**Fall**

2013

08

**Fall**

Submitted To:   
Dr. Christine Laurendeau  
3004 Object-Oriented Software Engineering  
School of Computer Science  
Carleton University

TAEval

Requirements Analysis Document

Ben Fairclough, Will Fairclough, Sean Malcolmson

**Team: Not\_A\_Virus**

Table of Contents

4 Introduction 3

4.1 Purpose of System 3

4.2 Overview of Document 3

5 Purposed System 3

5.1 System Overview 3

5.2 Functional Requirements 3

5.3 Non-Functional Requirements 4

6 System Models 5

6.1 Use Case Models 5

6.1.1 High-Level Use Cases 5

6.1.2 Detailed Use Cases 6

6.1.3 Use Case Flow of Events 6

6.2 Dynamic Model 7

6.2.1 Data Dictionary 7

6.2.2 Class Diagram 7

6.3 Object Model 7

6.3.1 Sequence Diagrams 7

6.3.2 State Machine Diagrams 7

7 Acronyms 7

8 Glossary 7

Tables

Table 1 - Functional Requirements 3

Table 2 - Non-Functional Requirements 4

Diagrams

Figure 1 - User Inheritance Use Case Diagram 5

Figure 2 - System Login Control Use Case Diagram 6

# Introduction

Teaching Assistants (TAs) provide a crucial role for courses at a university level. Teaching Assistants complete tasks assigned to them by their instructor to help in successfully delivering course content to the classes students. The tasks a TA may be assigned tasks involve marking assignments and tests, holding office hours, attending meetings and lectures, preparation time and much more. There is currently a lack of tools provided to instructors for managing their Teaching Assistants and the duties that must be performed by them.

## Purpose of System

The purpose of the TAEval system is to provide the instructors with a tool that will allow for easy managing of their assigned teaching assistants.

The TAEval system features will include:

* The ability to create, edit and delete tasks for a TA;
* The ability to enter evaluation data for a specific task assigned to a TA;
* The ability to run reports on TA evaluation data;
* The ability to manage essential system data, such as course offerings, instructors and TAs.

The system will support three types of users:

* Administrators
* Teaching Assistants
* Instructors

Each of these users will provide a different role around managing teaching assistants for courses. Administrators will manage the creation of users and course offerings, as well as running evaluation reports that will describe the success of the teaching assistants. Instructors will be responsible for creating tasks for each of their assigned teaching assistants, as well as evaluating them once they have completed the tasks. Teaching Assistants will then be able to view their assigned tasks and mark them as complete once they have finished. Teaching Assistants can also view their evaluations provided by the instructors of the course.

## Overview of Document

The purpose of this document is to describe in detail the business requirements of the TAEval system as outlined by the client, Dr. Christine Laurendeau in her system description she provided us. The Requirements Analysis Document provides a concise understanding of the proposed TAEval system as understood by the system developers on team *Not\_A\_Virus*, and the client.

This document contains the following sections:

* **Section 4 Introduction:** Provides an introduction to the application domain.
* **Section 4.1 Purpose of System:** Overview of the system and features of TAEval.
* **Section 4.2 Overview of Document:** Brief description of each of the sections.
* **Section 5 Purposed System:** Describes the features of the system in more detail.
* **Section 5.1 System Overview:** Brief description of the structure of the system.
* **Section 5.2 Functional Requirements:** Lists the features and interactions of the system.
* **Section 5.3 Non-Functional Requirements:** Lists the constraints the TAEval system must impose.
* **Section 6 System Models:** Provides details UML diagrams and tables describing the features and interactions with the TAEval System
* **Section 6.1 Use Case Models:** Describes the possible ways all the features can be used within the TAEval system.
* **Section 6.2 Object Model:** Describe system models with their attributes along with their associations to each other with UML class diagrams.
* **Section 6.3 Dynamic Model:** Describe the system behaviors and how all the objects communicate between each other using UML sequence and state machine diagrams.
* **Section 7 Acronyms:** A list of all the important Acronyms used within this document.
* **Section 8 Glossary:** A list of all the application domain specific terms used within this document.

# Purposed System

## System Overview

## Functional Requirements

The functional requirements of the TAEval application describe all the features and interactions that make up the system without describing their implementation. Table 1 shows the entire list of function requirements for the TAEval system.

Table - Functional Requirements

|  |  |
| --- | --- |
| ID | Description |
| FR-01 | User must be able to login/logout of the system. |
| FR-02 | The system must be able to determine the user type at login. |
| FR-03 | The system must be able to determine if the user is already logged in at another terminal and log them out. |
| FR-04 | Instructor users must be able to create new tasks. |
| FR-05 | Instructor users must be able to edit existing tasks that are in an uncompleted state. |
| FR-06 | Instructor users must be able to view a list of all their assigned courses. |
| FR-07 | Instructor users must be able to view a list of all the tasks they created for each of their assigned courses. |
| FR-08 | Instructor users must be able to delete tasks. |
| FR-09 | Instructor users must be able to assign tasks to a TA assigned to one of their courses. |
| FR-10 | Instructor users must be able to evaluate tasks in a completed state with a (1-5) value rating and feedback comments. |
| FR-11 | Instructors must be able to view their assigned courses. |
| FR-12 | Instructors must be able to view a list of TA users assigned to each of their assigned courses. |
| FR-13 | TA users must be able to view each of their assigned courses. |
| FR-14 | TA users must be able to view each of assigned tasks for each of their assigned courses. |
| FR-15 | TA users must be able to mark their assigned tasks as complete. |
| FR-16 | TA users must be able to view the evaluations for each of their completed tasks. |
| FR-17 | Administrator users must be able to create new courses specifying the courses Instructor. |
| FR-18 | Administrator users must be able to view and edit course details. |
| FR-19 | Administrator users must be able to delete courses. |
| FR-20 | Administrator users must be able to create new TA users. |
| FR-21 | Administrator users must be able to view and edit TA user details. |
| FR-22 | Administrator users must be able to delete TA users. |
| FR-23 | Administrator users must be able to assign TA users to specific courses. |
| FR-24 | Administrator users must be able to create new Instructor users. |
| FR-25 | Administrator users must be able to view and edit Instructor user details. |
| FR-26 | Administrator users must be able to delete Instructor users. |
| FR-27 | Administrator users must be able to view a list of Courses, Instructor users, and TA users. |
| FR-28 | Administrator users must be able to run new Evaluation reports. |
| FR-29 | Reports must include TAs average evaluation in each class in each term. |
| FR-30 | Reports must include ratings for all TA users for one term |
| FR-31 | Reports must include ratings for all TA users for one course offering |

## Non-Functional Requirements

The Non-Functional Requirements describe aspects of the TAEval system that are not directly related to the functional behavior of the TAEval system. In contrast to the functional requirements instead of describing what the TAEval system will be doing the Non-Functional requirements listed in Table 2 will describe how the system will be implemented to do those features.

Table - Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Category | Description |
| NFR-01 | Usability | System shall use colour themes based on schools they are used in. (Use a themes file) |
| NFR-02 | Usability | System shall use common UI elements on all views to enhance user experience |
| NFR-03 | Usability | System shall show tooltips on element hover for more help information |
| NFR-04 | Usability | System shall provide a "What's This?" Button on complicated views. |
| NFR-05 | Usability | System shall provide links to the help document on every appropriate view page |
| NFR-06 | Usability | System shall provide descriptive Error messages in failing states and provide the user with an appropriate solution |
| NFR-07 | Reliability | System shall try and reconnect 3 times before showing a Network Timeout Error. |
| NFR-08 | Reliability | Users shall be forced back to the login screen during Network Timeout Errors |
| NFR-09 | Reliability | User passwords shall be encrypted before storing them in the database |
| NFR-10 | Reliability | TA/Instructors accounts shall be locked out if failing to enter correct username and password 10 times |
| NFR-11 | Reliability | System shall provide the user with instructions for creating a recovery file of the servers database |
| NFR-12 | Reliability | System shall use session identifiers to improve security and identify client sessions. |
| NFR-13 | Performance | Client shall only request data from the server that it requires. |
| NFR-14 | Performance | System server shall respond to client requests within 2 seconds |
| NFR-15 | Performance | System server shall handle a minimum of 4 simultaneous client connections |
| NFR-16 | Implementation | System shall run on the 32-bit Ubuntu Linux VM |
| NFR-17 | Implementation | System shall be written in C++ |
| NFR-18 | Implementation | System shall use the Qt 4.8 C++ framework |
| NFR-19 | Implementation | System server shall store data in SQLite database |
| NFR-20 | Implementation | System shall use TCP sockets to communicate between the server and the client. |
| NFR-21 | Supportability | System shall use the Qt framework to make it easily portable to other operating system platforms. |
| NFR-22 | Supportability | Client shall only be able to connect to a server of equal version number |
| NFR-23 | Supportability | System shall provide a help document for how to use the system |
| NFR-24 | Supportability | System shall write log files for both client and server to support user issues |
| NFR-25 | Supportability | System server shall be given instruction to create database from recovery file |
| NFR-26 | Supportability | System server shall be given instruction to create database recovery file |
| NFR-27 | Packaging | Application shall be delivered to client on CD |
| NFR-28 | Packaging | Application shall have clear instruction on how to install to system |

# System Models

## Use Case Models

### High-Level Use Cases

The *User* actor depicted in Figure 1 represents an abstract actor that TeachingAssistance, Instructor, and Administrator actors all inherit from. In the use cases if the term *User* is used it is referring to all or any of the three *User* subclasses. Where as if a specific *User* subclasses name is used than only that *User* subclass has the represented features. As well it should be understood that all features associated with the abstract *User* actor are also available to the three *User* subclasses.

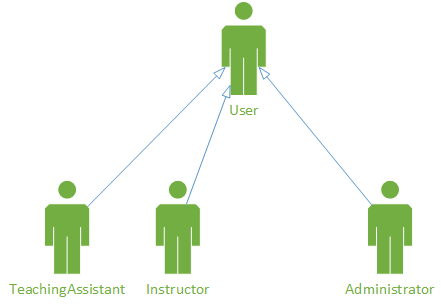


Figure - User Inheritance Use Case Diagram

The entry point for all *Users* is the system login screen, by functional requirement FR-01. All other use cases will depend on the *User* being logged in so it can be assumed that if a *User* is not logged in than no other features will be accessible. Figure 2 depicts the *System Login* and the two system errors that can occur during this case.

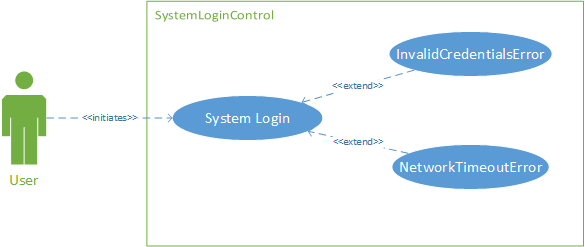


Figure - System Login Control Use Case Diagram

### Detailed Use Cases

### Use Case Flow of Events

## Object Model

### Sequence Diagrams

### State Machine Diagrams

## Dynamic Model

### Data Dictionary

### Class Diagram

# Acronyms

# Glossary