# Software Requirements Specification

## Project Name: Ontology Visualiser

**Team No. : 15**

Chandan Singh(201101194), Piyush Bansal,(201102022), Gv Karthik(201101051)

## Brief Problem Statement

The project aims at developing a visualization tool to deal with graphs. The tool is required to be very flexible so that it can be deployed easily and can be easily changed further to cater the needs the needs of the clients. The idea is to provide user with a platform where he can easily add and delete edges and nodes in a graph. The visualizer should represent the graph in a user-friendly way such that it can be visualized and manipulated easily. It should also give user a way to define certain rules such that the graph is changed according to that rules. Rules may involve to insert or delete a new node in between a cycle or to replace a subgraph with some other subgraph and so on.

## System Requirements

(i) HTML

(ii) CSS

(iii) JavaScript enabled browser (here is a list of few popular ones)

- Netscape Navigator (beginning with version 2.0)
- Microsoft Internet Explorer (beginning with version 3.0)
- Firefox
- Safari
- Opera
- Google Chrome

(iv) List of libraries required under JavaScript:

- JSON
- d3 (data driven documents)

## User Profile

The users are expected to be people from the CEH department from IIIT. It has been clarified by the client that the user using the Ontology Visualizer would have a basic understanding of computers and would have some prior experience with computers. They would have prior experiences dealing with such visualization tools written in some other formats like flash etc. However, to make things simpler and ensure that the end product is usable by anyone who wants to visualize and manipulate graphs, we have decided to make the User Interface as simple and crisp as possible.

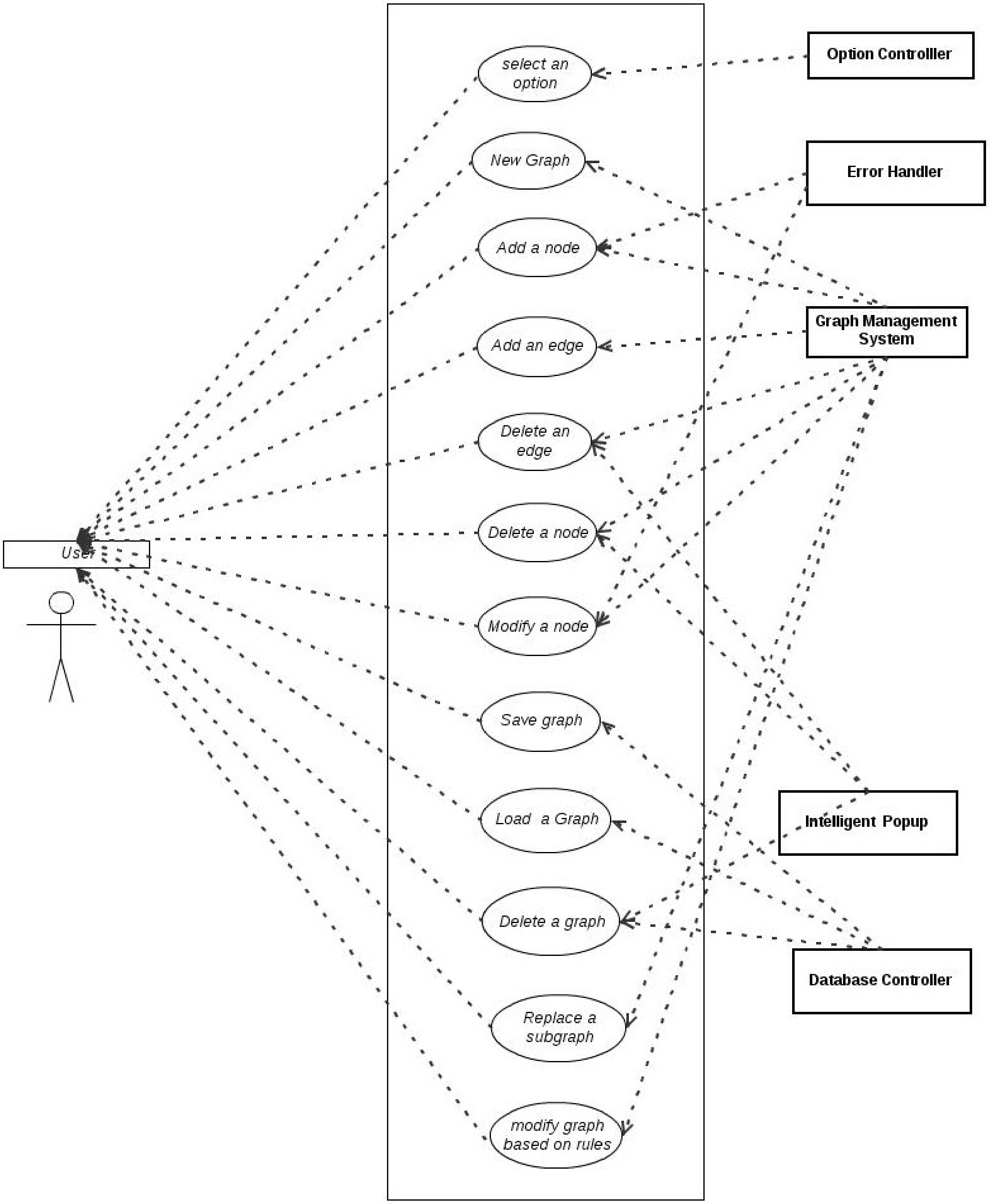
There would be two types of users, one type is using the application as standalone unit and visualizing graphs on local machine. The other type would be to use the application when it is hosted on some web server.

## Feature Requirements

| No. | Use Case Name | Description | Release |
|---|---|---|---|
| 1. | Select an option | The user selects an option from the GUI menu provided. The option may be to add or delete a node, to add or delete an edge or some other option like change colors of nodes etc. | R1 |
| 2. | New Graph | The user is provided with an empty graph where he can perform actions from various options provided. | R1 |
| 3. | Add a node | If user selects to add a node, a new node is created and the user is asked to enter a name for the node. Then the node is added to the graph | R1 |
| 4. | Add an edge | The user is asked to specify the existing two nodes between which he wants to add an edge. If an edge already exists, the user is notified else the edge is added to the graph. | R1 |
| 5. | Delete an edge | The user selects the edge he wants to delete by specifying the nodes between which the edge exists. If the edge does not exists, the user is notified else the edge is removed from the graph. | R1 |
| 6. | Delete a node | The user specifies the node he wants to delete and the node is deleted from the graph and all the edges arising from that node are also deleted. | R1 |
| 7. | Modify a node | The user selects a node and can change its name or colour. The user is notified if the name already exists else the node is renamed. | R1 |
| 8. | Save the graph | The graph is saved in a file which may be reloaded later | R2 |
| 9. | Load a graph | The user is asked to select a previously existing graph from the list of existing graphs. The selected graph is loaded into the current instance. | R2 |

| | | | |
|---|---|---|---|
| 10. | Delete the graph | The user can select to delete the entire graph in one go. A new empty graph is loaded after the selected graph is deleted. | R2 |
| 11. | Replace a subgraph | The user selects a portion of the graph and specifies to replace the selected portion (i.e., one or more nodes with their edges) of the graph with another subgraph. User is asked the name of the new nodes (if any) and checks if they are distinct. If required new edges are made based on previously existing edges. | R2 |
| 12. | Change the graph based on rules | The user specifies some rules based on which the graph is modified. The rules may be of the kind “replace a quadrilateral with triangle” and so on. Internally, the implementation of rules will be combination of adding and deleting nodes and edges. | R2 |

# Use Case Diagram



## Use case description

| | |
|---|---|
| **Use case number** | UC-01 |
| **Use case name** | Select an option |
| **Overview** | The user selects an option from the GUI |
| **Actors** | User, Option Controller |
| **Pre condition** | GUI menu is displayed |
| **Main flow** | 1. User clicks on the desired option |
| **Post Condition** | Action regarding selected option is taken |

| | |
|---|---|
| **Use case number** | UC-02 |
| **Use case name** | New graph |
| **Overview** | A new empty graph is loaded |
| **Actors** | User, Graph management system |
| **Pre condition** | User selects the option to create a new graph |
| **Main flow** | 1. A new empty graph is created<br>2. User is asked to enter the name for this graph |
| **Post Condition** | A new empty graph is created. |

| | |
|---|---|
| **Use case number** | UC-03 |
| **Use case name** | Add a node |
| **Overview** | A new node is added to the graph |
| **Actors** | User, Graph management system, Error Handler |
| **Pre condition** | User selects the option to add a new node |
| **Main flow** | 1. A new node is created<br>2. User is asked to enter the name for this node |

| | |
|---|---|
| | 3. User selects a color for the node |
| **Alternate flow** | If the user enters a name such that a node with that name already exists then an error is generated and user is again asked to enter the name. |
| **Post Condition** | A new node with user-defined name and color is created. |

| | |
|---|---|
| **Use case number** | UC-04 |
| **Use case name** | Add an edge |
| **Overview** | A new edge is added to the graph |
| **Actors** | User, Graph management system |
| **Pre condition** | 1. User selects the option to add an edge<br>2. No. of nodes is greater than 2 |
| **Main flow** | 1. User selects two nodes between which the edge is to be created |
| **Alternate flow** | If the user selects two nodes between which an edge exists a message is reported that an edge already exists. |
| **Post Condition** | An edge is created between the specified vertices. |

| | |
|---|---|
| **Use case number** | UC-05 |
| **Use case name** | Delete an edge |
| **Overview** | An existing edge is deleted |
| **Actors** | User, Graph management system |
| **Pre condition** | 1. User selects the option to delete an edge<br>2. Atleast one edge exixts in the graph |
| **Main flow** | 1. An edge is selected which is to be deleted<br>2. User is asked to confirm the delete option<br>3. User confirms and the edge is deleted |
| **Alternate flow** | If user denies the confirmation no action is taken. |

| | |
|---|---|
| **Post Condition** | The selected edge is deleted from the graph. |

| | |
|---|---|
| **Use case number** | UC-06 |
| **Use case name** | Delete an existing node |
| **Overview** | An existing node is deleted from the graph |
| **Actors** | User, Graph management system |
| **Pre condition** | User selects the option to delete an existing node |
| **Main flow** | 1. User selects the node to be deleted<br>2. User is asked to confirm the delete option<br>3. User confirms and the node is deleted along with its edges |
| **Alternate flow** | If user denies the confirmation no action is taken. |
| **Post Condition** | The selected node and all its edges are deleted from the graph. |

| | |
|---|---|
| **Use case number** | UC-07 |
| **Use case name** | Modify a node |
| **Overview** | Attributes of the selected nodes are modified |
| **Actors** | User, Graph management system, Error Handler |
| **Pre condition** | User selects the option to modify a node |
| **Main flow** | 1. User selects the node to be modified<br>2. User is given option to enter its new name/color<br>3. User enters new name/selects a color for the node |
| **Alternate flow** | If the user enters a name such that a node with that name already exists then an error is generated and user is again asked to enter the name. |
| **Post Condition** | The name/color of existing node is modified. |

| | |
|---|---|
| **Use case number** | UC-08 |
| **Use case name** | Save the graph |

| | |
|---|---|
| **Overview** | Current graph is saved |
| **Actors** | User, Database controller |
| **Pre condition** | 1. User selects the option to save the current graph<br>2. Atleast one node exists |
| **Main flow** | 1. User is asked to enter the name for the graph if the graph is never saved before |
| **Post Condition** | A file is created corresponding to the current graph. |

| | |
|---|---|
| **Use case number** | UC-09 |
| **Use case name** | Load a graph |
| **Overview** | An existing graph is loaded |
| **Actors** | User, Database controller |
| **Pre condition** | 1. User selects the option to load an existing graph<br>2. Atleast one graph exists |
| **Main flow** | 1. User is asked to select a graph from the list of existing graphs |
| **Post Condition** | The specified graph is loaded |

| | |
|---|---|
| **Use case number** | UC-10 |
| **Use case name** | Delete the graph |
| **Overview** | Delete the entire graph(all nodes and edges) |
| **Actors** | User, Database Controller |
| **Pre condition** | User selects the option to delete the graph |
| **Main flow** | 1. User is asked to confirm the delete the option<br>2. User confirms and the current graph is deleted(its database file is deleted) |
| **Alternate flow** | If the user enters a name such that a node with that name already exists then an error is generated and user is again asked to enter the name. |
| **Post Condition** | A new node with user-defined name and color is created. |

| | |
|---|---|
| **Use case number** | UC-11 |
| **Use case name** | Replace a subgraph |
| **Overview** | A subgraph is replaced by another subgraph |
| **Actors** | User, Graph management system |
| **Pre condition** | User selects the option to replace a subgraph |
| **Main flow** | 1. User selects a part of the graph<br>2. User is asked to enter the number of nodes/edges he wants to add/delete<br>3. Use cases for adding/deleting nodes/edges is called |
| **Post Condition** | Selected subgraph is replaced with specified subgraph. |

| | |
|---|---|
| **Use case number** | UC-12 |
| **Use case name** | Change the graph based on rules |
| **Overview** | Change the graph based on some specific user-specified rules |
| **Actors** | User, Graph management system |
| **Pre condition** | User selects the option to change the graph based on rules |
| **Main flow** | 1. User is asked to specify/select a rule<br>2. Specified rule is implemented using use cases for adding/deleting nodes/edges. |
| **Post Condition** | The graph is changed according to specified/selected rules. |