

Requirement Analysis Document (RAD)

Software Engineers

Ahmet Hakan Şimşek

Metehan Kurucu

Erdem Açıldı

Arda Yiğit Kaynar

Oğuzhan Karabudak

Mevlüt Eren Topal

Sena Dilara Yangöz

Elif Gülay

Customers

Murat Can GANİZ

Lokman ALTIN

1. Introduction
 - 1.1. Purpose
 - 1.2. Description
 - 1.3. Glossary
2. Requirements
 - 2.1. Functional Requirements
 - 2.2. Non-Functional Requirements
3. Domain Model

INTRODUCTION

This document details the project plan for the development of registration system of Computer Engineering at Marmara University.

It is intended for designers, developers, and testers working on registration as well as project investors. This plan will include a summary of:

- How the system will function
- Added features since last iteration
- The scope of the project from the development viewpoint
- The technology used to develop the project, and
- The metrics used to determine the project's progress
- Overall Description

1.1 Purpose

Our project is a student registering system. Registering is the process of assigning students to register courses that are eligible for student's current situations like prerequisite, credit etc. via a user interface by students. Registering course are determined by entered index of course and if system let to register then registration is done and written into transcript file of specific student.

1.2 Description

Program will let students to register courses by hand in certain conditions.

First of all this program read lists of teachers and create object for each teacher, after reads list of lectures creates lecture object for each lecture according to its type non-technical elective course, technical elective course or regular course. After it reads list of students, creates object for each student assign them to id.

After reading all Json files, it starts to organize transcript randomly for each of students by considering prerequisite and credit. Each student has success rate by this rate they are successful or fail from courses. A student can take maximum 10 lectures in a semester. If they are successful, their completed credit incremented by credit of passed course.

After filling transcript student can select courses that are eligible thanks to console manually. Student can register non-technical elective course if quota is not full, student also can register technical elective course if quota is not full, credit is enough to take, and can take maximum 2 or 3 courses by semester. Student also can register graduation project if they have 165 credits completed.

After registration process is done transcript Json file is written for each student.

1.3 Glossary

Technical elective course : An elective lecture that can be taken if credit completed is more than 155 and cannot taken in semester 2 or 3 and schedule is not overlap

Non-technical elective course : An elective lecture that can taken if quota is not full, no credit completed is required and schedule is not overlap

Regular course : A lecture that can be taken if schedule is not overlap.

Student : A person of registration system to take lectures appropriately.

Teacher : A person of registration system to be an adviser of student

Prerequisite : A course can be taken if all prerequisite lectures are passed successfully.

Transcript : Belongs to each student which show the taken lectures history of student.

Id : A unique number which is assigned for each student by registration system to distinguish persons from each other.

Entropy : Calculation of the randomness and irregularity of the labeling process.

REQUIREMENTS

2.1 Functional Requirements

- **Input** : List of courses, teachers and students which are appropriately designed.
- **Registration** :Users of system can take lecture by registering via this registration system.
- **Output** : Writes taken lectures to json file in transcript format.

2.2 Non-Functional Requirements

- **Random** : At this point of project, the program assigns lectures to students to fill transcript by randomly.
- **LinkedList** : Data such as instances, registered lectures, user are held in linked lists.
- **.json Handling** : json.simple and jackson libraries are used to handle .json files.
- **Student ID** : Users will need a user ID to register lectures.

Domain Model

