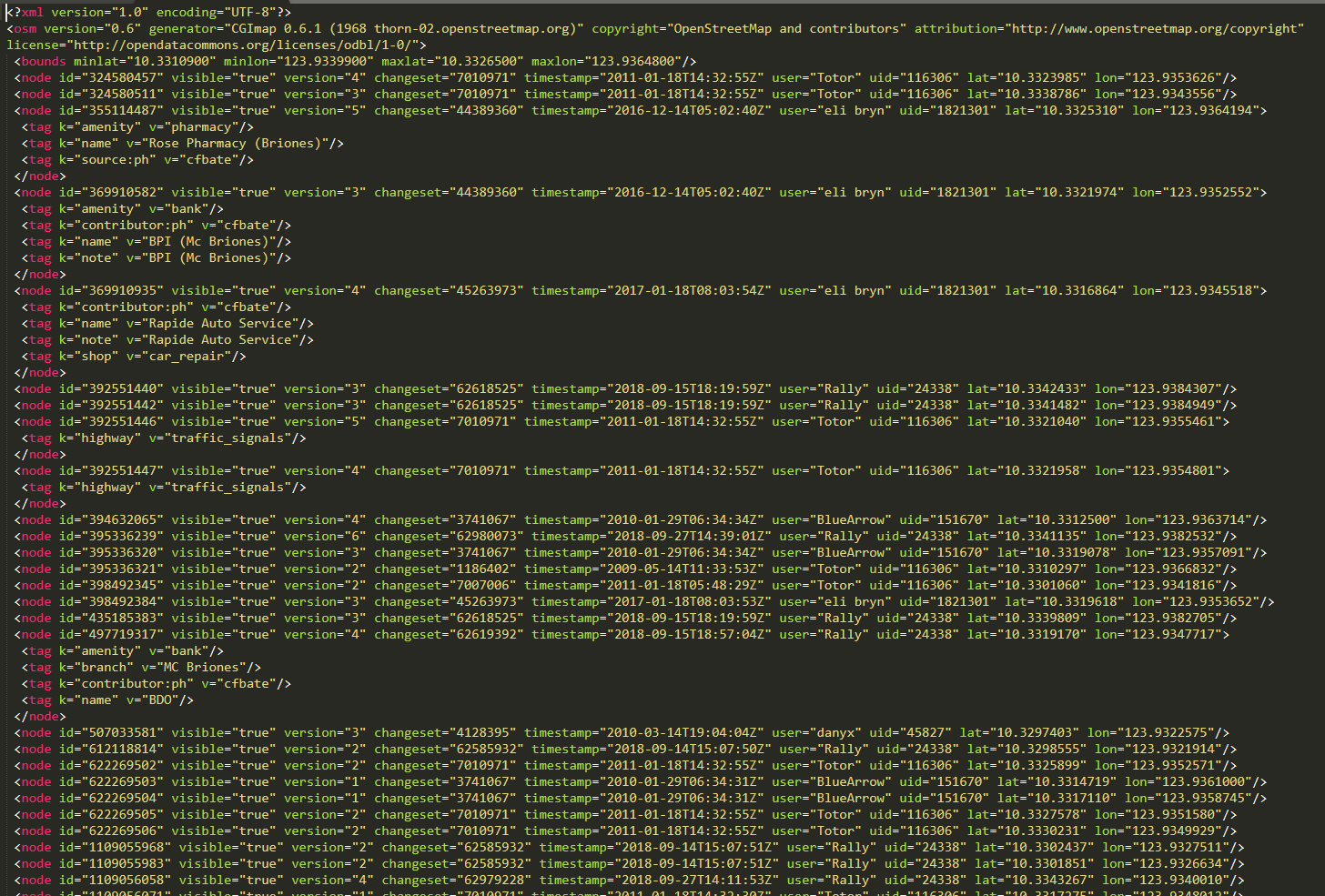
*Table 2. List of Intersections found in Mandaue City*

Table 2 presents the table for the intersection locations found in Mandaue City and are to be used for the traffic simulator. These locations will be stored for the simulator to use, and inputted locations that are not from the table will not be accessed.

**Data Preprocessing**

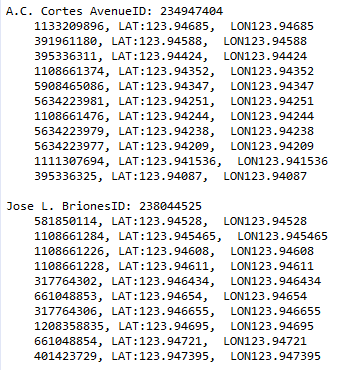
The initial process to be made done to developing the simulator would be to grab the data from the OSM API. These data are composed of the node types: junction and traffic light/s, and road types: motorway, primary, secondary, and tertiary.

Below is an image of the actual OSM XML file that is to be parsed and to be made use by the researcher to build road map and access OSM roadType and nodeType data.



*Figure 25. OSM XML file*

In building the actual road map using OSM API, the researcher will be making use of the road node coordinates. Below is a screenshot of the data collected to build a road for simulation.



*Figure 26. Parsed XML data*

Now, this part of the pre-processing phase is very crucial to the simulator because most of what is needed to run the simulator would be basing off the OSM API, and to properly assess and make use of the API, the researcher needs to be thorough on what variables to use and what he can make use. For example, one of the node types of the API would be the location of traffic light/s. Now, this is very important since the simulator would then now be able to project a simulation of traffic at a location with a traffic light and plus with the road type variable then the researcher would be able to discern the type of traffic that would normally undergo at that particular road. Hence, the labelling of the road types are as follows: ***primary*** being for the busiest routes and main roads being used, same with the ***secondary*** roads which are also labelled to be routes of similar busyness with the primary road type which means that these two are the urban road areas, the ***tertiary*** would be the road areas outside or beyond that of urban areas, which means that these are roads that would have lesser traffic, same goes for road type labelled ***unclassified***, which would mean that these roads are way off the attention of traffic or just that the traffic is not so busy and that the roads may be quieter than its other road types.

The data that would be collected by the simulator is a product of the user’s initial human interaction. The user will be required to input those variables to the simulator for it to function, and as such as it serves for its retrieval of data it at the starting point is that beginning of the processing of those data, mainly saving them to the database for it to be handled by the back-end component of the system. As the user will input the location for the simulation, then the system will begin to save the data for the whole simulation run, unless if it’s being ended than the database filled with simulation data will be refreshed. So after the input of locations, the user would now be presented with the maps, and then the user will have to add vehicles to input vehicle/s to run over the system, the vehicle/s would require an input of width, height, type and speed which are all important components to building a vehicle object. As those attributes are being saved to the database, next to be done by the user is to drag n’ drop the vehicle/s to the simulator, and as they begin to “run” the traffic simulation, the user is or may not be required to input road regulations and at the same time enable or disable traffic lights.

Now, as all of the user required actions are done, the system will now be implementing the LOS algorithm which is the initial running formula for producing the final output. The LOS algorithm will be making use of the speed, volume, and capacity of the vehicle and lane. The LOS is a major deciding factor to act as a basis to which lane should be given the right of way or to proceed to its destination. The traffic light and traffic timers will also be based at the LOS interpreted output because the proper timing and transition of those two instruments are crucial to providing a less congested traffic situation.

**LOS Algorithm**

The LOS algorithm is an important factor in developing the formula for producing an automated traffic light and time allocator. The LOS will be the balancing scale to which the Right Of Way algorithm will be basing its outputs and decision making to which particular lane will be given right to move to its designated direction and how much time will be given for it to pass over with respect to awaiting lanes and future lanes that will be occupied with new and other vehicles.

The back-end component will be handling the computations and based on the output it will discern to what LOS category Lane X belongs to, it may be from LOS A-F.

The basis for how this formula was formulated was derived from the Signal Timing Manual and the Intersection Capacity Utilization formula. The Signal Timing Manual stated that volume is one of the key components to arrive at a decision to which lane is allowed to pass through, made to go along the human knowledge, this was the right lane of thinking. But as the researcher saw that the speed of cars is also a pressing matter to look into with respect to road and traffic systems, it was not made done by certain traffic enforcers to keep track of the speed that was being brought out by the drivers of the vehicles. Along with that input was the Intersection Capacity Utilization, the researcher saw from this tool that the capacity of an intersection, road, or lane are also pressing components that should be made noticed to how the traffic situation should be handled. So with that information, the formula was formulated to also look into effect the capacity of the lane and how many volumes of vehicles can it hold up.

Here is the code snippet of the implementation of the formula to the simulator system.

**Intersection Capacity**

The intersection capacity is the formula used to determine the possible capacity that vehicle/s can occupy an intersection and also a lane.

Below is the formula for computing the capacity of an intersection.

Here is the code snippet of the implementation of the formula to the simulator system.

**Green Split Calculations**

This is the formula used for determining the length of time that is available for green signal indications on a traffic cycle run. This is an important factor to creating the dynamic traffic light and time allocator because it may serve as a service threshold to the random forest classifier as it will work out the distribution of right of way to lanes.

Below is the formula for the green split calculations.

**gi=(V/s)i/S(V/s)\*GT**

Where:  
gi = The length of the green interval for phase "i" (sec)  
(V/s)i = The critical flow ratio for phase "i"  
GT = The available green time for the cycle (sec)

Here is the code snippet of the implementation of the formula to the simulator system.

**Intersection Clearance Time**

The intersection clearance time algorithm is an important computational feature on the traffic simulator system that computes the amount of time given to a certain lane to pass over or move along its respective destination. This will make use of the already computed LOS per lane. The LOS is a key component for the rest of the computations that will be undergone throughout the simulation phase.

Below is the formula for computing for the clearance time to pass through the intersection.

**T = (LOS + L + W)/(1.47\*Vo)**

Where:

T = Intersection clearance time (sec)

Vo = Initial velocity (mph)

L = Length of the vehicle (ft)

W = Width of the intersection (ft)

Here is the code snippet for implementing the intersection clearance time algorithm to the simulator system.

**Right Of Way Algorithm**

**Random Forest Classifier**

**TESTING PROCESS**

The testing process is comprised of.

**Performance Testing**

**Data Retrieval Performance**

**Data Preprocessing and processing performance**

**Accuracy Testing**

**CHAPTER IV**

**SUMMARY, CONCLUSION, AND RECOMMENDATION**

**SUMMARY OF FINDINGS**

**CONCLUSION**

**RECOMMENDATION**

**FUTURE WORKS**

**BIBLIOGRAPHY**

**INTERNET SOURCES**

1. Graham, Ronald L., et al. "Concrete mathematics: a foundation for computer science." Computers in Physics 3.5 (1989): 106-107.
2. Barceló, Jaume. Fundamentals of traffic simulation. Vol. 145. New York: Springer, 2010.
3. Banks, Jerry, et al. "Discrete-Event System Simulation. International Series in Industrial and Systems Engineering." (1984).
4. Krajzewicz, Daniel, et al. "SUMO (Simulation of Urban MObility)-an open-source traffic simulation." Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002). 2002.
5. Leshem, Guy, and Yaacov Ritov. "Traffic flow prediction using adaboost algorithm with random forests as a weak learner." Proceedings of World Academy of Science, Engineering and Technology. Vol. 19. 2007.
6. Sokolowski, John A., and Catherine M. Banks, eds. Principles of modeling and simulation: A multidisciplinary approach. John Wiley & Sons, 2011.
7. Morales, J. M. Improving Traffic Signal Operations: A Primer. No. FHWA-SA-96-007. 1995.
8. He, Qing, K. Larry Head, and Jun Ding. "Multi-modal traffic signal control with priority, signal actuation and coordination." Transportation Research Part C: Emerging Technologies 46 (2014): 65-82.
9. Ghazal, Bilal, et al. "Smart traffic light control system." Electrical, Electronics, Computer Engineering and their Applications (EECEA), 2016 Third International Conference on. IEEE, 2016.
10. Mese, John Carl, et al. "Smart traffic signal system." U.S. Patent No. 6,989,766. 24 Jan. 2006.
11. Koonce, Peter, and Lee Rodegerdts. Traffic signal timing manual. No. FHWA-HOP-08-024. United States. Federal Highway Administration, 2008.
12. SrinivasaSunkari, P. E. "The benefits of retiming traffic signals." ITE journal (2004): 26.
13. Alam, J., and M. K. Pandey. "Design and analysis of a two stage traffic light system using fuzzy logic." J Inform Tech Softw Eng 5.162 (2015): 2.
14. Krajzewicz, Daniel, et al. "Recent development and applications of SUMO-Simulation of Urban MObility." International Journal On Advances in Systems and Measurements 5.3&4 (2012).Taha, Mohammad A., and Laheeb Ibrahim. "Traffic simulation system based on fuzzy logic." Procedia Computer Science 12 (2012): 356-360.
15. Koza, John R., et al. "Automated design of both the topology and sizing of analog electrical circuits using genetic programming." Artificial Intelligence in Design’96. Springer, Dordrecht, 1996. 151-170.
16. Petkovic, Dragutin, et al. "Setap: Software engineering teamwork assessment and prediction using machine learning." Frontiers in Education Conference (FIE), 2014 IEEE. IEEE, 2014.
17. Koonce, Peter. Traffic signal timing manual. No. FHWA-HOP-08-024. United States. Federal Highway Administration, 2008.
18. Mathew, Tom V., and KV Krishna Rao. "Introduction to Transportation engineering." *Civil Engineering–Transportation Engineering. IIT Bombay, NPTEL ONLINE, http://www. cdeep. iitb. ac. in/nptel/Civil% 20Engineering* (2006).
19. Albagul, A., M. Hrairi, and M. F. Hidayathullah. "Design and development of sensor based traffic light system." American Journal of Applied Sciences 3.3 (2006): 1745-1749.
20. Liang, Xiaoyuan, et al. "Deep reinforcement learning for traffic light control in vehicular networks." arXiv preprint arXiv:1803.11115 (2018).
21. Lämmer, Stefan, and Dirk Helbing. "Self-control of traffic lights and vehicle flows in urban road networks." Journal of Statistical Mechanics: Theory and Experiment 2008.04 (2008): P04019.
22. Gradinescu, Victor, et al. "Adaptive traffic lights using car-to-car communication." Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th. IEEE, 2007.
23. Mese, John Carl, et al. "Smart traffic signal system." U.S. Patent No. 6,989,766. 24 Jan. 2006.
24. Alexander, Rob, and Tim Kelly. "Supporting systems of systems hazard analysis using multi-agent simulation." Safety science 51.1 (2013): 302-318.
25. Behrisch, Michael, et al. "Sumo–simulation of urban mobility." The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain. Vol. 42. 2011.
26. Kamusoko, Courage, and Jonah Gamba. "Simulating urban growth using a Random Forest-Cellular Automata (RF-CA) model." ISPRS International Journal of Geo-Information 4.2 (2015): 447-470.
27. Osigbemeh, Michael, Michael Onuu, and Olumuyiwa Asaolu. "Design and development of an improved traffic light control system using hybrid lighting system." Journal of traffic and transportation engineering (English edition) 4.1 (2017): 88-95.
28. Chandler, David L. “Traffic lights: There’s a better way” (2014) MIT News Office
29. Crommelin, R W "Employing Intersection Capacity Utilization Values To Estimate Overall Level of Service", Transportation Research Board, 2010, accessed December 8, 2010.
30. Petkovic, Dragutin, et al. "Using the random forest classifier to assess and predict student learning of Software Engineering Teamwork." Frontiers in Education Conference (FIE), 2016 IEEE. IEEE, 2016.
31. Breiman, Leo. "Random Forests Leo Breiman and Adele Cutler." Random Forests-Classification Description (2015).
32. Margiotta, Richard, and Scott Washburn. Simplified Highway Capacity Calculation Method for the Highway Performance Monitoring System. No. PL-18-003. 2017.