In fact, when developing a new software, the software development team must describe the software requirements in software requirements specification before they do the design and coding. They describe the software requirements by using text, formulas, drawing models and table description. The software tools department at FPT University want to develop a new software tool named EduNext website to allow students to learn many courses in the constructivism way. Following the constructivism way, the students will build up their knowledge more actively by searching content on the internet, reading text books, interacting with other students in the same group or in the same class. The EduNext website need to have many functions which allow importing data about student list from FAP system, displaying the list of students, updating the list of students, adding questions, updating questions, listing questions, creating student groups in the same class by teacher, answering the questions by students, voting answers by students, voting answers by teacher, signing in EduNext by using @fpt.edu.vn email accounts of students or teachers. Besides, EduNext can record data of the interaction processes between students when students at FPT University post the answers of the questions, reply or comment the answers of other students, vote the answers of other students or even chat with other students. Use case approach is one of the most effective ways to describe the software requirements.

**All the answers must be in English and reflect this exam paper. If your answers of the questions in this exam paper have any keywords not related to this exam paper, the answers of the questions will get ZERO.**

**Question 1:** Complete all parts of the first page of the software requirements specification. (1 point)

Software Requirements Specification

for

<Project> (0.25 point)

Version 1.0 approved

Prepared by <author> (0.25 point)

<organization> (0.25 point)

<date created> (0.25 point)

**Question 2:**

Use case diagram is an effective way to visualize the interaction between actors and the software system. One Use case diagram need one rectangle which represents one system or one software application. One oval represents one use case. The name of the use case must begin with the verb and follow by an object.

1. In this exam paper, the name of the rectangle is: ………( 0.1 point)
2. The actor may be human, other software systems or devices. In this exam paper, list the name of >= 3 actors is: ……..(0.6 point)
3. In this exam paper, list the name of >= 5 use cases are: …….(1 point)
4. The actors are inside or outside of the rectangle? ……… (0.1 point)
5. The ovals represent the use cases are inside or outside of the rectangle? ………(0.1 point)
6. The primary actor of the use case **add new question** is: …….(0.2 point)
7. The secondary actor of the use case **add new question** is:….. (0.2 point)
8. Based on the fact you already used EduNext website. The use cases in this exam paper may have relationship.

8.a List the name of two use cases that have **extend** relationship: …….(0.3 point)

8.b The name of **base use case (in 8.a)** is: …..(0.2 point)

8.c The name of **included use case (in 8.a)** is: ……(0.2 point)

8.d The dashed line with an arrow that points to which use case? …..(student write the name of use case) (0.1 point)

1. Based on the fact you already used EduNext website, the use cases in this exam paper may have relationship.

9.a List the name of two use cases that have **include** relationship: …….(0.3 point)

9.b The name of **based use case (in 9.a)** is: …..(0.2 point)

9.c The name of **extend use case** **(in 9s.a)** is: ……(0.2 point)

9.d The dashed line with an arrow that points to which use case? …… (student write the name of use case) (0.2 point)

**Question 3:** Writing detail one use case, the use case **adds (+) new question** in the following format. (4 points)

1. UC ID and Name: (0.1 point).
2. Created By: (0.1 point).
3. Date Created: (0.1 point).
4. Primary Actor: (0.2 point).
5. Secondary Actors: (0.2 point).
6. Trigger: (0.3 point).
7. Description: (0.1 point)
8. Preconditions: (0.3 point).
9. Postconditions: (0.3 point).
10. Normal Flow: 0.7 point).
11. Alternative Flows: (0.7 point).
12. Exceptions: …..(0.3 point).
13. Priority: (0.1 point).
14. Frequency of Use: (0.1 point).
15. Business Rules: (0.2 point).
16. Other Information: (0.1 point).
17. Assumptions: (0.1 point).

**Question 4:** writing 2 non-functional requirements.

1. The term of the most important non-functional requirements and at least 2 statements. Should be specific in number, explain the number so that the tester can test the software and make conclusion the test is passed or failed. (0.5 point)
2. The term of the second important non-functional requirement and at least 2 statements. Should be specific in number, explain the number so that the tester can test the software and make conclusion the test is passed or failed. (0.5 point)