TAEval  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Requirements Analysis Document

**Team Romero’s Severed Head**  
Sean Benjamin  
Dylan Kristolaitis  
Justin Kung

Steven Wu

Submitted to:

Dr. Christine Laurendeau

COMP3004 Object-Oriented Software Engineering  
School of Computer Science  
Carleton University

Contents

1. Introduction………………………………………………………#
   1. Purpose of System………………………………………#
   2. Overview of Document………………………………..#
2. Proposed System……………………………………………….#

2.1. Overview…………………………………………………….#

2.2. Functional Requirements……………………………..#

2.3. Non-functional Requirements………………………#

2.4. System Models……………………………………………..#

2.4.1. Use Case Models…………………………………#

2.4.2. Object Model……………………………………...#

2.4.3. Dynamic Model………………………………….#

1. Glossary…………………………………………………………….#

Figures

tbd

Tables

tbd

1. Introduction

<Sean’s intro here, to have content appended by Steven & edited by Steven>  
1.1 Purpose of System  
1.2 Overview of Document

1. Proposed System
   1. Overview

In this section we outline the technical details of our proposed system, TAEval, by clearly defining functional requirements, non-functional requirements, and outlining unambiguous and complete system models.  
  
TAEval is a client-server application that is designed to optimize the line of communication between an instructor and his or her teaching assistants by automating the issuing and tracking of tasks, task evaluations, and metrics that can quantify the TA’s body of work.

* 1. Functional Requirements

Functional requirements are the concise, explicit details of what the system will be able to do with respect to functionality. For TAEval, for example, there is a distinct difference between the system allowing   
  
Table 1 – Functional Requirements

|  |  |
| --- | --- |
| Traceability Codes | Functional Requirements |
| FR-00 | TAs must be able to view their assigned tasks assigned by the course instructor. |
| FR-01 | TAs must be able to view their tasks’ respective evaluation evaluated by the course instructor. |
| FR-02 | TAs must only be assigned to a maximum of one course at any given time. |
| FR-03 | Instructors must be able to create a task at the beginning of the term for each TA for each class they are instructing. |
| FR-04 | Instructors must be able to edit their existing delegated tasks. |
| FR-05 | Instructors must be able to delete their existing delegated tasks. |
| FR-06 | Instructors must be able to enter evaluation data for each existing delegated task. The evaluation scheme is 1-> ‘poor’, 2-> ‘fair’, 3-> ’good’, 4-> ’very good’, 5-> ’excellent’ |
| FR-07 | Instructors must be able to view a list of courses they are teaching in a specific term. |
| FR-08 | Instructors must be able to view the list of TAs that are assigned to a specific course they are instructing. |
| FR-09 | A course must have an existing instructor associated with it upon its creation. |
| FR-10 | Administrators must be able to run reports on TA evaluation data, such as: TA evaluation ratings for one TA spanning all terms, TA evaluation ratings for all TAs spanning one term, TA evaluation ratings for all TAs for a particular course offering |
| FR-11 | Administrators must be able to view a list of courses offered in a given term. |
| FR-12 | Administrators must be able to view a complete list of all instructors. |
| FR-13 | Administrators must be able to view a complete list of all TAs. |
| FR-14 | Administrators must be able to add course offerings. |
| FR-15 | Administrators must be able to edit course offerings. |
| FR-16 | Administrators must be able to delete course offerings. |
| FR-17 | Administrators must be able to add instructors. |
| FR-18 | Administrators must be able to edit instructors. |
| FR-19 | Administrators must be able to delete instructors. |
| FR-20 | Administrators must be able to add TAs. |
| FR-21 | Administrators must be able to edit TAs. |
| FR-22 | Administrators must be able to delete TAs. |
| FR-23 | Administrators must be able to assign existing TAs to any existing course at any time. |

* 1. Non-functional Requirements
  2. System Models  
     2.4.1 Use Case Model  
     2.4.2 Object Model

2.4.3 Dynamic Model

1. Glossary