Requirement Analysis Design

1. Description

Aim of this project is to basically design a university course registration system. Student registration system is a function in which students may enroll their courses virtually and courses are controlled by supervisors.

2. Developers

- Furkan Bozkurt 150119767
- Emre Cihan Varlı 150119711
- Gökhan Taşkın 150119681
- Gülsüm Ece Günay 150121539
- Dilara Nur Menteş 150119911
- Dilan Dilen 150119912
- Ayşe Sena Aydemir 150119735
- Reyyan Rana İnan 150119723
- Fatih Erkam Dinler 150119567

3. Glossary

- **app:** Database of the system.
- **FileHandler:** Read data from database.
- **Student** : A person who applies with the system.
- **Course** : Holds the information about courses. For example, credit, quota etc.
- **Course Manager:** Where courses are created.
- **Course Section :** gives the section of the courses**.**
- **Fte Courses:** Full time equivalent courses.
- **Nte Courses:** Non technical elective courses.
- **Te Courses:** Technical elective courses.

- **Student Manager:** Manages all student information.
- **Advisor:** A supervisor who checks course requirements.
- **TranscriptManager:** Generating transcript and computing GPA by failed and successful courses.
- **Transcript:** recorded material that contains the student's letter grade and GPA
- **Java** : A programming language
- **Functional Requirement:** A feature that the system absolutely needs to have.
- **Non-Functional Requirement**: A condition that details the proper method the system should use.
- **JSON** : JSON is a text-based data format that is a simple replacement for XML, which is frequently used on the Web for data exchange.

4. Functional Requirements

• Students must be selected randomly by the system.

• The course data must be incorporated into the system.

• The transcript file has to be read by the system.

• The system has to record anything in the transcript file.

• Transcript files must be created by the system.

• The names of the students must be generated randomly by the system.

• Prerequisites for courses should be under the authority of advisors.

5. Non - Functional Requirements

Usability

• The system must produce accurate and simple-to-understand outputs and logs.

Flexibility

- The system ought to have simple integration for new course additions.
- When new pupils are introduced, the system must be able to include them quickly.

Performance

- Amount of time for the process takes minimum time with maximum quality.

Reliability

- We'll test the project code to make sure it runs smoothly.

Data Integrity

- JSON files will be used to store all course and student data.

Security

- The student data that is stored in the system is private and shouldn’t be disclosed. As a result, the system cannot be accessed from the outside.

Maintainability

- The system determines any potential mistakes and logs them in the appropriate places if they happen. Log files allow for the observation of all mistakes and outputs.

6. Domain Model