**Software Requirements and Design Document**

**For**

**Group <6>**

Version 1.0

**Authors**:

Jay Daniels

Marlan McInnes-Taylor

Keaun Moughari

Daniel Riley

Hayden Rogers

# Overview

The Programming Contest Suite will be used for the semesterly FSU ACM Programming Contest. The system will consist of several components used for different phases of the contest progress. The Suite will allow participants to register for the contest, and form teams with other participants. The participant will also be able to select the computer science department courses in which he or she is currently enrolled, in order to potentially receive extra credit for contest participation.

All necessary information will be automatically exported from the registration database to the DomJudge contest judging database. The Suite will also allow FSUCard sign in on the day of the contest. Once the contest concludes, the results of the contest will automatically be emailed to the computer science department faculty. The faculty member will be able to see the which students in his or her class(es) participated, and how many questions were correctly answered.

# Functional Requirements

* Read mag stripe, and parse input to put in database.
* Generate extra credit lists, send them to appropriate faculty.
* Handle pre-registration, and same day registration
* communicate information to DomJudge

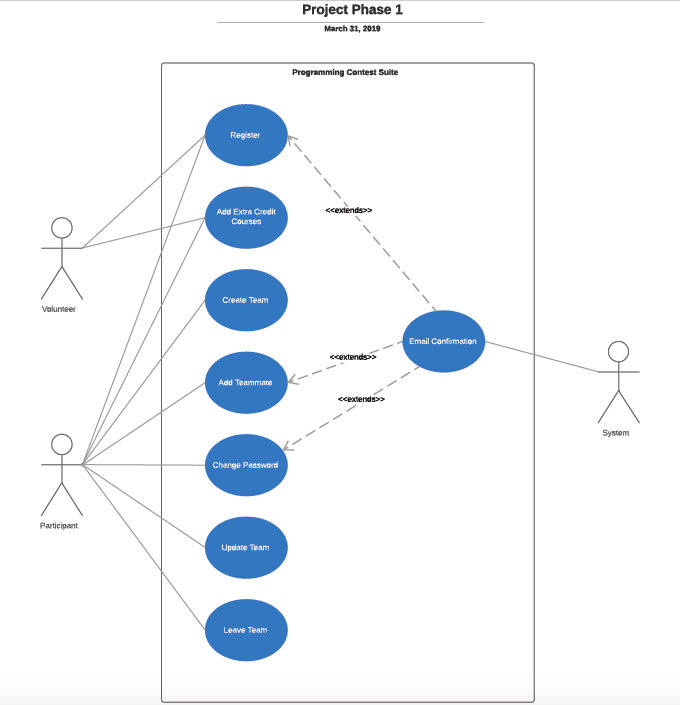
*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

# Non-functional Requirements

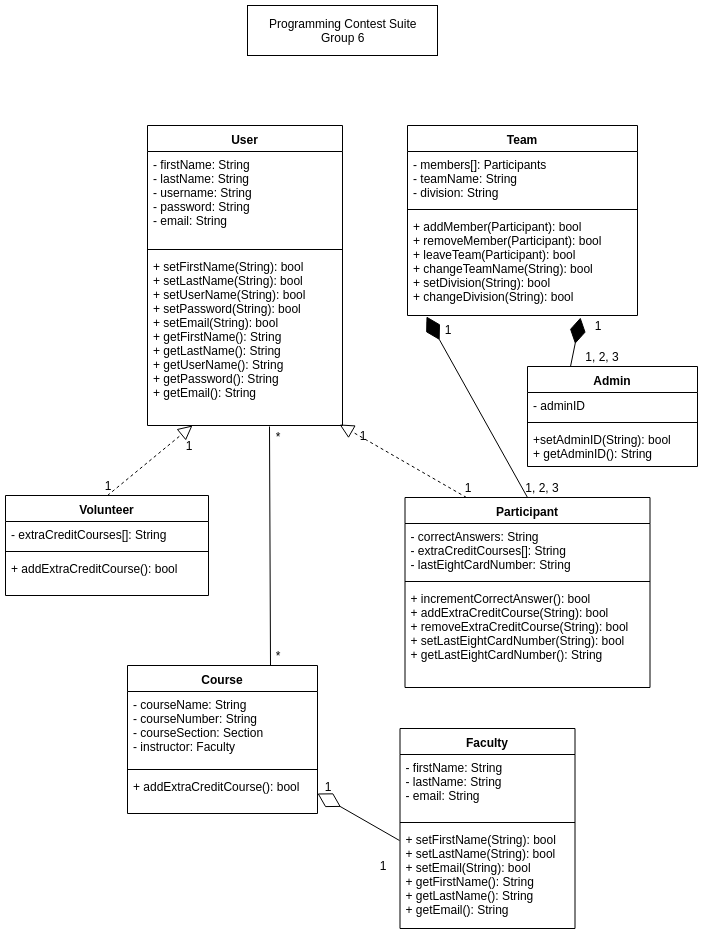
* Store user information securely
* Reliably register participants for contest with few to no errors
* Construct source code in such a way to promote easy maintenance in the long term
* Must function quickly

*List the* ***non-functional requirements*** *of the system (any requirement related to security, safety, software quality, performance, reliability, etc.) You may provide a brief design rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

# Use Case Diagram



# Class Diagram



# Operating Environment

Hardware:

2x AMD Opteron 2216

6GB RAM

OS:

Ubuntu 64 bit

Kernal 4.4.0

# Assumptions and Dependencies

* DOCKER FUNCTIONALITY
* CARD SWIPE INTEGRATION
* Django has reliable and pertinent functionality to facilitate a relatively quick assembly of app.

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*