Analysis and Design Document

Student: Bulea Ioana - Florina

**Group: 931**

# Requirements Analysis

## Assignment Specification

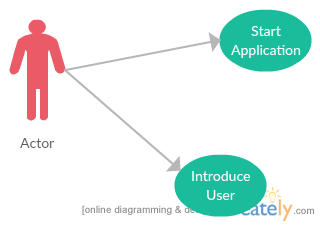
Assignment 3

We would like to design and implement an online whiteboard to be used as a means of connecting remote teams within our company. This application has already been broken down in the following modules.

Approval Module

Accounts can have multiple roles – architect, technical lead, developer. Once a design for a project is considered acceptable any user can initiate a review process. The review process consists of the following workflow: first technical leads are asked to review the design. If they are unable to decide then architects are asked to review the design. Any further changes to the whiteboard marks it as not reviewed and the process is resumed.

# Use-Case Model



**Use-Case description**:

Use case: Introduce User

Level: sub-function

Primary actor: Client

Main success scenario: The Client is presented with the option to introduce his own user into the list of users of the application. The Client has to type “yes” to do that, and then he is presented with some options to introduce data. The introduced user is then added to the user list of users of the application

# Design

The application is a Windows Console application, created using the C# programming language.

The application solves the Approval Module part of the Assignment. The app is launched in the Program.cs class, where it creates the list of users and it initiates the execution of the approval functionality for several designs, each with different levels of difficulty. In the class Utils.cs we have start implementing the Chain of Command design pattern. For each of the users in the list, the next in the chain of command is the subsequent user.

Then, for each ones, we the app tries to get them to approve a certain design, with a certain difficulty. If the experience of the user, added to a random number (which I call readiness), is greater than the difficulty of the design, the user can approve or deny the design. If the previous condition is not satisfied, the app moves down the chain of command.

## Class Diagram



## 3.2 Patterns

The chain of responsibility pattern creates a chain of receiver objects for a request. This pattern decouples sender and receiver of a request based on type of request. This pattern comes under behavioral patterns.

In this pattern, normally each receiver contains reference to another receiver. If one object cannot handle the request then it passes the same to the next receiver and so on.