Software Requirements Specification

For

Cloudy Mesh

Prepared By:

Lucky People:

Umut Ağıl

Emre Avan

Gökhan Uras

Muhittin Burak Üçok

Change History

VERSION NUMBER	***DATE***	***NUMBER OF FIGURE, TABLE OR PARAGRAPH***	***ADDED/ MODIFIED/ DELETED***	***TITLE OR BRIEF DESCRIPTION***
Version 1.0	11.11.2012			Original version
Version 1.1	17.01.2013	All the document	Modified	The document layout has been changed to make it more readable.
Version 1.1	17.01.2013	figure 1	Added	Use cases diagram is added.
Version 1.1	17.01.2013	Section 3.23	Added	Neighborhood sellection use case is added.

Table of Contents

1. Introduction

1.1. Purpose

This document details the software requirements specification for mesh editing tool called Cloudy Mesh project. It also follows the IEEE standard for software requirements specification documents.

Computer aided engineering (CAE) is the use of computer software to aid in engineering problems for a wide range of industries. Software tools that have been developed to assist these engineering activities are CAE tools. One of the most important branches of CAE tool is mesh editing tools. However, mesh editing tools can only be installed and used in computers with high computation power and there is no application that allows people to use them on the browser. The purpose of this project is to provide a 3D mesh editing tool which will work on browsers using a cloud computing system.

1.2. Document Conventions

This version of the document was created to show what changes have been made, so some of the requirements stated before, have been altered with respect to changes made in design process. It is very important to update this document with every future requirement and clarify its priority for consistency purposes, so that this document can remain useful.

1.3. Intended Audience and Reading Suggestions

Software Requirements Specification documents is aimed for:

- Developers of the project, who can review and change the capabilities of the project, to easily, decide where they should focus to improve or add more features to it. It also helps them in design and code application parts.
- While testing the project to help testers deciding their testing strategy and test inputs.
- The users who may want to read about the specification of the program.

1.4. Project Scope

Cloudy Mesh will be a mesh editing tool which will work on browsers using a cloud computing system. The GUI (graphic user interface) provides the user options that will allow him/her edit meshes.

In the first version GUI is supposed to contain these plugins:

- Group individual meshes
- Disintegrate groups of meshes
- Import/export meshes
- Scale objects
- Zoom in and out
- Rotate and translate objects
- Select geometric objects from list

1.5. References

IEEE. IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Overall Description

2.1. Product Perspective

Cloudy Mesh is a multi-functional tool for mesh editing. It is free and open source with a GNU General Public License (GPL). Although there are many programs for mesh editing, there is not any program to do this on the cloud system and let user use the program on the browser. Therefore, this program is important to demonstrate the effectiveness of using cloud system on a mesh editing tool.

2.2. Product Features

Basic version of the program provides these:

- Creating meshes for certain objects in 2D and 3D
- Choosing edges with mouse
- Choosing vertices with mouse
- Choosing meshes with mouse
- Grouping meshes and giving specific attributes to groups
- Disintegrating the groups of meshes into individual meshes
- Importing meshes
- Exporting meshes
- Zooming
- Scale objects
- Rotate objects
- Translate objects
- Selecting objects from list
- Create an account
- Login into account
- Logout from account

Further versions of the program may contain these features:

- Creating vertices
- Deleting vertices
- Creating edges
- Deleting edges
- Save mesh
- Load mesh

2.3. User Classes and Characteristics

This project refers to certain type of users of a computer system. Since this project will serve as a research for further development of mesh editing tools on cloud systems, it is expected to have limited user types:

1. Developers who works in EDA Tasarım, because they suggested the idea of making a mesh editing tool working on a cloud system.

2. Engineers whose work requires mesh editing tools can use this program effectively without installing the program to their computers.

2.4. Operating Environment

User side:

- Operating systems :
  - Microsoft Windows
  - Mac OS X
  - Linux
- Web browsers :
  - Any browser that supports HTML5 and WebGL.

Cloudy Mesh will work on any operating system, since it requires only specified web browsers.

- Disk space needed: 250MB

250MB disk space is needed for browser data and for saving mesh object data to user’s computer.

- Graphics Card, having higher or equal attributes with Nvidia GT520M.

NVIDIA GeForce GT 520M is an entry level laptop graphics card.

Cloud Side:

- Operating System:
  - Linux

  - Windows
- Disk space needed: 50GB

This amount of disk space is needed for storing users' mesh data and account information.

- Graphics Card, having higher or equal attributes with Nvidia GT555M

Nvidia GT555M is one of the most powerful mobile graphic card for laptops. Therefore that graphic card is chosen.

2.5. Design and Implementation Constraints

Cloudy Mesh is platform independent and will be written in HTML5, WebGL and python.
Cloudy Mesh is GNU GPL licensed and everyone, that does or is going to develop or use Cloudy Mesh, should agree and fully accept the terms of this kind of license.

2.6. Assumptions and Dependencies

User side of the program must have HTML5 and WebGL compatible browser.

3. System Features

All of the system features are related to the user directly.

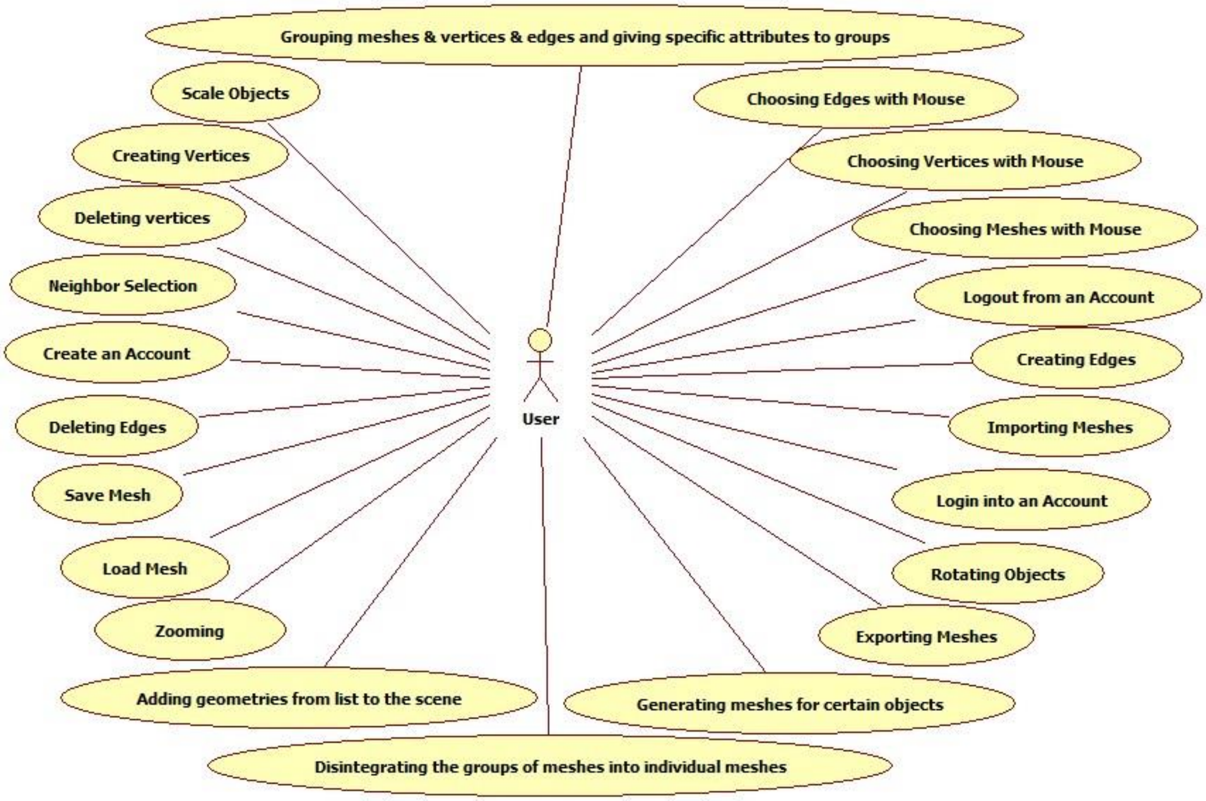
Grouping meshes & vertices & edges and giving specific attributes to groups
Scale Objects
Choosing Edges with Mouse
Creating Vertices
Choosing Vertices with Mouse
Deleting vertices
Choosing Meshes with Mouse
Neighbor Selection
Logout from an Account
Create an Account
Creating Edges
User
Deleting Edges
Importing Meshes
Save Mesh
Login into an Account
Load Mesh
Rotating Objects
Zooming
Exporting Meshes
Adding geometries from list to the scene
Generating meshes for certain objects
Disintegrating the groups of meshes into individual meshes

Figure 1 : Use cases

3.1. Adding geometries from list to the scene

Figure 2: Use Case Diagram 1

3.1.1. Description

The user can choose geometry from the geometry list, then he can add it to the scene.

3.1.2. Stimulus/Response Sequences

First, the user must click the “Geometry List” button. Then a list of geometries will appear. The list contains the following geometries:

- Triangle
- Square

- Cube
- Rectangular prism
- Triangular prism

When the user clicks on one of the geometry names on the list, an object of the chosen geometry will be drawn to the center of the scene.

Objects will have a default size and color. After drawn, they can be modified by users.

3.1.3. Functional Requirements

Req-1) the user can only choose one geometry at a time.

Req-2) when the user clicks on one of the geometry names, the list must disappear.

3.2. Generating meshes for certain objects

user
Generating meshes for certain objects

Figure 3: Use Case Diagram 2

3.2.1. Description

The user can create meshes inside the objects that are displayed on the scene. Then s/he can implement mesh operations on the objects.

3.2.2. Stimulus/Response Sequences

First, the user must choose an object that is displayed on the scene, by clicking on it. After that, when the user clicks the “Generate Mesh” button another list appears. The user decides what the type of the meshes will be. They can be 2D meshes which are triangle or quadrilateral or 3D meshes which are quadrilateral. Then with mesh triangulation algorithms provided by “EDA Tasarım” the chosen object will be divided into meshes. Number of meshes will be default. After mesh generation is done, the object will be drawn with wireframes.

3.2.3. Functional Requirements

Req-1) The user can choose only one object before clicking the “Generate Mesh” button.

3.3. Choosing Edges with Mouse

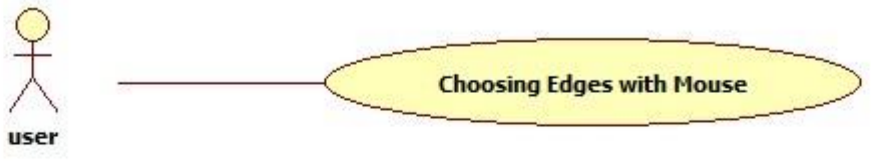

Figure 4: Use Case Diagram 3

3.3.1. Description

The user can choose edges inside the objects that are displayed on the scene. Then s/he can various operations on the objects.

3.3.2. Stimulus /Response Sequences

In order to choose edges, user must select the edges button in the selecting menu in the GUI. If s/he doesn’t select the button, while dragging the mouse both edges, vertices and meshes are chosen.

First, the user chooses the edges by dragging mouse and clicking the left button. The user can release the chosen edges by clicking on them and pushing CTRL button at the same time. After user chose edges with mouse, color of the chosen edges will be altered. User can group the chosen edges by grouping function in the GUI.

3.3.3. Functional Requirements

Req-1) The user can choose edges only when there is an object on the scene and the object must have at least one edge

Req-2) The user can choose edges of a single object.

3.4. Choosing Vertices with Mouse

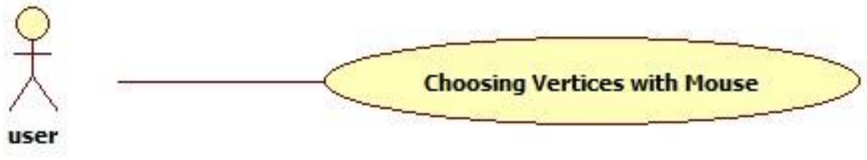

Figure 5: Use Case Diagram 4

3.4.1. Description

The user can choose vertices inside the objects that are displayed on the scene. Then s/he can various operations on the objects.

3.4.2. Stimulus / Response Sequence

In order to choose vertices user must select the “vertices” button in the selecting menu in the GUI. If s/he doesn’t select the button, while dragging the mouse both edges vertices and meshes are chosen.

First, the user choose the vertices by dragging mouse and clicking the left button. The user can release the chosen vertices by clicking on them and pushing CTRL button at the same time.

After user chose vertices with mouse, color of the chosen vertices will be altered. User can group the chosen vertices by grouping function in the GUI.

3.4.3. Functional Requirements

Req-1) The user can choose vertices only when there is an object on the scene.

Req-2) The user can choose vertices of a single object.

3.5. Choosing Meshes with Mouse

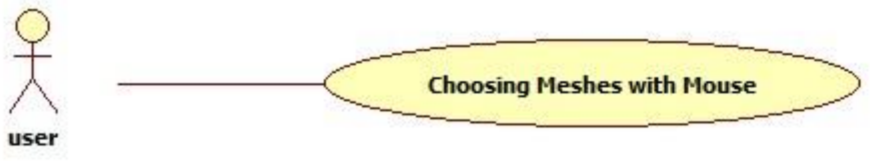

Figure 6: Use Case Diagram 5

3.5.1. Description

The user can choose meshes inside the objects that are displayed on the scene. Then s/he can various operations on the objects.

3.5.2. Stimulus / Response Sequences

In order to choose meshes user must select the mesh button in the selecting menu in the GUI. If s/he doesn’t select the button, while dragging the mouse both edges vertices and meshes are chosen.

First, the user choose the meshes by dragging mouse and clicking the left button. The user can release the chosen mesh by clicking on them and pushing CTRL button at the same time.

After user chose meshes with mouse, color of the chosen vertices will be altered. User can group the chosen meshes by grouping function in the GUI.

3.5.3. Functional Requirements

Req-1) The user can choose meshes only when there is an object on the scene.

Req-2) The user can choose meshes of a single object.

3.6. Importing Meshes

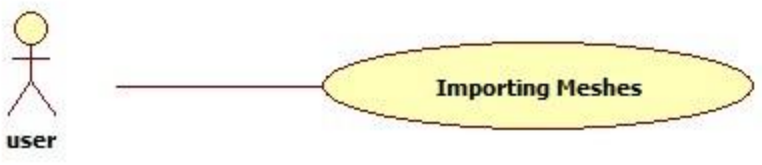

Figure 7: Use Case Diagram 6

3.6.1. Description

User may want to use his/her own mesh data in the program. The user can import his/her own meshes into the program and can visualize them in the scene.

3.6.2. Stimulus / Response Sequences

When the user clicks on the “Import Mesh” button, a file chooser dialog appears. User needs to choose a mesh file to import. Afterwards, the file chooser dialog closes and mesh data in the selected file is imported.

Then the mesh object is drawn and displayed in the center of the scene. Position and rotation of the imported mesh object will be default, but after importing is finished user may change them.

3.6.3. Functional Requirements

Req-1) User can choose only one file in the file chooser dialog.

Req-2) Imported mesh data format must be compatible with the program.

3.7. Exporting Meshes

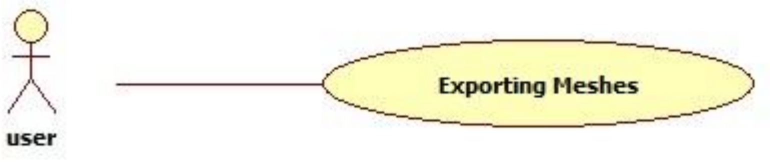

Figure 8: Use Case Diagram 7

3.7.1 Description

After generating or editing meshes user may need to save them. They can still be in progress and user may want to finish them later. Maybe user may want to use them in another application.

3.7.2. Stimulus / Response Sequence

In order to export mesh data to files, user must click on to "Export Mesh" button in the selecting menu of the GUI. After that, a file chooser dialog opens. User chooses an existing file or writes a file name to create a new file to write into. After clicking "open" button, file chooser dialog closes and mesh data is saved to that file.

However, this process applies both on the user computer and the computer on the cloud server. It is saved in the user's account on the database of the cloud.

3.7.3 Functional Requirements

Req-1) Saved mesh data format must be compatible with the import mesh feature.

Req-2) User must have logged into his/her account, if s/he wants to export mesh on the cloud..

3.8. Zooming

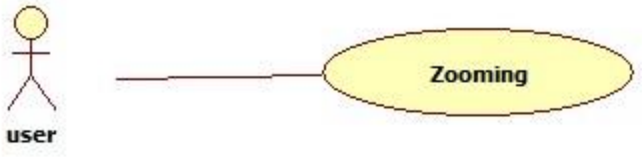

Figure 9: Use Case Diagram 8

3.8.1 Description

User can zoom in to focus specific parts of the object or can zoom out to have general perspective.

3.8.2 Stimulus / Response Sequence

In order to zoom in to meshes user should press w button on keyboard, then user will have closer viewpoint. Besides, in order to zoom out to meshes user should press s button on keyboard,then user will have outer viewpoint.

3.8.3 Functional Requirements

Req-1)User can only zoom in or zoom at the same time.

3.9. Rotating Objects

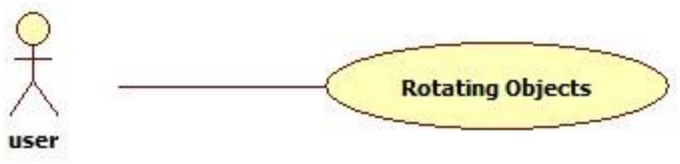

Figure 10: Use Case Diagram 9

3.9.1. Description

After generating or editing meshes user may need to work on different side of the object. To do this user can rotate object.

3.9.2. Stimulus / Response Sequence

In order to rotate an object, user should click left button of the mouse on the object which will be rotated. Then, to rotate object, user should hold and drag the mouse until having desired viewpoint. After that, user should release the left button.

3.9.3. Functional Requirements

Req-1) User can rotate object only when there is at least one object on the scene.

Req-2) User can rotate an object in x, y and z dimensions.

Req-3) User can rotate only one object at a time.

3.10. Grouping meshes & vertices & edges and giving specific attributes to groups

user
Grouping meshes & vertices & edges and giving specific attributes to groups

Figure 11: Use Case Diagram 10

3.10.1. Description

The user can group meshes & vertices & edges that are displayed on the scene. By using grouping, user can work on a complicated scene.

The groups that are created by the user can be seen as list and user can select a group in the list. After s/he chooses, color of the group will altered.

3.10.2. Stimulus / Response Sequence

In order to group meshes user must select meshes, vertices or edges by using the button accordingly.

After s/he chose the parts of object, user clicks the group button in the GUI and a pop-up screen will show up. User will enter a group name in this pop-up screen.

3.10.3 Functional Requirements

Req 1) User can group the parts of a single object.

3.11. Disintegrating the groups of meshes into individual meshes

user
Disintegrating the groups of meshes into individual meshes

Figure 12: Use Case Diagram 11

3.11.1. Description

The user can disband the groups on the group list . This feature will be done automatically if the object is erased.

3.11.2. Stimulus / Response Sequence

First, user clicks the group list button which is next to the group button in the GUI. After s/he clicks the button a pop-up screen will show up and user can search the group that s/he will disintegrate. Then user will right click the group name in the list that will be disintegrated and another pop-up menu will show up. In the second popup menu there is a disintegrate button. If user clicks the disintegrate button and confirm the disbanding operation, the group will be deleted in the group list.

3.11.3. Functional Requirements

Req 1) There should be at least one group in the list in order to operate this function.

3.12. Scale Objects

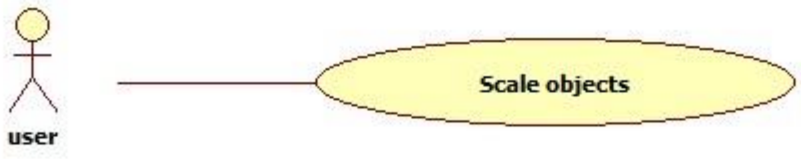

Figure 13: Use Case Diagram 12

3.12.1. Description

The user can scale the object in the scene by using GUI.

3.12.2. Stimulus / Response Sequence

First user clicks the objects that s/he wants to scale. After s/he clicks right side of the screen another menu will show up. In the menu user can see some informations about the objects such as size, positions, number of meshes, vertices and edges. In that menu user can also change the size of the object by using Scale submenu. In the scale submenu there are 3 textboxes that are the scale coefficients of the objects in the x, y and z axis. These coefficients are initially at 1. By changing the coefficients in the textbox, objects size will be altered according to the coefficients.

3.12.3. Functional Requirements

Req 1) User must choose an object in order to apply scale operation.

3.13. Translate Objects

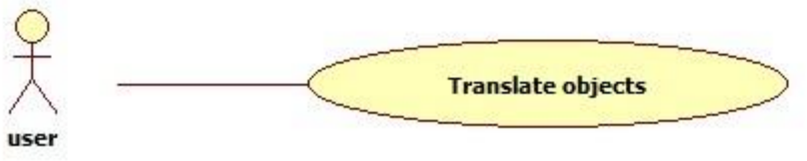

Figure 14: Use Case Diagram 13

3.13.1 Description

The user can translate the object in the scene by using the GUI.

3.13.2. Stimulus / Response Sequence

First user clicks the objects that s/he wants to translate. After s/he clicks right side of the screen another menu will show up. In the menu user can see some informations about the objects such as size, positions, number of meshes, vertices and edges. In that menu user can also change the positions of the center of the object by using Position submenu. In the position submenu there are 3 textboxes that are the coordinates of the objects center. By changing the coordinates in the textbox, objects will translate into new position.

3.13.3. Functional Requirements

Req 1) User must choose an object in order to apply translate operation.

3.14. Creating Vertices

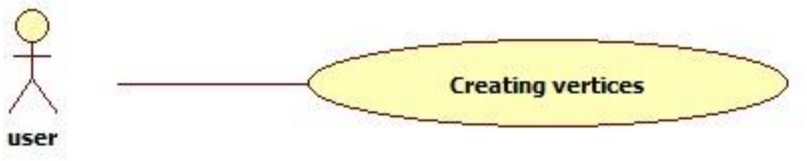

Figure 15: Use Case Diagram 14

3.14.1. Description

User can create vertices and add new vertices to the objects.

3.14.2. Stimulus / Response Sequence

First, user clicks the vertices button at the primitives submenu in the Geometry List. After that, user can add new vertices in the scene by clicking on scene and can combine with an object by using creating edges function.

3.14.3. Functional Requirements

Req 1) None

3.15. Deleting vertices

user
Deleting vertices

Figure 16: Use Case Diagram 15

3.15.1. Description

User can delete vertices in the objects or the vertices that is not bound to an object.

3.15.2. Stimulus / Response Sequence

First, user chooses the vertices that s/he want to delete by using choosing vertices in the object feature. After vertices are chosen, user presses the delete button in order to delete vertices.

3.15.3. Functional Requirements

Req 1) There should be at least 1 vertex in order to deleting function to work.

3.16. Creating Edges

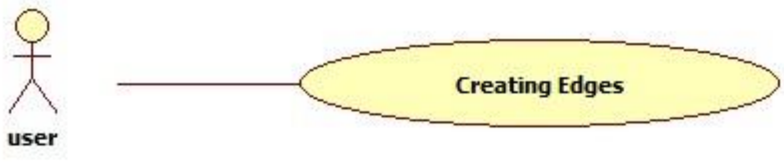

Figure 17: Use Case Diagram 16

3.16.1. Description

User can create edges and add new edges to the objects.

3.16.2. Stimulus / Response Sequence

First, user clicks the edge button at the primitives submenu in the Geometry List. After that, user can add new edges in the scene and can combine with an object by using a vertice of the object as a side of the edge.

3.16.3. Functional Requirements.

Req 1) None.

3.17. Deleting Edges

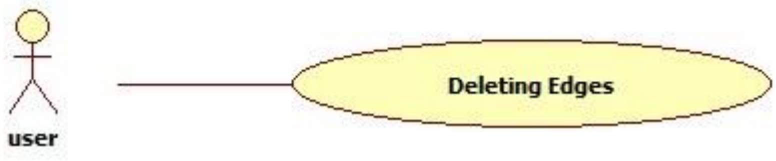

Figure 18: Use Case Diagram 17

3.17.1. Description

User can delete edges in the objects or the edges that is not bound to an object.

3.17.2. Stimulus / Response Sequence

First, user chooses the edges that s/he want to delete by using choosing edges in the object feature. After edges are chosen, user presses the delete button in order to delete edges.

3.17.3. Functional Requirements

Req 1) There should be at least 1 edge in order to deleting function to work.

3.18. Save Mesh

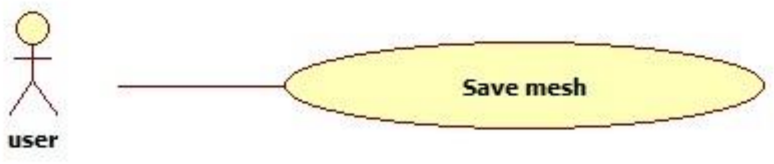

Figure 19: Use Case Diagram 18

3.18.1. Description

User can save the meshes that s/he are working on. Data is stored in the cloud with the client information, therefore there will not be any security issues since data cannot be reach by other clients

3.18.2. Stimulus / Response Sequence

In order to save meshes, user presses the save button in the GUI. After s/he presses the button, a text box for mesh name appears. After user enters the name of mesh, the mesh is saved into the cloud server.

3.18.3. Functional Requirements

Req. 1) User must have logged into his/her account.

3.19. Load Mesh

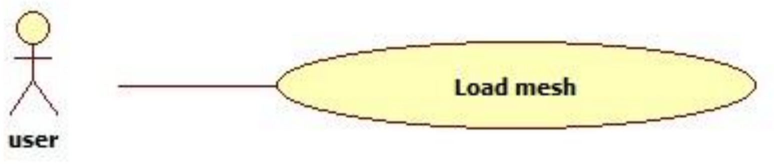

Figure 20: Use Case Diagram 19

3.19.1. Description

User can load the meshes that s/he has saved before. Data is stored in the cloud with the client information, therefore there won't be any security issues since data cannot be reach by other clients

3.19.2. Stimulus / Response Sequence

In order to load meshes, user presses the load button in the GUI. After s/he presses the button, a list of saved meshes appears. After user selects what s/he wants to load, the mesh is loaded from the cloud server.

3.19.3. Functional Requirements

Req. 1) User must have logged into his/her account.

3.20. Create an Account

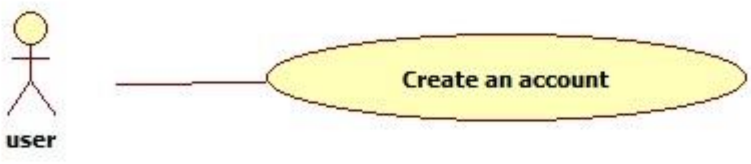

Figure 21: Use Case Diagram 20

3.20.1. Description

User can create an account to be able to use features like exporting meshes, saving and loading meshes on the cloud.

3.20.2. Stimulus / Response Sequence

In order to create an account, user clicks register the button in the GUI. After s/he presses the button, a new page with a registration form opens. After user enters the required information like username and password and clicks create account, the account is created.

3.20.3. Functional Requirements

Req. 1) None

3.21. Login into an Account

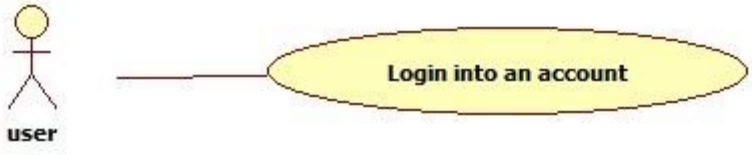

Figure 22: Use Case Diagram 21

3.21.1. Description

User can login into an account to use the features like exporting meshes, saving and loading meshes on the cloud.

3.21.2. Stimulus / Response Sequence

In order to login into an account, user clicks the login button in the GUI. After s/he presses the button, a popup that contains 2 textboxes for username and password appears. After user enters the required information which are username and password and clicks login, user logs into his/her account.

3.21.3. Functional Requirements

Req. 1) User must be created an account.

3.22. Logout from an Account

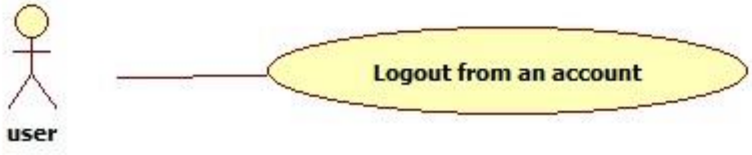

Figure 23: Use Case Diagram 22

3.22.1. Description

The user may logout to protect his/her account from unwanted access.

3.22.2 Stimulus / Response Sequence

In order to logout from account, user clicks the logout button in the GUI. After s/he presses the button, user logs out from his/her account.

3.22.3. Functional Requirements

Req. 1) User must have logged into his/her account.

3.23. Neighborhood selection

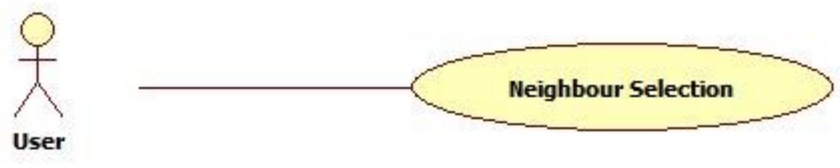

Figure 24: Use Case Diagram 23

3.23.1. Description

The user may select neighbors of the selected vertex, edge or polygon. Depth of neighborhood is decided by the user then the function finds nearest neighbors and makes them selected.

3.23.2. Stimulus / Response Sequence

First, user must select an edge, vertex or a polygon then clicks the “Find Neighbor” button and determines number of neighbors. After that given depth of neighborhood edges, vertices or polygons are added to selected section.

3.23.3. Functional Requirements

Req. 1) User must have logged into his/her account.
Req. 2) User must have loaded a mesh to the scene.
Req. 3) User must have selected an edge, vertex or a polygon.

4. External Interface Requirements

4.1. User Interfaces - GUI

Cloudy Mesh GUI will be designed as an user friendly interface which will allow user editing meshes by supporting following plugins:

- Grouping individual meshes
- List of groups
- Disintegrating groups of meshes
- Import/export meshes

- Scaling meshes
- Zooming
- Rotating and translating objects
- Selecting meshes of the objects
- Selecting vertices of the objects
- Selecting edges of the objects
- Selecting geometric objects from list

4.2. Hardware Interfaces

Hardware interfaces subtopic is not applicable to Cloudy Mesh project.

4.3. Software Interfaces

Because Cloudy Mesh works on web browser; it is compatible with every system supports a web browser which supports HTML5 and WebGL, regardless of operating system of the computer system on which it runs.

It will be tested on versions of Microsoft Windows, GNU/Linux distributions and Mac OS X and also versions of web browsers: Chromium, Mozilla Firefox, Safari and Opera.

4.4. Communications Interfaces

The network connection is a must because the Cloudy Mesh is a web application which will use the connection to communicate between user and cloud server. The connection will be established from web browser.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

Cloudy mesh is an application that needs a few system resources to work. This program is designed not to delay the system or reduce the system performance. Only the visual parts of the program are computing by GPU of the client computer and all the other parts of the system are computed by the cloud. The response time of the program is depend on the internet connection and the workload of the cloud server.

Program is stored in the cloud server. Therefore user does not need to check for updates, system will update itself automatically at a given time such as 03.00 AM every day in order to not blocking users work. Cloudy Mesh will be always up-to-date with all new supported features and bug fixes.

5.2. Safety Requirements

Cloudy mesh must ensure that, saved data shouldn’t be corrupted. No modification is allowed to the saved data without permission of the owner of the data.

Moreover, the application should continue to run even in cases of wrong data insertion or wrong settings. Since there could be multiple users that are using cloud server, system must respond all the users without blocking another.

5.3. Security Requirements

Cloudy mesh must ensure that, private data of the user can’t be seen by other users without permission. This obligation is ensured by using user account system with proper authentication and authorization methods.

5.4. Software Quality Attributes

Cloudy Mesh must provide a user friendly and pleasant graphical user interface that are mentioned in the above feature. Any user that used any CFD or CAD program should be able to use Cloudy Mesh. Program also contains the user manuals and help messages will help users to use Cloudy Mesh easily.

The project should be developed as a modular, component based system to improve maintainability and future development.

5.5. Other Requirements – GNU GPL License

The project is released under the GNU General Public License. The philosophy of this license implies some basic principles which apply to the project.

The GPL is a free software license, and therefore it permits people to use and even redistribute the software without being required to pay anyone a fee for doing so.

− The freedom to run the program, for any purpose.

− The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.

− The freedom to redistribute copies so you can help your neighbor.

− The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

6. Appendix

6.1. Appendix A: Glossary

GUI: Graphical User Interface

CFD: Computational Fluid Dynamics

CAD: Computer Aided Design

Mesh: Collection of vertices, edges and faces that defines the shape of object.

Vertex: Point, corner point of a polygon.

Edge: Line segments that joining of two vertices.

GNU GPL: Free software license.

WebGL: Web graphics library.

HTML5: HyperText Markup Language

Python: General purpose programming language.

Cloud Computing: Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the internet).

CAE: Computer aided design, computer software to aid in engineering tasks.