1. **Introduction**
   1. **Purpose**

To fix the problems with course management, we want to develop an extensive online course registration and management system that complies with Marmara University rules.

* 1. **Document Conventions**

Conventions have not been used in this document.,

* 1. **Intended Audience and Reading Suggestions**

This document was created to give our client a thorough introduction to the features of the system we constructed. The product's specifications and structure are briefly summarized in the second section. The third chapter details the software's user interface, and the final portion discusses the external requirements.

* 1. **Product Scope**

By adhering to the guidelines outlined in the Marmara University rule, this project intends to overcome the difficulties advisers and students faced when choosing and managing courses.  This system can do tasks including course selection, managing received courses, requesting advisor approval for selected courses, managing enrolled students, and generating student transcripts.

* 1. **References**

<https://bys.marmara.edu.tr>

<http://dosya.marmara.edu.tr/eng/cse/documents/general/CSE_Prerequisite_Tree_2018.pdf>

<http://cse.eng.marmara.edu.tr/dosya/eng/cse/documents/general/PrerequisiteTree_2020.pdf>

1. **Overall Description** 
   * 1. **Product Perspective**

This software project is a stand-alone project which can work independently and is self-contained. It does not contain any graphical user interface (only command line interface). It is created as an alternate version of other course management systems.

* + 1. **Product Functions**

This system’s purpose is to allow students to register to courses under some restrictions which is controlled by the system itself. After the student selects a course, the system controls the restrictions for that student to register to that particular course and decides to approve or reject the registeration process. A transcript is created for each student that can be accessed by them.

* + 1. **User Classes and Characteristics**

All students and advisors can use this software but, students does not have access permission to all parts of it.

Students: They can enroll to courses in new semesters and access their transcripts.

Advisors: They can control their students' course enrolling period. They can approve or disapprove students’ schedule for the semester.

* + 1. **Operating Environment**

Course management system is a software product produced in Java programming language. It runs on the ide console and can be used on many platforms such as: Intellij, NetBeans, Eclipse, Visual Studio Code. The operating environments for development and product use is as follows:

* Development environment of the product: Java
* Database: consists of JSON files that contain data of students and courses.
* Operating Systems: Windows and Linux
  + 1. **Design and Implementation Constraints**
* The software should be implemented in Java programming language, sufficient experience and knowledge in this language is required for the development.
* An object oriented (OOSD) approach should always be priority during both designing and implementing phases of the course management system.
* The developer should be able to create JSON files, add data to them, extract data from them, and delete data from them.
  + 1. **User Documentation**

A detailed user manual should be delivered with the software. The user manual should contain all kinds of details, but not unnecessary information. This document hugely important for users seeking to understand products and processes so it should be written in such a way that even a novice user can understand both the content and usage of the program.

The developer document, on the other hand, is in a format that more programmers can understand. They will be prepared to access information they can add.

* + 1. **Assumptions and Dependencies**

At this stage, no third-party applications will be used at this stage, so there will be no dependencies. Code sharing and communication in the project will be via github. In this case, we are dependent on github. Finally, our project is not dependent on software components of any similar project.

1. **SYSTEM FEATURES**

**3.1 Access to source code**

The user can make changes at each stage and improve each stage.

**3.1.1 Description and Priority**

The code can be customized according to the user's request. Every part can be changed and improved using ide or a code editor.This feature gives the user full authority, so this situation can lead to some dangers.The user may unintentionally cause very large errors and may not be able to solve the problem, so they should be very careful.Although the user does not make any changes, the software will work smoothly.

**3.1.2 Stimulus/Response Sequences**

Since we have given the user full authority, any changes made by the user will be reflected in the execution of the program.

**3.1.3 Functional Requirements**

The number 1 requirement for users is a computer with a Java program, they can run the software with its help.A mouse is optional but a keyboard is a mandatory need.

**3.2 Access to student and course features**

**3.2.1 Description and Priority**

Just like the source code, JSON files which contain properties of students and courses are available to change. Users can change anything about any student's information, they can be created, changed, or deleted. This feature is also vulnerable to any bad intentions. Users can see and observe everything but the system could get corrupted at any moment.

**3.2.2 Stimulus/Response Sequences**

Every change made on a JSON file will be visible on other JSON files.

**3.2.3 Functional Requirements**

Operating systems like Windows, Mac or Linux already has simple text editors that can edit JSON files.