|  |
| --- |
| HD in Cloud Computing |
| Requirements Specification (RS) |
| CADCloud |

|  |
| --- |
| Leszek Dubicki x14125439  7/11/2015 |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 12/07/2015 | 1 | Created and submitted for grading | Leszek Dubicki |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Arghir Moldovan | Lecturer |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| CADCloud Use Case Diagram |  |

**Table of Contents**

[Requirements Specification (RS) 1](#_Toc424500934)

[Document Control 1](#_Toc424500935)

[Revision History 1](#_Toc424500936)

[Distribution List 1](#_Toc424500937)

[Related Documents 1](#_Toc424500938)

[1 Introduction 4](#_Toc424500939)

[1.1 Purpose 4](#_Toc424500940)

[1.2 Project Scope 4](#_Toc424500941)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc424500942)

[2 User Requirements Definition 6](#_Toc424500943)

[3 Requirements Specification 6](#_Toc424500944)

[3.1 Functional requirements 7](#_Toc424500945)

[3.1.1 Use Case Diagram 7](#_Toc424500946)

[3.1.2 Requirement 1 <Register User> 7](#_Toc424500947)

[3.1.3 Requirement 2 <Send an Atomic Data to Server> 9](#_Toc424500948)

[3.1.4 Requirement 3 <Get an Atomic Data from Server> 10](#_Toc424500949)

[3.1.5 Requirement 4 <Send Sketch to Server> 12](#_Toc424500950)

[3.1.6 Requirement 5 <Get Sketch from Server> 13](#_Toc424500951)

[3.1.7 Requirement 6 <Manage Users> 14](#_Toc424500952)

[3.2 Non-Functional Requirements 15](#_Toc424500953)

[3.2.1 Performance/Response time requirement 15](#_Toc424500954)

[3.2.2 Availability requirement 15](#_Toc424500955)

[3.2.3 Recover requirement 16](#_Toc424500956)

[3.2.4 Security requirement 16](#_Toc424500957)

[3.2.5 Reliability requirement 16](#_Toc424500958)

[3.2.6 Portability requirement 16](#_Toc424500959)

[3.2.7 Extendibility requirement 16](#_Toc424500960)

[4 GUI 16](#_Toc424500961)

[5 System Architecture 17](#_Toc424500962)

[6 System Evolution 17](#_Toc424500963)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of a cloud based information sharing service for cad and engineering applications.

The intended customers are engineering companies, preferably companies working with many smaller but complex projects involving cooperation between multiple departments.

## Project Scope

The scope of the project is to develop a cloud based information sharing service for cad and engineering applications. The system shall have a CADCloud server that would provide or retrieve data from either CAD application through embedded feature or from custom engineering application (e.g. Shaft Calculation application which will be done as a part of this project).

The project will include special embedded CAD features/plugins (for FreeCAD and optionally for Solidworks) interacting with cloud based application calculating shaft loads and shaft deformations based on information from connected CAD client. The application will also be able to send data to clients requesting them.

Data that will be shared include text and numerical values (to be stored as global variables or custom properties in CAD files) and basic 2D geometry (points, lines, arcs and circles) stored as sketches or curves in CAD files. This data will be either synchronized with connected cloud service or will be inserted only on user request.

## Definitions, Acronyms, and Abbreviations

CAD – Computer Aided Design the use of special computer systems in mechanical design

CAD Software – application used by designer to design mechanical devices / products. For the purpose of this project open source FreeCAD will be used

Scripted Object or Python Feature – Special FreeCAD object that allows to include python code in CAD files for creating custom geometry or add custom logic to a file.

CC – acronym for CADCloud

CAPTCHA – standard challenge – response test to verify if the user is a human or not (if not it means it is a computer program accessing the site)

# User Requirements Definition

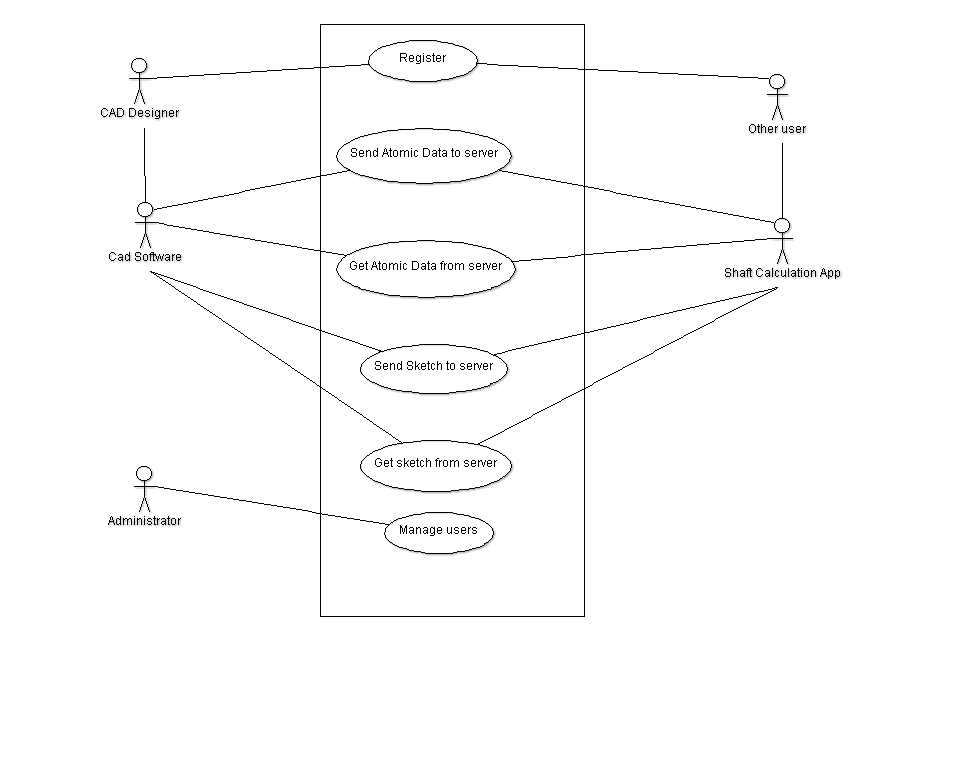
The system must release CAD designer from importing static data from external sources and provide tool to share data online between different systems.

# Requirements Specification

The basic requirement for the system is to be easy to use and transparent for the designer. It means that custom CC feature would be inserted into the tree view like any other objects. This CC feature would be easily configurable (server instance and login details must be provided, than available resources of the server must be presented for the user)

## Functional requirements

### Use Case Diagram



Picture CADCloud Use Case Diagram

### Requirement 1 <Register User>

#### Description & Priority

Every user accessing the CC server must be registered to prevent data from unauthorized access. In enterprise environment the login information should be imported/verified on server hosting user profiles (like Windows Server Domain). However, for stand-alone CC server instances (for example set by some hobbyist community) user registration functionality must be provided

#### Use Case

**Scope**

The scope of this use case is to register new user (create user account)

**Description**

This use case describes the user registration process

**Flow Description**

**Precondition**

The user has access to browser connected to internet

**Activation**

This use case starts when an user presses Sign Up button on a web page.

**Main flow**

1. The User enters Nickname of player
2. The System checks if nickname is available
3. The User enters First Name
4. The User enters Second Name
5. The User enters email address
6. The System provides CAPTCHA to user
7. The User enters CAPTCHA
8. The User accepts terms and conditions
9. The User presses Submit Button
10. The System creates user record in user’s database
11. The System sends confirmation / activation email on provided email address
12. The System receives confirmation through provided link

**Alternate flow**

A1: Nickname exists

1. inform user that nickname exists

A2: Account not confirmed for more than 30 days

1. send another confirmation request on provided email

A3: Account not confirmed for more than 60 days

1. send information about deleting user account
2. delete user record

A4: CAPTCHA code not valid

1. re-create CAPTCHA code
2. after third non-successful try 5 minutes delay (to prevent attacks)

A5: Email address already exists in database

1. inform user that email exists
2. ask user if they want to retrieve their password.
3. generate one-time use password
4. send email with generated one-time password

A6: Not all mandatory fields are entered (nickname / email / name / terms and conditions)

1. inform user about missing information

A7: Provided email address doesn’t exist

1. user that email address is not valid

**Exceptional flow**

None

**Termination**

The system marks user account as confirmed

**Post condition**

User Account created and stored in a users database

### Requirement 2 <Send an Atomic Data to Server>

#### Description & Priority

CAD feature embedded in CAD file is set to send an atomic data (ie String or Number) to the server. This operation will be performed either every time the file is rebuilt or manually by calling appropriate method (e.g. right click and send).

#### Use Case

**Scope**

The scope of this use case is to send string or number data (together with variable name, e.g. “a” : 10) to the server.

**Description**

This use case describes sending atomic data to the server

**Flow Description**

**Precondition**

The Scripted Object is put into the CAD File (using macro) and configured (username, password, project id and server address is set, as well as data to be sent to the server)

**Activation**

This use case starts when a user calls method to send data to the server (e.g. from right-click context menu) or the part is recomputed

**Main flow**

1. The CAD feature checks availability of the server API (http HEAD method)
2. The System confirms availability
3. The CAD Feature gets the proper data from the file (e.g. dimension value)
4. The number of tries variable is set to zero
5. The CAD Feature sends the data to the server (HTTP POST method)
6. The System sends back the status (transaction confirmed)

**Alternate flow**

A1 : Service not available

1. The System doesn’t respond within given timeout (set in CAD Feature)
2. The CAD Feature informs the User that the service is offline

A2 : The System didn’t confirm the transaction

1. The number of tries variable is incremented
2. The CAD Feature checks if number of tries variable is not too high (to prevent infinite loop of tries)
3. If the number of tries is below maximum the use case continues at position 1 of the main flow
4. If the number of tries is greater or equal to the maximum value The CAD Feature informs the User that the service is offline

**Exceptional flow**

None

**Termination**

The System stores the data in a database for given project ID

**Post condition**

The system goes into a wait state

### Requirement 3 <Get an Atomic Data from Server>

#### Description & Priority

CAD feature embedded in CAD file is set to get an atomic data (ie String or Number) to the server. This operation will be performed either every time the file is rebuilt or manually by calling appropriate method (e.g. right click and send).

#### Use Case

**Scope**

The scope of this use case is to retrieve string or number data (together with variable name, e.g. “a” : 10) to the server.

**Description**

This use case describes getting atomic data from the server

**Flow Description**

**Precondition**

The Scripted Object is put into the CAD File (using macro) and configured (username, password, project id and server address is set, as well as data to be sent to the server)

**Activation**

This use case starts when a user calls method to get data from the server (e.g. from right-click context menu) or the part is recomputed

**Main flow**

1. The CAD feature checks availability of the server API (http HEAD method)
2. The System confirms availability
3. The number of tries variable is set to zero
4. The CAD Feature gets the data from the server (HTTP GET method)

**Alternate flow**

A1 : Service not available

1. The System doesn’t respond within given timeout (set in CAD Feature)
2. The CAD Feature informs the User that the service is offline

A2 : The System didn’t confirm the transaction

1. The number of tries variable is incremented
2. The CAD Feature checks if number of tries variable is not too high (to prevent infinite loop of tries)
3. If the number of tries is below maximum the use case continues at position 1 of the main flow
4. If the number of tries is greater or equal to the maximum value The CAD Feature informs the User that the service is offline

**Exceptional flow**

None

**Termination**

The CAD Feature stores the data in proper form in the file (file custom property, it’s own custom property or sets some other feature dimension or property – will be defined in CAD Feature configuration or by the type of CAD Feature)

**Post condition**

The system goes into a wait state

### Requirement 4 <Send Sketch to Server>

#### Description & Priority

CAD feature embedded in CAD file is set to send a Sketch geometry to the server. This operation will be performed either every time the file is rebuilt or manually by calling appropriate method (e.g. right click and send).

#### Use Case

**Scope**

The scope of this use case is to send a sketch geometry (together with sketch name) to the server.

**Description**

This use case describes sending sketch geometry to the server

**Flow Description**

**Precondition**

The Scripted Object is put into the CAD File (using macro) and configured (username, password, project id and server address is set, as well as sketch name to be sent to the server)

**Activation**

This use case starts when a user calls method to send data to the server (e.g. from right-click context menu) or the part is recomputed

**Main flow**

1. The CAD feature checks availability of the server API (http HEAD method)
2. The System confirms availability
3. The CAD Feature collects the proper data from the file (all sketch geometry objects)
4. The number of tries variable is set to zero
5. The CAD Feature sends the data to the server (HTTP POST method)
6. The System sends back the status (transaction confirmed)

**Alternate flow**

A1 : Service not available

1. The System doesn’t respond within given timeout (set in CAD Feature)
2. The CAD Feature informs the User that the service is offline

A2 : The System didn’t confirm the transaction

1. The number of tries variable is incremented
2. The CAD Feature checks if number of tries variable is not too high (to prevent infinite loop of tries)
3. If the number of tries is below maximum the use case continues at position 1 of the main flow
4. If the number of tries is greater or equal to the maximum value The CAD Feature informs the User that the service is offline

**Exceptional flow**

None

**Termination**

The System stores the sketch data in a database for given project ID

**Post condition**

The system goes into a wait state

### Requirement 5 <Get Sketch from Server>

#### Description & Priority

CAD feature embedded in CAD file is set to get a Sketch geometry from the server. This operation will be performed either every time the file is rebuilt or manually by calling appropriate method (e.g. right click and send).

#### Use Case

**Scope**

The scope of this use case is to get a sketch geometry (together with sketch name) from the server.

**Description**

This use case describes getting sketch geometry from the server

**Flow Description**

**Precondition**

The Scripted Object is put into the CAD File (using macro) and configured (username, password, project id and server address is set, as well as sketch name to be retrieved from the server)

**Activation**

This use case starts when a user calls method to send data to the server (e.g. from right-click context menu) or the part is recomputed

**Main flow**

1. The CAD feature checks availability of the server API (http HEAD method)
2. The System confirms availability
3. The number of tries variable is set to zero
4. The CAD Feature gets the sketch data from the server (HTTP GET method)

**Alternate flow**

A1 : Service not available

1. The System doesn’t respond within given timeout (set in CAD Feature)
2. The CAD Feature informs the User that the service is offline

A2 : The System didn’t confirm the transaction

1. The number of tries variable is incremented
2. The CAD Feature checks if number of tries variable is not too high (to prevent infinite loop of tries)
3. If the number of tries is below maximum the use case continues at position 1 of the main flow
4. If the number of tries is greater or equal to the maximum value The CAD Feature informs the User that the service is offline

**Exceptional flow**

None

**Termination**

The CAD Feature writes all the sketch data into the proper sketch.

**Post condition**

The system goes into a wait state

### Requirement 6 <Manage Users>

#### Description & Priority

Users should be given permissions to use resources. This permissions will be set by System Administrator.

#### Use Case

**Scope**

The scope of this use case is to add or remove users from list of users allowed to use CC resources under given project ID.

**Description**

This use case describes adding or removing user from list of users allowed to get/set data related to particular project.

**Flow Description**

**Precondition**

Administrator has access to browser connected to internet

The user account is created

**Activation**

This use case starts when Administrator accesses the Administrator Panel

**Main flow**

1. The Administrator selects user they want to manage
2. The Administrator adds/removes user from the list of users allowed to use particular resource

**Alternate flow**

1. The Administrator deletes user permanently from the users database

**Exceptional flow**

None

**Termination**

User information in database is changed

**Post condition**

The system goes into a wait state

## Non-Functional Requirements

### Performance/Response time requirement

Because the System is designed to work with very fast and efficient CAD systems and due to the fact that parts/assemblies are rebiult/recalculated many times during the work day (practically every minute) the response time for the System must respond instantly. However, the embedded features can be set to synchronize manually, in this case the System response is not that critical

### Availability requirement

The aim is that the system will be available to users for both standard users (Designers / web apps) and Administrators 24/7

### Recover requirement

Generally not needed since data is taken from other source (CAD or Web Application)

### Security requirement

Due to possibility of sharing sensitive or confidential technical data proper level of encryption must be used

### Reliability requirement

The System will be used by many users at a time so failures must be kept to an absolute minimum

### Portability requirement

It is required to use the System from different CAD systems (like Solidworks, CATIA, Creo Parametric and many more), but each of this systems would require its own plug-in to be developed

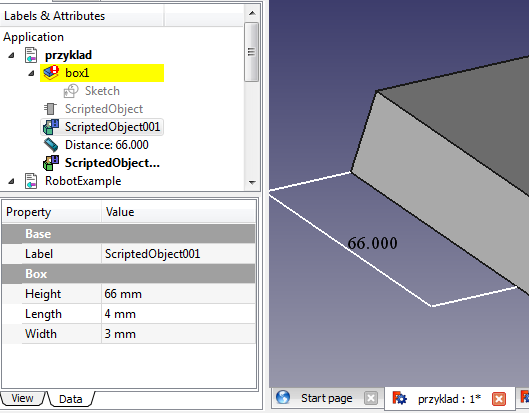
### Extendibility requirement

In case of serving big amounts of data to multiple companies distributed architecture must be taken into consideration.

# GUI

In general the System will be a RESTful API that would be used by CAD applications and by other Web Applications created for particular tasks. The only GUI of the System itself will be registration page (although it will be possible to register using some command line tools ad well, similar to Yowsup-cli WhattsAp client ) and user management page for the Administrator.

The GUI in CAD embedded features will be dependent on CAD Software used, in FreeCAD it will be standard feature configuration panel like seen in picture Picture 1



Picture Future Configuration Example in FreeCAD

# System Architecture

Use a class diagram to outline the structure of the system. Explain briefly why you have chosen this architecture. You might want to use Visio or Rational Rose to create these.

# System Evolution

The presented project has a big potential to become more complex CAD data sharing system. An example is sharing 3D geometry between applications, or even a logic (transfer code that would be executed in CAD environment).

Typical use of a system would be building custom web API that would provide services used directly inside CAD application, which would be the whole new technology in mechanical engineering software.

The system could be very useful in companies which develop big amount of projects, each of them in isolation from each other – it would improve data coordination between Sales, Design, Manufacturing Quality departments.

The system might be very useful in big-project environment with very big amount of data that needs to be shared among different design teams (e.g. R&D team working on product and very often releasing new revision and production equipment team that needs to analyze changes and adapt their models/drawings for every new revision of product).