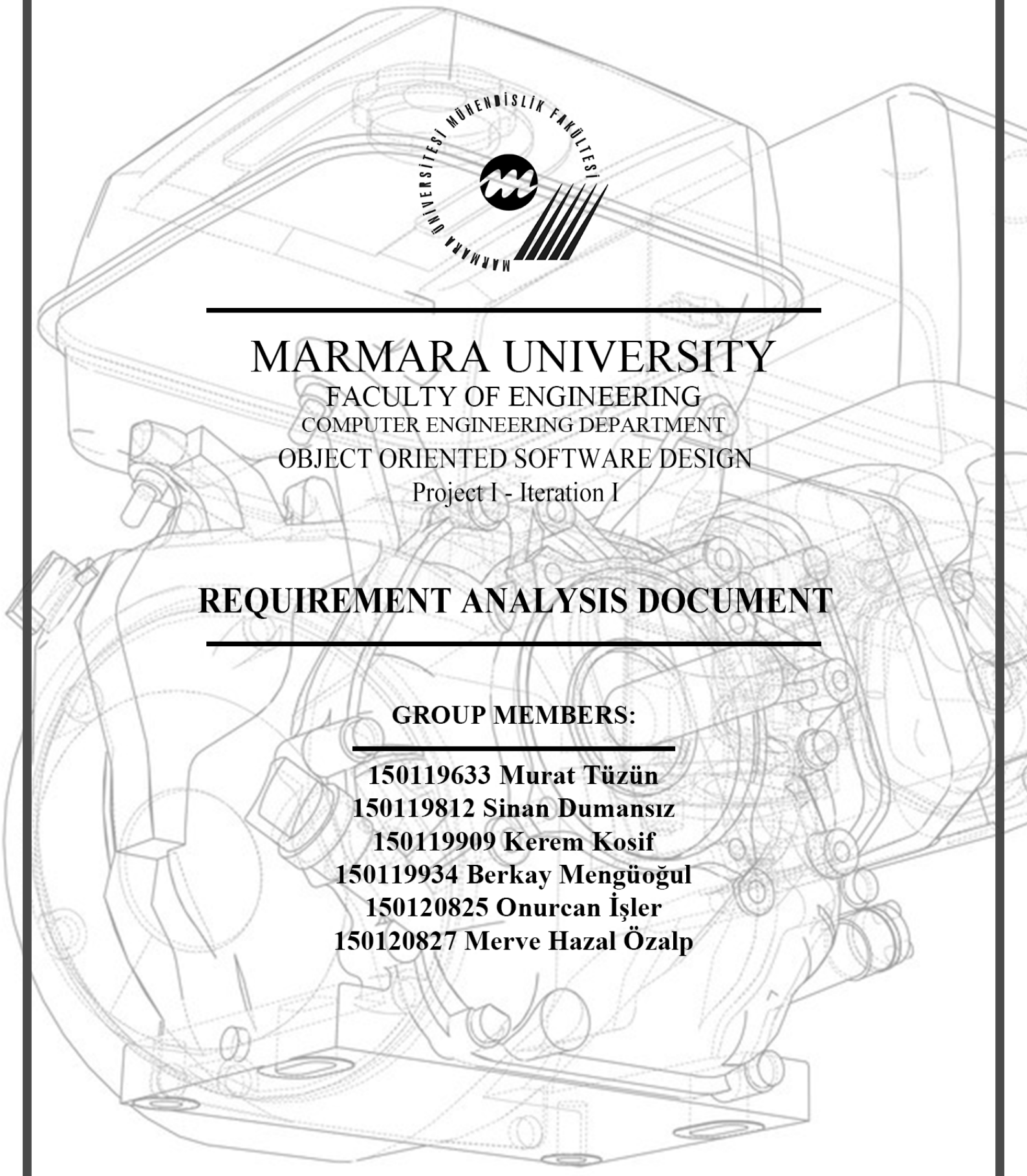

MARMARA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

OBJECT ORIENTED SOFTWARE DESIGN

Project I - Iteration I

REQUIREMENT ANALYSIS DOCUMENT

GROUP MEMBERS:

150119633 Murat Tüzün
150119812 Sinan Dumansız
150119909 Kerem Kosif
150119934 Berkay Mengüoğul
150120825 Onurcan İşler
150120827 Merve Hazal Özalp

Table of Contents

1. Introduction

1.1 Vision

We are tasked with developing an Online Course Registration System. The new system will allow students to add courses, register for courses, view their weekly schedules, and review their transcripts. Academic affairs, advisors, can use the system as a means to manage students' registrations for courses. The system provides a better solution and policy for course registration at Marmara University. The system will accurately provide all the regulations and procedures applied at Marmara University during the course registration of students. The system reduces much time, cost, and resources required in processing the registration information of students.

1.2 Scope

To be noticed on the scope of the system that this system is an Online University Registration System. It is not a University Management System that is much larger than the system we try to build. It is only a part of the university management system. Therefore, we have to pay attention to building applications supporting students to do registration, academic affairs to manage information related to students' course registration.

The aims of our project:

- To provide a system that will speed up the course registration process of the students and minimize the problems experienced by the students during their registrations.
- To provide a better system so that students can fairly enroll in courses at the university.
- To provide a system that can keep track of all the actions that take place during the course selection process. In this way, university administrators will be able to be directly aware of the problems that occurred during the course registration week.

1.3 Constraints

- Devices that can run the JAVA Runtime environment are required.
- Student transcripts must be stored in JSON format.
- Since the program is going to work offline, it is not capable of synchronizing on multiple devices, so that only one user will be able to use the program at a time.

2. Glossary

Advisor: To control a student's class registration process, intelligent registration systems are used. However, some classes need to be checked by an assigned professor due to the complications and complicated conditions of some of the classes.

Credit: It is the weight of a course that helps to measure the status of students. After earning a specific amount of credit, students may either enroll credit required courses or graduate.

CGPA: It is used in universities to evaluate a student's overall academic achievement.

Elective Course: Elective courses are general courses that can be selected by any student in the university. And all students need to take a certain amount of elective courses to graduate.

GPA: Grade point average also known as GPA is one of the student rating standards. This standard shows approximately the success of a student.

Grade: It is the overall measurement for the success of a course. Grades are represented with letters. Following grades indicates that a student successfully passed: "AA, BA, BB, CB, CC, DC, DD, FD". The remaining grades indicate that the student failed the course: "FF, FG, DZ, U”.

Java: An object-oriented programming language which his going to use to develop the system

Log: A log file is a file in computing that records occurring events in an operating system or other program runs, as well as messages sent between users of communication software.

Mandatory Course: These courses are mandatory to take by any student in the University. These courses can vary by the department of a student.

Prerequisites: Some classes in universities require prior knowledge to understand and learn. To avoid giving classes to students with a lack of prior knowledge University system created a prerequisite system. In this system, some courses can only be attended if certain prior courses are passed.

Quota: It is the limit of how many students can enroll in a course. Mandatory Courses don't have a quota, but elective courses have a quota. So, only a limited amount of students will be able to enroll in elective courses.

Runtime environment: The environment in which a program or application is run is referred to as the runtime environment.

Semester: Academic years is divided into two sections that each section is 15 weeks long. A section is called semester and each academic year has fall and spring semesters.

Technical Elective Course: These courses also vary in the department of a student. In the last 7th and 8th semesters, students get access to select Technical Elective Courses and learn topics they like about their department.

Transcript: A transcript is a way to show the success of a student. Transcripts include all the courses a student took with its dedicated grades and at what time courses are taken.

Course Status: It states whether the course is available or not to the student considering the requirements of the student.

Registration Status: It claims the information about enrollment to the course considering the approval of the advisor.

3. Specific Requirements

3.1. Functional Requirements

3.1.1 University

- University hires advisors or registers information of faculty members to the system.
- University registers every student gives them id and uniformly distribute them to each semester.
- University keeps the tracks the current semester.
- University creates every course in the Computer Science department, provides them a name, id, type, credit, lecture hours, and quota if it is an elective course.

3.1.2. Course

- Some courses must require one or more prerequisite courses for students to be able to enroll.
- Depending on the type of the course, if it is an elective course, it must have a quota.
- Available quotas of elective courses must decrease when a student registers for that course.
- Every course has its curve for grading the scores of students.

3.1.3. Student

- Each student must be assigned to an advisor, advisors are necessary to be able to complete the course registration.
- Students will register for courses one by one. Every course registration will be sent to the assigned advisor.
- Students will get an overall score at the end of the semester.
- Depending on the overall score of the student, the course will be classified as passed or failed.
- Students can see their weekly schedule concerning the registered courses during the occurring semester.
- Students can keep the track of their course records at their transcripts.

3.1.4. Advisor

- Advisors in the system have a different set of students who are going to be responsible for their course registration.
- When students send a course registration request, the advisor should be able to identify the students, access their total credits, access their cgpa, access they're failed, passed, and registered courses. Concerning accessible courses advisor should be able to take action of either approving or rejecting the request.

3.1.5. StudentAffairs

- Student affairs create a prerequisite tree of courses.
- They assign prerequisite courses to the related course in the system.
- They distribute the courses into eight different semesters concerning lecture hours, course type, and prerequisite tree to refrain students from enrolling in collapsing courses.
- They create weekly schedules for each semester.

3.1.6. StudentTranscript

- Transcripts keep the record of grades for each student.
- At the end of each semester updated version of each semester should be displayed.
- Old records should be displayed as well most updated ones too.
- The transcript is responsible for identifying is cause has failed or passed.
- Transcript calculates the CGPA

3.1.7. Grade

- The grade is responsible for representing and displaying the score of each student as an integer or letter.
- Grade handles corresponding letter matching of the overall score concerning the curve of the course.

3.2 Nonfunctional Requirements

3.2.1 Usability

The interface is very simple and plain, so it is easy to use. After users provide their credentials, depending on their status they will be able to select available courses.

3.2.2 Reliability

Our product can track every course, student, and advisor participating in terms of courses. Related information changes are logged and stored. So, it can continue where it is left after being shut down. So, this feature makes our product reliable.

3.2.3 Performance

During the registration period. Many students might register for courses in parallel. Responses to each action are handled immediately. So, no student waits for using the program.

3.2.4 Portability

Our product can run on any device which has a java runtime environment.

3.2.4 Security

The product requires authentication that every user is bound to their limitations. From the student's perspective, students can only adjust their courses, actions that can affect other students are restricted. However, the system of the user can access any student by their student id which can cause harm to other students.

3.2.5 Maintainability

The product is extensible which means it is easy to maintain and it is sustainable. Therefore it is effortless to add new features with regards to customers' desires.

3.3 Domain Class Diagram

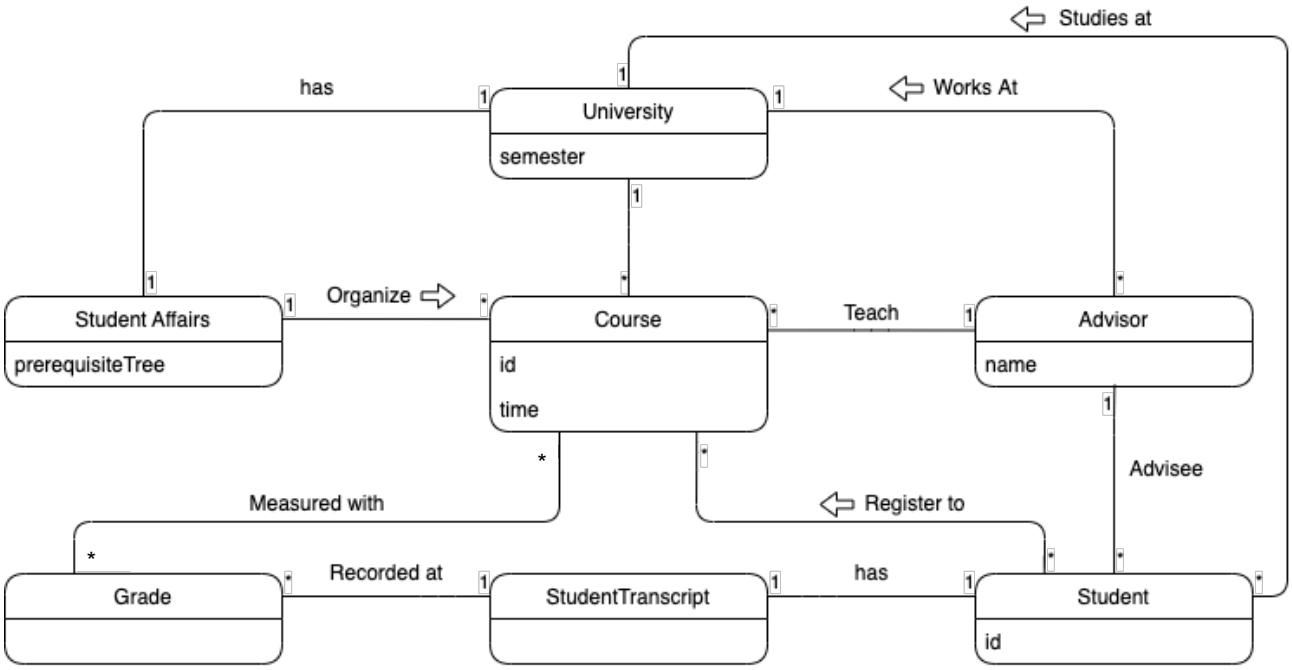

Figure 1: Domain Class Diagram of the System

3.4 Use Cases

3.4.1 Use Case: Register Course

Actors: Student

Stakeholders and Interests:

Student: Wants to register for the right set of courses to be able to ready for the semester.

Advisor: Wants to ensure that students completed their registration process successfully.

System: Wants to provide a platform that both advisors and students can perform tasks accurately and quickly.

Preconditions:

Students must be registered in the system beforehand.

Courses are organized for each semester and their prerequisite course is stated by the Student Affairs.

Main Success Scenario:

1. Student logs in to the system.
2. The system verifies that the student id number is valid.
3. Student searches for the courses by year, semester, faculty, department to enroll in.
4. The system displays available courses for the logged-in student.
5. The student selects multiple courses to register for.
6. The system verifies the maximum course load for the student that has not been exceeded and there is no overlap in the weekly program, checks that prerequisites are satisfied and the student has enough credit to take the course.
7. After the selection process is over, the student confirms the enrollment and sends it to the advisor for approval.
8. Advisor checks all the conditions and approves the registration.
9. The system indicates the student's schedule and notifies that registration is complete.
10. System logs action
11. Student logs out.

Extensions:

1a. At step 1, if an invalid number is entered, the system does not allow access.

5a. Step 5 can be repeated until the student manages all the courses she/he wishes to.

6a. At step 6, if the system can not add the course for any reason then it displays a message to indicate that.

8a. At step 8, if the advisor does not approve one or more courses she/he refuses registration.

8b. The student selects another course.

8c. Student drops the unapproved course.

Special Requirements:

A platform to be able to run the system.

Consistency of courses with their quotas.

3.4.2 Use Case: Advisor Approval

Scope: Advisor Approval

Actor: Advisor

Stakeholders and Interests:
Advisor: Wants to ensure that advised students match the requirements of requested courses.
Student: Wants to register for courses.

Precondition: The student has to send the courses to the advisor's approval.

Main Success Scenario:
1. Advisor logs in to the system.
2. Advisor selects a student from those who sent their registration request.
3. Advisor checks the course considering passed and failed courses of the student.
4. First, if the student has failed courses from the previous semesters, the advisor must register the student for these courses. (High priority).
5. Advisor checks if the student has a Technical Elective course in his registration list. If it has then, It checks the 155 credit rule to take T.E. courses and 4th-year projects.
6. Advisor checks if the student's GPA is below 1.80. Within the system, if the student has a DD or DC grade, it allows them to take them again first.
7. As long as the student's GPA is matching with requirements, the advisor can approve a maximum of 3 courses from the upper semesters.
8. If the student has not completed these checking points the advisor does not approve the courses and sends it back to the student and the student registers for proper courses again.
9. If all requirements are met, the advisor approves the courses.

Extensions:
4a. If the student has less than 165 credits, the Advisor cant give upper semesters courses.
6a.Advisor cant approve if the student selects more than 10 courses in a semester.

3.5 System Sequence Digrams

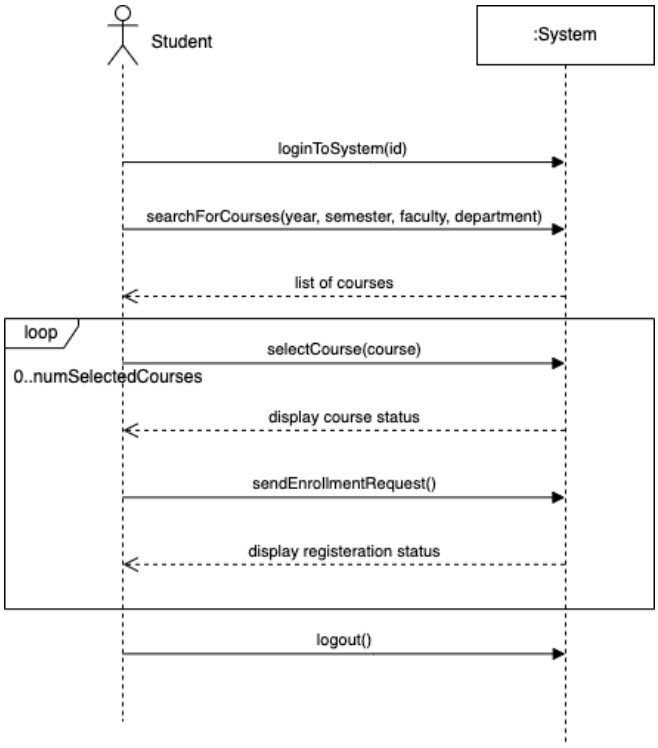

Figure 2: System Sequence Diagram of Use Case 3.4.1

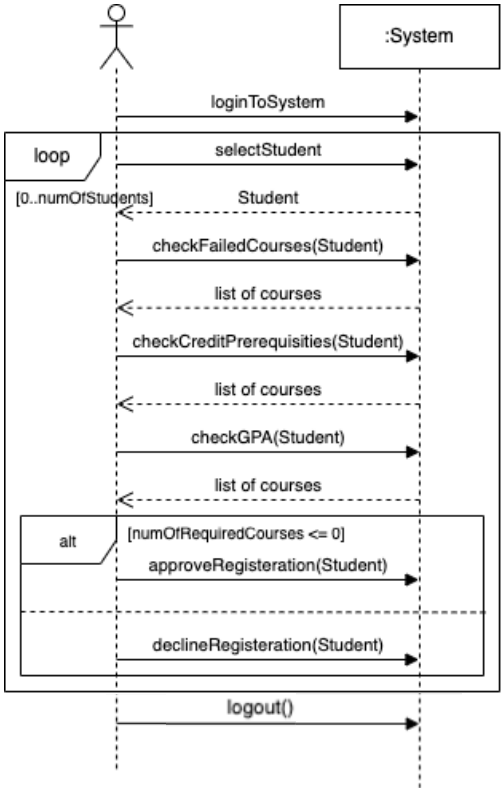

Figure 3: System Sequence Diagram of Use Case 3.4.2

4. References

[1] Fowler, M. (2015). *UML distilled : a brief guide to the standard object modeling language.* Boston: Addison-Wesley.

[2] Larman, C. (2005). *Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development.* Upper Saddle River, N.J.: Prentice Hall Ptr.

[3] Booch, G. (2007). *Object-oriented analysis and design with applications.* Reading, Massachusetts: Addison-Wesley.

[4] Mclaughlin, B., Pollice, G. and West, D. (2007). *Head first object-oriented analysis and design.* Beijing: O'reilly, I.E.

[5] www.tutorialspoint.com. (n.d.). *Object Oriented Analysis & Design Tutorial - Tutorialspoint.* [online] Available at: https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm.

[6] App.diagrams.net. 2021. Flowchart Maker & Online Diagram Software. [online] Available at: https://app.diagmrams.net [Accessed 10 November 2021].

[7] www.marmara.edu.tr 2021. *Curriculum - Computer Engineering - Faculty of Engineering - Marmara University.* [online] cse.eng.marmara.edu.tr Available at: http://cse.eng.marmara.edu.tr/en/undergraduate-program/curriculum [Accessed 11 November 2021].