

Documentation

“University Management System”

Dmytro Andrieiev

s21353

Group 12c

Table of Contents

- 1.** User requirements.
- 2.** The use case diagram.
- 3.** The class diagram – analytical.
- 4.** The class diagram – design.
- 5.** The scenario of “Review assignment”.
- 6.** The activity diagram “Review assignment”.
- 7.** The state diagram for “Student”.
- 8.** The interaction (sequence) diagram for “Review assignment”.
- 9.** The GUI design.
- 10.** The discussion of design decisions and the effect of dynamic analysis

1. User requirements.

For the “University management system” we want to Tutor and Student to be presented in the system. Every User in the system must have Name, Surname, Date of Birth and Address. Unregistered user can register in the system. Registered user must have possibility to login and logout also he/she should have possibility to change password. Also, he/she must have possibility to send application for work and/or study program.

Student in the system must have student unique id, information about current semester, scholarship is optional. Also, student described by his/her status { "enrollee", "stationary student", "online student», “distance mode”, “on academic leave", "graduated" }. Student must have possibility to apply for scholarship, send assignment, send thesis, and there should be possibility to update student status.

Thesis are connected to student. They have title and maximum period to write them which equals to 4 semesters.

Each student attends some number of courses, which is described by course tag, it’s description and include course program.

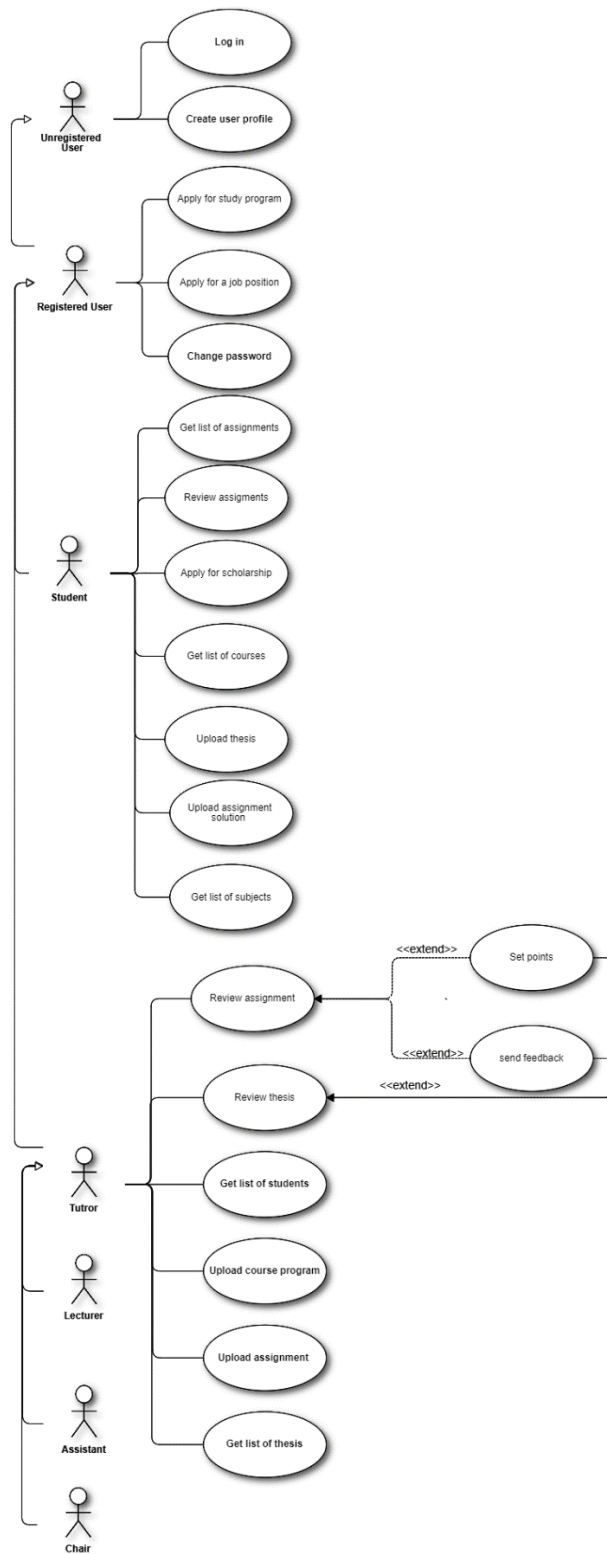
Course is connected to the number of subjects which have names, their type and topic that they related to.

For each subject there are number of assignments, which have description, start date of assignment and end date of assignment , maximum points available for this task and points which student has for this assignment.

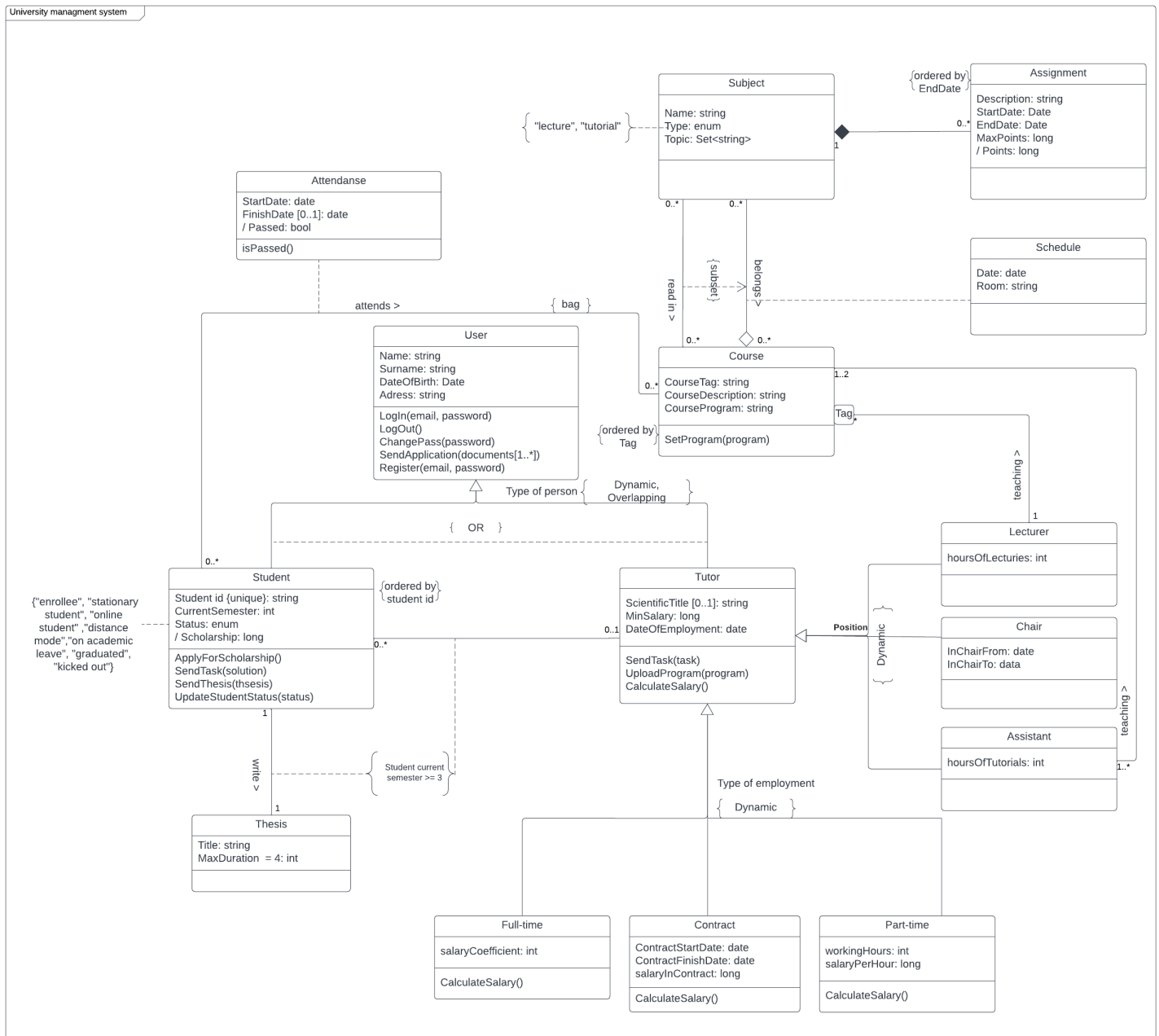
Course and Subject are connected by Schedule which has fields: time and room where this course is read in.

Each course is read by tutor. Tutor has scientific title, minimal salary, and date of employment. Tutor has position in university (Lecturer, Chair, Assistant), depends on the position tutor (lecturer) has number of lecture hours per month. Tutor(chair) has start and end date on the dean position. Tutor (assistant) has number of hours for the tutorials for which he/she responsible for. Also, Tutor has type of employment (Full-time, Part-time, Contract). Depends on type of employment salary is calculated in different ways. Tutor with full-time of employment has coefficient on which minimum salary is multiplied. Tutor with part-time employment has some number of hours per month which can be multiplied by salary per hour, in this way salary for part-time worked is calculated. Tutor with contract has salary which is written in contract. Each tutor must have possibility to upload assignment and subject program.

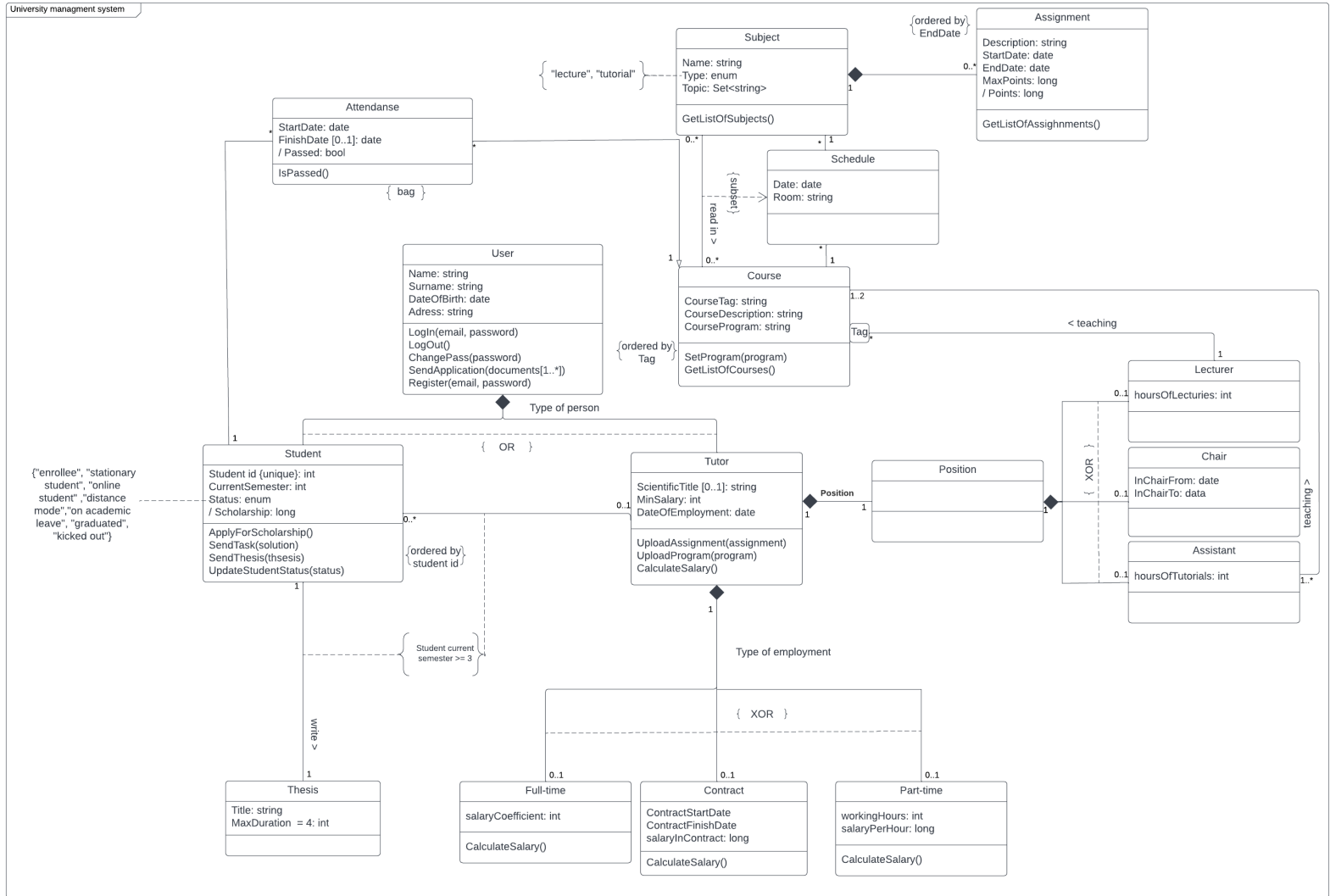
2. The use case diagram.



3. Class diagram – analytical.



4. The class diagram – design.



5. The scenario of “Review assignment”.

1. Use-case: Review assignment

1.1. Actors

Student

1.2. Purpose and context

Student wants to review assignment

1.3. Dependencies

1.3.1. Included use-cases

None

1.3.2. Extended use-cases

None

1.4. Assumptions and pre-conditions

Valid and verified person

1.5. Initiating business event

None

1.6. Basic flow of event

1. Student select course.
2. Student select subject.
3. Student select assignment.
4. Student click on assignment.
5. Information about assignment are shown to a student.

1.7. Alternative flows of event

- 1.7.1. There is no assignment to choose.
 - 3a1 Student does not choose any assignment.
 - 3a2. Student finish his/her session with system.
- 1.7.2. There is no course to choose.
 - 1a1. “No course(s)” is shown to a student.
 - 1a2. Student finish his/her session with system.
- 1.7.3. There is no subject to choose
 - 2a1. “No subject(s)” is shown to a student.

2a2. Student finish his/her session with system.

1.8. Extension points

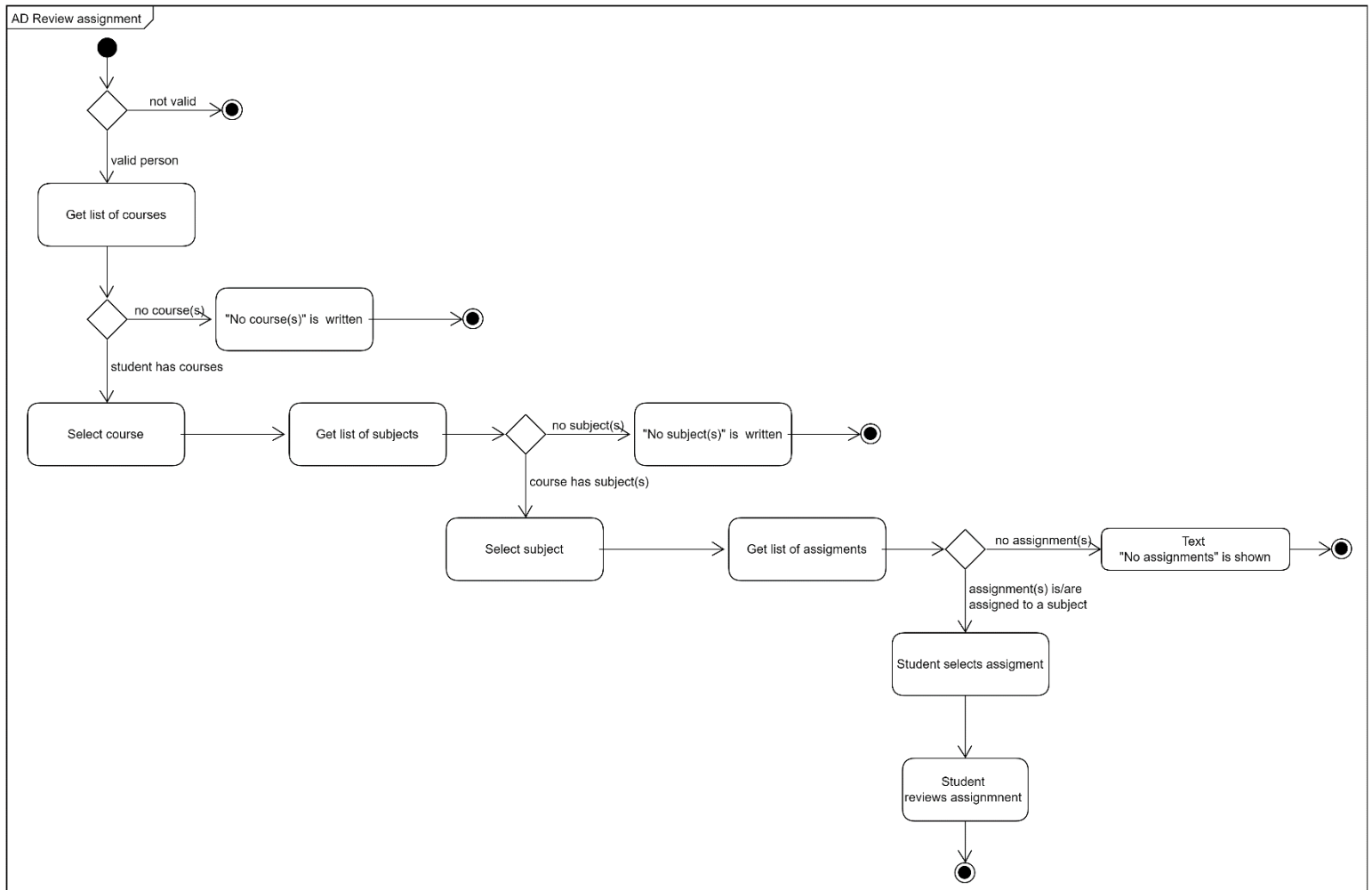
None

1.9. Post-conditions

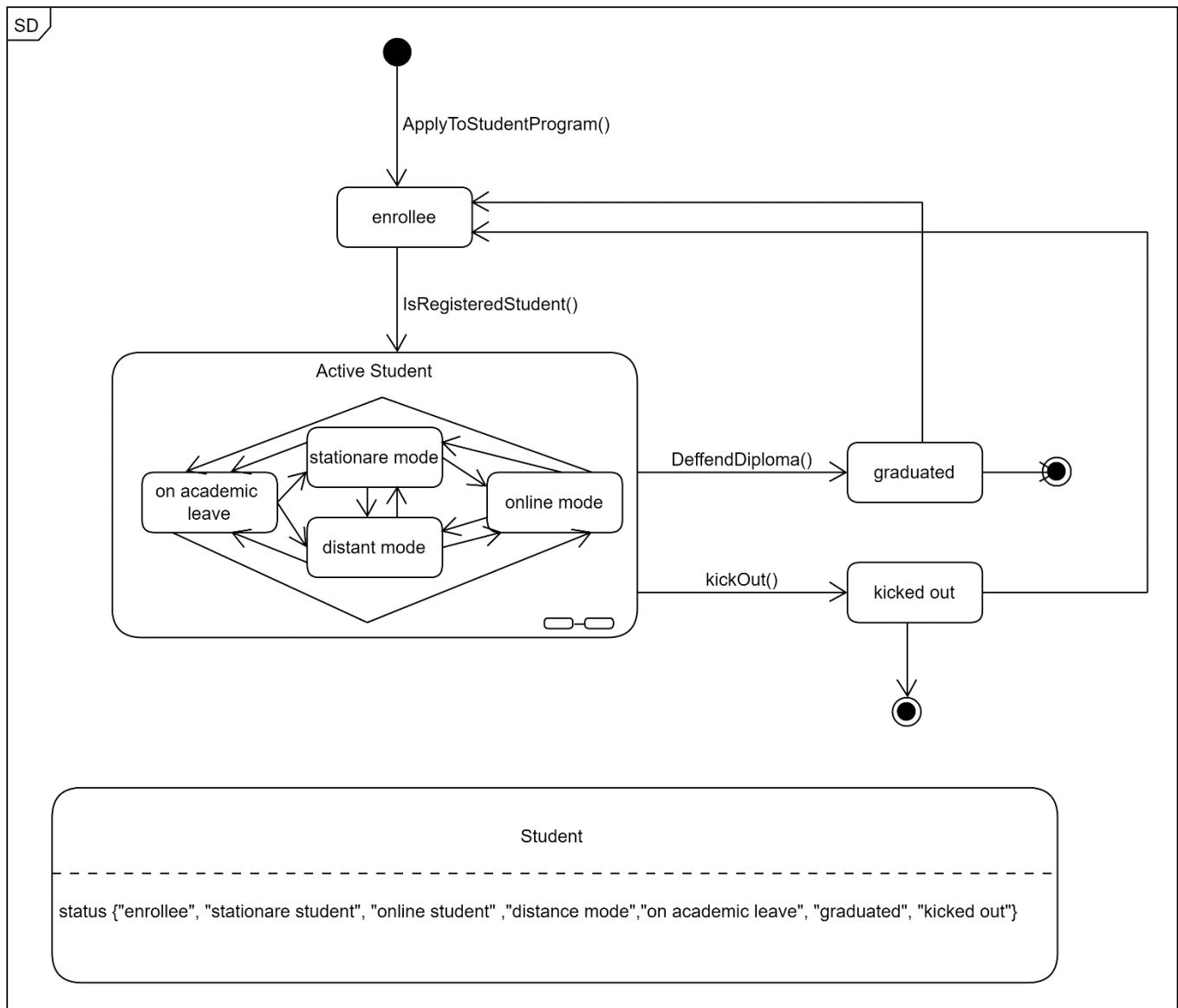
Assignment is shown to a student

.

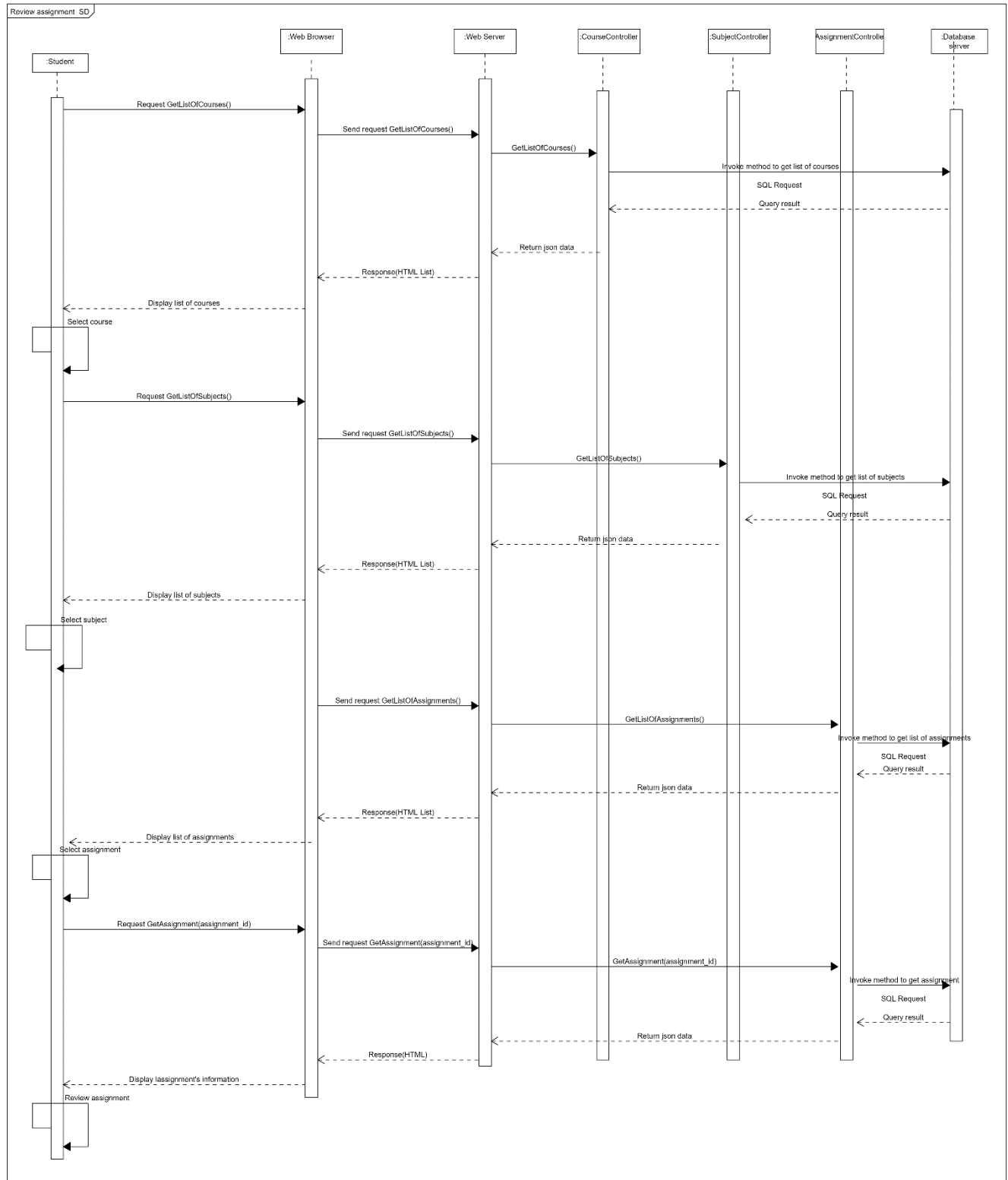
6. The activity diagram “Review assignment”.



7. State diagram for “Student” class.



8. The interaction (sequence) diagram for “Review assignment”.



9. GUI design.

The image displays a minimalist user interface within a light gray rectangular frame. At the top, there are two side-by-side dropdown menus. The left menu is labeled 'Select course' and the right one is labeled 'Select subject', both with a small downward arrow icon on their right side. Below these two menus is a search bar, which is a horizontal rectangle containing a magnifying glass icon followed by the text 'Search'.

Figure 1 Basic Screen

Select course▼

"No course(s)"

Select subject▼

🔍 Search

Figure 2 Empty Course List

Select course

Graphic

Data Science

Software engineering

Search

Select subject

Figure 3 Course List With Courses

Data Science

Select Subject

"No subject(s)"

 Search

Figure 4 Subject List Without Subjects

Data Science

▼

Select Subject

▼

Data Mining

Machine Vision

Data Pipelines

Basics of AI

🔍 Search

Figure 5 Subject List With Subjects

Data Science

Machine Vision

Search

Figure 6 Selected Course and Subject

Data Science

Machine Vision

Search

Assignments:

1

OpenCV

2

SIFT algorithm

3

CNN

4

Optical Flow

5

Keras

6

Identification of coins

Figure 7 List with Assignments

Data Science

▼

Machine Vision

▼

🔍 Search

Assignments:

"No assignments"

Figure 8 List without Assignments

Machine Vision

OpenCV

Description

Please write in Python using OpenCV, TensorFlow and Keras libraries a code that recognizes three different fruits (banana, orange, lemon) in the image. The following points should be included:

1. Prepare a training set using basic graphic operations. The set must consist of at least 450 photos and descriptors.
2. Design a convolutional neural network (with intermediate maxpooling layers)
3. Propose proper activation functions
4. Optimize the network.
5. Perform a test classification. The training data set and the code, along with the video generated from its optimization and operation, should be delivered via the Teams platform within the time specified in the task.

Start date: 15.06.2022, 11:59 PM

End date: 25.06.2022, 11:59 PM

Points: 0/10

Figure 9 Assignment description

10. The discussion of design decisions and the effect of dynamic analysis

In case of Student and Tutor classes it was decided to use both dynamic and overlapping inheritance because student can become tutor and otherwise. Also, student can be in a role of tutor and student at the same time. It was decided to implement dynamic and overlapping inheritance by using composition with {OR} constraint.

In case of type of employment, it was decided to use the dynamic inheritance because tutor can become as well as part time worker or full-time worker or assigns contract with university. It was decided to implement dynamic inheritance by using composition with {XOR} constraint.

For position (Chair, Assistant, Lecturer) of tutor it was chosen to use dynamic inheritance. It was decided to implement dynamic inheritance by using composition with {XOR} constraint.

As Tutor implements two types of inheritance the multi-aspect inheritance was implemented. It was decided to implement it by replacing Position hierarchy with a composition.

Regarding constraints. System contains unique constraints in case of student id, because student id must be unique for each student. Also, system contains subset constraint in case of subject. Ordered constraints are placed for student class (ordered by student id), course class (ordered by tag) and assignment (ordered by deadline). Bag constraint was used in case of Student-Course association, because each student can take each course few times and it should be possible to have a few records with the same objects. There are few places where for an attribute constraint was used. In the first place (Thesis class) we limit the number of the semesters to complete thesis. In case of Tutor we want to set a minimum salary and check every time if salary is equal or higher than minimal.

Concerning dynamic analysis of chosen use-case ("Review assignment") has only 3 alternative flows. All of them are end use-cases. The first alternative case works in case when there are no courses to which student is assigned, "No courses" is displayed. As a result of the second alternative flow message "No subject(s)" is shown to the student if

there are no subjects assigned to a course. In case if there are no assignments “No assignment(s)” is displayed to a student.

UI was done in minimal way by using MarvelApp web platform. It consists of minimal number of elements in order not to overload UI with extra elements.

References

<https://drive.google.com/drive/folders/1sC6zgpQg3sS1Y5u8ljnfmrUSSmyg7r6l?usp=sharing>