**Project Report**

Project TrAn-SiSS

Traffic Analyzers and Signal System Simulator

**Project Name:- Traffic Analyzers and Signal System Simulator**

**Project abbreviation:- TrAn-SiSS**

**Date of Commencement:-28/02/2015**

**Date of Completion:- 20/04/2015**

**Project Statement** :-

Traffic Analyzers and Signal System Simulator(TrAn-SiSS) features a unique analytical Network Model, it is capable of incorporating different types of intersection and simulating diverse Traffic Movement. TrAn-SiSS allows to design and simulate vehicle traffic for a map designed by the user. It facilitates the user to design a road map network consisting of streets and junctions. Once a road map network is ready, during the simulation the street disciplines and all movement data can be specified, and all lane flow calculations, capacity, performance estimation and signal timing calculations are carried out by various algorithms formulated by the developing team.

**Team Details**

|  |  |
| --- | --- |
| **Member Name** | **Email ID** |
| Vyom Saxena | vs590@snu.edu.in |
| Pranshu Srivastava | ps372@snu.edu.in |
| Nilesh Gupta | ng945@snu.edu.in |
| Niladri Dutta(SPOC) | nd933@snu.edu.in |

**DESIGN:-**

**Description:**

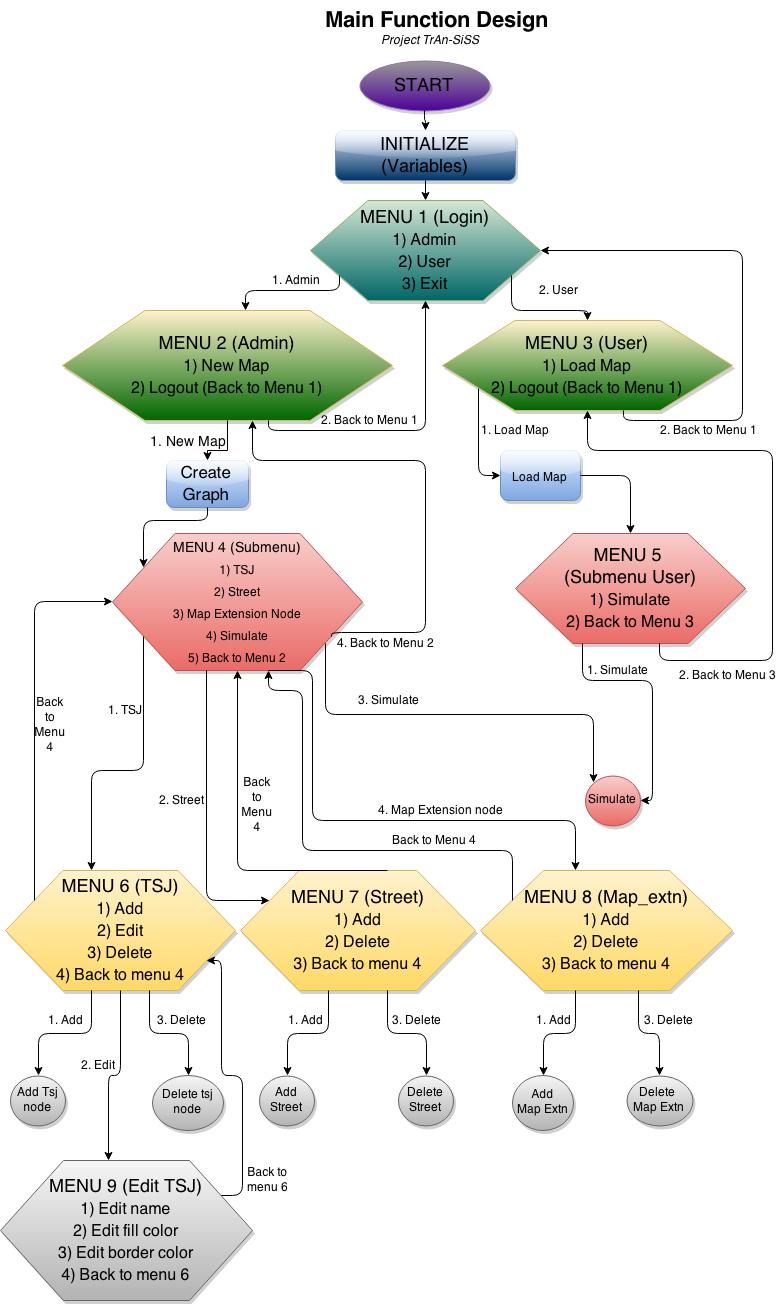
TrAn-SiSS allows to design and simulate vehicle traffic for a given map design. {The user can design a road network in the form of a graph.} The intersection/junctions of the roads will act as the nodes of the graph network. The roads between each junction(node) of the graph will be the edges of the graph. Each road will be implemented as a queue data structure with a specified length(capacity of vehicles). At each node there will be a traffic signal with a traffic light corresponding to each road directed to that junction(node). The traffic light will operate using an analyser which will determine the switching of the traffic signal light.

**User and Use Cases:**

The user in this software:-

**Administrator** : User who can add/modify/delete map designs and also add/delete users. Also the administrator can change the access criteria of each user.

**Standard Users** : These users can simulate the vehicle movement in the map and view the different pattern of vehicle traffic created on the map.



**DESCRIPTION**

The above flow chart depicts the main function design of the project. When the software starts, at first the variables are initialized and a menu is displayed asking to choose among *admin* and *user*.

Admin menu asks the admin to login with his credentials and on successful login, takes him to another menu asking him to choose whether to *Create a new graph* or *load graph* from an existing file. Both the options take the admin to a submenu asking him to choose among *traffic signal junction, street, map extension node* or *simulate* the program. Traffic signal junction menu allows the admin to *add, edit* or *delete* a traffic signal junction.Edit traffic signal junction further allows the admin to *edit name, fill color* or *border color* of a junction.Street menu allows the admin to *add* or *delete* a street while map extension node menu allows him to *add* or *delete* a node.

Now inside user menu, the user is asked to login with his credentials and on successful login, he is allowed to *load graph from a file* and *simulate* the program.

**CODE SKELETON:-**

**Data Types**

# **1) Traffic Signal Junction (traffic\_sig\_jn)**

**Description:-**

Nodes which together form a graph data structure and contain the traffic signal controls at the lane junctions.

**2) Lane (lane)**

**Description:-**

Queue of vehicle (struct vehicle data type) which would contain all the vehicles queues in a lane.

**3) Street (street)**

**Description:-**

Contains the lanes between two Traffic Signal Junction Nodes.

**4) Traffic Signal Light (traffic\_light)**

**Description:-**

Traffic signal lights (Red, Yellow & Green ) control.

# **5) Vehicle(vehicle)**

**Description:-**

Contains data variables describing the vehicle object.

# **6) Route (route)**

**Description:-**

Queue of steps for taking the route from the starting to the end map extension node.

**7) Map Extension Node(map\_extn)**

**Description:-**

Nodes that are an extension to the map boundaries. This object will be static in the program.

**8) Map Extension (map\_extn\_list)**

**Description:-**

Structure to represent each node which is used to extend the graph of TSJs.

**9) Vertex Node (vertex\_node)**

**Description:-**

Structure to represent each node(source node) of the Adjacency List.

**10) Adjacent List (adj\_list)**

**Description:-**

Structure to represent each linked list of the Adjacency List. The structure itself is a linked list.

**11) Graph (graph)**

**Description:-**

Structure to represent each node of the Adjacency List.

**WORK BREAKUP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Number** | **Task Name** | **Description** | **Assigned to** | **Status** |
| 1 | All Possible Traffic Light Combinations | List out All Possible Traffic Light Combinations | Niladri | Done |
| 2 | All Possible Traffic Light Combinations at a Junction(refer Task 2 Info) | List out All Possible Traffic Light Combinations at Junction(2way) | Nilesh & Vyom | Done |
| List out All Possible Traffic Light Combinations at Junction(4way & 6 way) | Nilesh | Done |
| 3 | Make structures (Backend)  (refer to Data Structure Description) | Map Extension Node | Nilesh & Vyom | Done |
| Traffic Signal Junction | Nilesh & Vyom | Done |
| Street | Pranshu | Done |
| Traffic Signal Light | Nilesh & Vyom | Done |
| Vehicle | Nilesh & Vyom | Done |
| Direction | Nilesh & Vyom | Done |
| Lane | Pranshu | Done |
| Graph Data Structure  (Backend)  (refer Graph Representation file) | Adjacent List Node | Pranshu | Done |
| Adjacent List | Pranshu | Done |
| Graph | Pranshu | Done |
| Make Structures  (Graphics - Frontend)  (refer to the Graphics Description file) | Coordinate | Pranshu | Done |
| Style | Pranshu | Done |
| 4 | Make Functions  (Backend) | Main | Pranshu | Done |
| Menu(s) | Pranshu | Done |
| Create TSL | Nilesh | Done |
| Create TSJ | Nilesh | Done |
| Set TSL Mode | Pranshu | Done |
| Set TSJ Mode | Pranshu | Done |
| Create Map Extension | Nilesh | Done |
| Map Extension Operations(Add,Delete,Edit) | Nilesh | Done |
| Set ID | Pranshu | Done |
| Get ID | Pranshu | Done |
| Create Street | Nilesh | Done |
| Create Lane | Nilesh | Done |
| Road Queue( create, delete, insert) | Nilesh | Done |
| Make Functions (Simulator) | Simulate | Niladri | Done |
| Analyze | Nilesh & Vyom | Done |
| Get Directions | Nilesh & Vyom | Done |
| Update Map Extension Nodes | Nilesh | Done |
| Update Vehicles | Nilesh | Done |
| Make Functions  (Graphics-Frontend) | Initialize Coordinates | Niladri | Done |
| Initialize Style | Niladri | Done |
| Set Coordinate | Niladri | Done |
| Set Style | Niladri | Done |
| Draw Lane | Niladri | Done |
| Draw Street | Niladri | Done |
| Draw TSJ | Niladri | Done |
| Draw TSL | Niladri | Done |
| 5 | Compiling the Code - Header Files | Graph | Vyom | Done |
| Map Extension Node | Vyom | Done |
| Traffic Components | Vyom | Done |
| Graphics | Niladri | Done |

**CHALLENGES**

* Conditions in the function “update vehicles()”.
* Integrating the project as a whole project and make it work.
* Implementation of graphics to present the functioning of the program.
* Handling and accessing highly complex data structures while programming.
* Debugging ‘n’ number of errors(logical, syntactical and memory).

**THINGS WE LEARNT FROM THE PROJECT**

* Mutual understanding and working together as a team.
* Designing a project and executing the design in the form of code during the development of a software project.
* Getting familiar with the process by which softwares are developed in the industry.
* Completion of the project in a limited time.