* Data Importer

|  |  |
| --- | --- |
| Functional Requirement | |
| R1.1 | UI can save settings in ImporterSettings |
| R1.1.1 | UI can select SNS which can be fetched in ImporterSettings |
| R1.1.2 | UI can select url/target/user which can be fetched in ImporterSettings |
| R1.1.3 | UI can select items/columns which can be fetched in ImporterSettings |
| R1.2 | Importer can import data from selected SNS |
| R1.2.1 | Importer can import data from Facebook |
| R1.2.2 | Importer can import data from Twitter |
| R1.2.3 | Importer can import data from Youtube |
| R1.2.4 | Importer can import data from Flickr |
| R1.3 | Importer can load data from file(graphML) |
| R1.4 | Core can load settings and execute with corresponding concrete importer. |

GraphStorageManagement

|  |  |
| --- | --- |
| Non-Functional Requirement |  |
| R2.1 | Memory usage should be more efficient than original NodeXL |
| Functional Requirement |  |
| **R2.2** | Table Design |
| **R2.2.1** | Design Table Naming method |
| **R2.2.2** | Design Different Social Media table schema & primary key |
| **R2.2.2.1** | Facebook Schema |
| R2.2.3 | Design Index method |
| R2.3 | Table Function |
| R2.3.1 | List Tables |
| R2.3.2 | Add Table |
| R2.3.3 | Delete Table |
| R2.3.4 | Select Table |
| R.2.3.5 | UI Design |
| R2.4 | Filter rows by derived restriction |
| R2.5 | Save graph structure to DB(new table) |
| R2.6 | Convert selected data to "Graph" structure according to Settings |
| R2.6.1 | UI can select which column of the data to be used as vertices and save them in Settings |
| R2.6.2 | User can Match an (weighted) edge according to relationship between vertices (select edge) and save in Settings |

* Metric

|  |  |
| --- | --- |
| R4.1 | Calculate graph metrics according to Settings and generate result. |
| R4.1.1 | Compute ClosenessCentrality of each vertex in a graph |
| R4.1.2 | Compute BetweennessCentrality of each vertex in a graph |
| R4.1.3 | Compute EigenvectorCentrality of each vertex in a graph |
| R4.1.4 | Compute PageRank of each vertex in a graph |
| R4.1.5 | Compute Max/Avg GeodesicDistances of a graph |
| R4.1.6 | Compute Modularity of a graph |
| R4.1.7 | Generate clusters of a graph by Girvan-Newman algorithm |
| R4.1.8 | Generate communities of a graph by Clauset Newman Moore algorithm |
| R4.1.9 | Identify all Cliques of a graph |
| R4.1.10 | Calculate clusters using the Wakita-Tsurumi algorithm |
| R4.1.11 | Calculate in/out degree of each vertex in a graph |
| R4.1.12 | Identify all strongly connected components of a graph |
| R4.1.13 | Calculate intergroup edges of any pair of group and within a group |
| R4.1.14 | Calculates the reciprocated vertex pair ratio for each of the graph's vertices |
| R4.1.15 | Generate a subgraph according to a specified vertex |
| R4.1.16 | Partition the graph into motifs |
| R4.2 | UI can select some algorithm to calculate graph metrics and save in Settings. |

* Graph Representation

|  |  |
| --- | --- |
| Non-Functional Requirement |  |
|  | Unify the Rectangle to the same type: System.Windows.Rect (WPF) |
| Functional Requirement |  |
| R5.1 | NodeXL has a view to show graph on it |
| R5.1.1 | To show graph on the view, we first layout the graph then draw them on the view |
| R5.2 | When layouting, we specify the location of each vertex in the graph according to the layout context and layout type. |
| R5.2.1 | The types of layouts includes: CircleLayout,GridLayout, Random Layout, SinusoidLayout, etc. |
| R5.2.2 | The layout type is selected by the user |
| R5.2.3 | We can only plot one type of layout for a graph at one time. |
| R5.2.4 | It is easy to add new types of layouts. |
| R5.2.5 | The layout context describe the width,height of the layout view. |
| R5.2.6 | The location of each vertex contains the x,y coordinates |
| R5.3 | When drawing graph, we draw the vertices and edges in the graph on the view according the layout context and the properties of vertices and edges |
| R5.3.1 | The vertices and edge in the graph can be draw with different color, border |
| R5.3.2 | The used color are depend on user's selection |
| R5.4 | The user is able to Export current layout view into image file |

* UI

|  |  |
| --- | --- |
| Functional Requirement |  |
| R6.1 | There are 3 section in the application : 1. Toolbar on the top. and below the toolbar - 2. Table on the left 3. Graph on the right |
| R6.1.1 | User can choose to load data from file, database or SNS importer. |
| R6.1.2 | After the data is loaded, user can choose which column of the data to be vertices and input restrictions to filter rows. |
| R6.1.3 | User can choose to export the data to image file or graphML file (need to specify the file path) |
| R6.1.4 | User can select multiple metric to be computed at once (check box) |
| R6.1.5 | Once the metrics are computed, user can use one of the metrics to autofill columns(color, width, style, opacity, visibility, label) |
| R6.1.10 | There is a dropdown list widget, which includes the layout that user can choose. |
| R6.1.6 | User can select a layout from dropdown list. |
| R6.1.7 | User can refresh the graph |
| R6.1.8 | User can Show/hide table |
| R6.1.9 | User can Show/hide graph |

* IIR

|  |  |
| --- | --- |
| **IIR1** | **Importer read setting info from UI** |
| **IIR2** | **Imorted data is stored to DataBase** |
| **IIR3** | **Graph can ask data from DataBase** |
| **IIR4** | **UI can ask Mertic to compute metrics** |
| **IIR5** | **Layout can get data from Graph to show.** |
| **IIR6** | **Layout can ask for setting info from UI.** |
| **IIR7** | **After computated, Layout can notify UI to render the result.** |
| **IIR8** | **Metric can ask data from Graph** |
| IIR9 | UI can ask Importer to Import data and save in DB. |
| IIR10 | UI can select which Metric to be calculated and save in settings. |

* OIR

|  |  |
| --- | --- |
| OIR1 | Importer use "SocialNetworkImporter" to get data from SNS |
| OIR2 | DataBase use "SQLite" to save and fetch data. |