**STUDENT COURSE REGISTRATION SYSTEM**

**REQUIREMENTS ANALYSIS DOCUMENT**

**Purpose**

Our initial goal for the Student Course Registration System is to produce a simulation of student course registration for our department (CSE). It will be modeled our department's rules and procedures for enrolling in courses. We are going to build our system as a simulation that takes arguments from a user-supplied command-line input file. The simulation software continues until all of the student transcript documents are generated.

**Scope**

The initial version of the Student Course Registration System will be a simulation. A dataset is going to sent to this system as a JSON file. There are requirements for the courses, and other business logic will be based on the rules and procedures of our department for enrolling in classes. In this round, courses will be offered at random, and students will be chosen at randomly. The system will produce output as a Json file after handling each instance.

**Functional Requirements**

* Costumers can view students’ transcript document screen. In this screeen there are grades, courses which are already taken, GPA, CGPA, given credits, completed credits, offered courses and errors during the registration.
* User profile can create students again.

**Nonfunctional Requirements**

* This program written by Java to handle object oriented software programming more effectively.
* The capacity is: Totally 280 students (70 students for each year.)
* Whenever the user wants to use it, they can access and view the system. It’s always available.
* It is access to student information and available courses. The student can quickly finish the registration procedure.

**Planning of the Iteration Sequence**

We will have a student registration system that can operate both step-by-step and as a simulation as a consequence of this project. In each of the eight semesters, we will have an equal number of students. There will be requirements when registering for such students, such as finishing the necessary courses and staying within the course quota. Furthermore, students must also earn a certain amount of credits in order to enroll in final year courses. The user interface will be alerted of the activities made after the registration procedure is complete.

**GLOSSARY**

**Collabrators**

Customer: MURAT CAN GANİZ

Developers:

* Tarık Işın
* Yağmur Koçoğlu
* Ayşenur Tüfekçi
* Ali Batuhan Yamanarslan
* Abdullah Enes Dizer
* Oğuzalp Acıkan
* Mervan Aburşu
* Mert Efe Karaköse
* Muhammet Görkem Gülmemiş

**Using Case – Registration to Course**

**Actors: Advisors and Students**

1- Students enters to the system.

2- The system presents the list of the courses which students can choose and take.

3- Students check over listed courses.

4- Students pick courses from list.

5- Students forward the courses that she/he selected to the their advisor’s approval.

6- The system displays the result of the advisor’s aprrove of the student who selected courses.

**Alternative Case: Selection Error**

4a. In step 4, the system could show error for selection overlapping courses. The system gives permission to the student to re-select courses in the list.

4b. System warns the error because o allocation(quota) problem for selected courses.

**Using Case – Advisor’s Approval**

**Actors: Advisors and System**

1- Advisor logins to the system.

2- System shows the student’s selection course request to the advisor.

3- Advisor browses the courses selected by the student.

4- Advisor starts to the process off approving course selection request.

5- Advisor approves the courses.

**Alternative Case: Disapprove the selections**

5a. At step 5, the advisor do not approve selections because of overlapped courses.

5b. The advisor do not approve graduation project because student completed credits lower than 165.



