David Wells

Jeremy Klas

Josh Armstrong

Logan Evans

Maxine Major

Xiaozhe Shen

Zach Curtis

UML Diagrams  
Sunday, October 16, 2011

**Document Manipulation**

|  |  |
| --- | --- |
| *Use case name* | **NewFile** |
| *Participating Actor* | User |
| *Flow of Events* | 1. The user selects new file from menu 2. Program requests user select a diagram type (includes ChooseDiagramType use case) 3. The program responds by creating a new file with a blank drawing canvas |
| *Entry condition* | User selects New File from commands |
| *Exit condition* | Program successfully opens new file |
| *Quality requirements* | TBD |

|  |  |
| --- | --- |
| *Use case name* | **OpenFile** |
| *Participating Actor* | User |
| *Flow of Events* | 1. User selects Open File from menu 2. Program opens an explorer window 3. User selects a file 4. Program loads selected file into program |
| *Entry condition* | User selects open file from menu |
| *Exit condition* | File is successfully opened |
| *Quality requirements* | User selected file must be of program type |

|  |  |
| --- | --- |
| *Use case name* | **SaveFile** |
| *Participating Actor* | User |
| *Flow of Events* | 1. The user selects save file 2. Program saves file    1. If file has not been previously saved, ask user where to save    2. If file has been previously saved, ask user where to save |
| *Entry condition* | User initiates SaveFile |
| *Exit condition* | Program saves file (in xml format) |
| *Quality requirements* |  |

*Note: Save file can also include “export document as…” functions*

|  |  |
| --- | --- |
| *Use case name* | **PrintDiagram** |
| *Participating Actor* | User |
| *Flow of Events* | 1. User initiates Print 2. Program sends to printer |
| *Entry condition* | User initiates Print |
| *Exit condition* | Diagram successfully sent to printer |
| *Quality requirements* |  |

|  |  |
| --- | --- |
| *Use case name* | **CloseProgram** |
| *Participating Actor* | User |
| *Flow of Events* | 1. User clicks X 2. If file is not saved, include SaveFile 3. Program Exits |
| *Entry condition* | User initiates CloseProgram |
| *Exit condition* | Files successfully saved and program exits |
| *Quality requirements* |  |

|  |  |
| --- | --- |
| *Use case name* | **ChooseDiagramType** |
| *Participating Actor* | Included from NewFile use case |
| *Flow of Events* | 1. User selects a diagram type 2. Program loads & displays only that specific diagram’s objects |
| *Entry condition* | Included from NewFile use case |
| *Exit condition* | Included from NewFile use case |
| *Quality requirements* |  |

**Diagram Manipulation**

|  |  |
| --- | --- |
| *Use case name* | **CreateObject** |
| *Participating Actor* | User |
| *Flow of Events* | 1. User selects an object from the list of valid objects 2. User places object on drawing surface |
| *Entry condition* | Toolbar is loaded with valid objects for diagram type |
| *Exit condition* | Object has been successfully placed on drawing canvas |
| *Quality requirements* |  |

|  |  |
| --- | --- |
| *Use case name* | **ConnectObject** |
| *Participating Actor* | User |
| *Flow of Events* | 1. User selects two objects 2. User selects a connector type 3. Program places connector between objects |
| *Entry condition* | At least two valid objects are placed on canvas and selected |
| *Exit condition* | Connector successfully placed between the objects |
| *Quality requirements* | Connectors cannot cross each other Connectors will connect to the center of the side of the object |

|  |  |
| --- | --- |
| *Use case name* | **EditObject** |
| *Participating Actor* | User |
| *Flow of Events* | 1. User edits object. 2. Program extends this case to one of the following use cases:  * Resize * Copy/paste * Edit text * Delete * Move * Edit/link to subdiagram |
| *Entry condition* | Object(s) exist on canvas |
| *Exit condition* | Object edits completed successfully |
| *Quality requirements* |  |