**Exercise 1:**

Do you know that it costs a lot of money to get a “Certiﬁed Java Programmer” certiﬁcate? It could cost you thousands of euros. Let’s imagine we will develop a browser-based training system to help people prepare for such a certiﬁcation exam.

* A user can request a quiz for the system.
* The system picks a set of questions from its database, and compose them together to make a quiz.
* It rates the user’s answers, and gives hints if the user requests it.

In addition to users, we also have tutors who provide questions and hints. And also examinators who must certify questions to make sure they are not too trivial, and that they are essential.

Quests:

* *Make a use case diagram to model this system*.
* Work out some of your use cases.
* Since we don’t have real stake holders here, you are free to ﬁll in details you think is essential for this example.

Java Certification Online

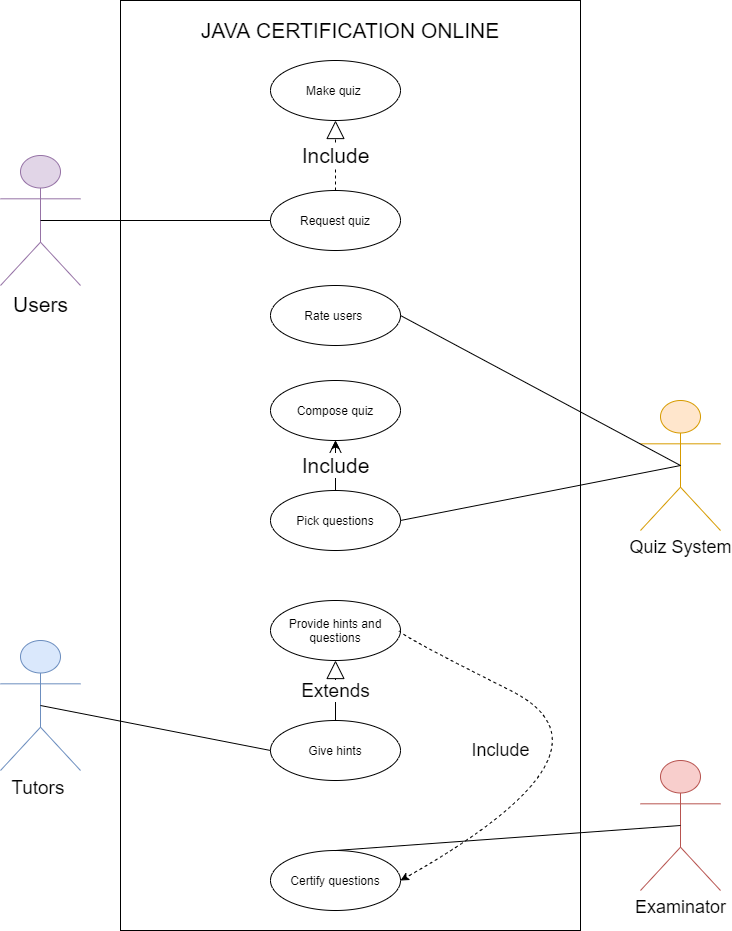
List of actors:

* Users: can be programmers, students, etc. Anyone wants to be certified with Java certification
* Quiz System
* Tutors
* Examinators

List of use cases:

|  |  |
| --- | --- |
| **Use case** | **Description** |
| Request quiz |  |
| Make quiz |  |
| Pick set of questions |  |
| Compose quiz |  |
| Rate user |  |
| Gives hints | User reporting posts from users due to some reasons |
| Provide questions | User or admin can view new message in message panel |
| Provide hints | User or admin can delete message in their inbox |
| Certify questions | Every user can create group |

Use case diagram:



**Exercise 2:**

Suppose we want to develop software for an alarm clock.

* The clock shows the time of day.
* Using buttons, the user can set the hours and minutes ﬁelds individually, and choose between 12 and 24-hour display.
* It is possible to set one or two alarms.
* When an alarm ﬁres, it will sound some noise -> clock.
* The user can turn it off, or choose to ’snooze’.
* If the user does not respond at all, the alarm will turn off itself **after 2 minutes**->clock. ’Snoozing’ means to turn off the sound, but the alarm will ﬁre again after some minutes of delay. This “snoozing time” is pre-adjustable.
* Identify the top-level functional requirement for the clock, and model it with a use case diagram

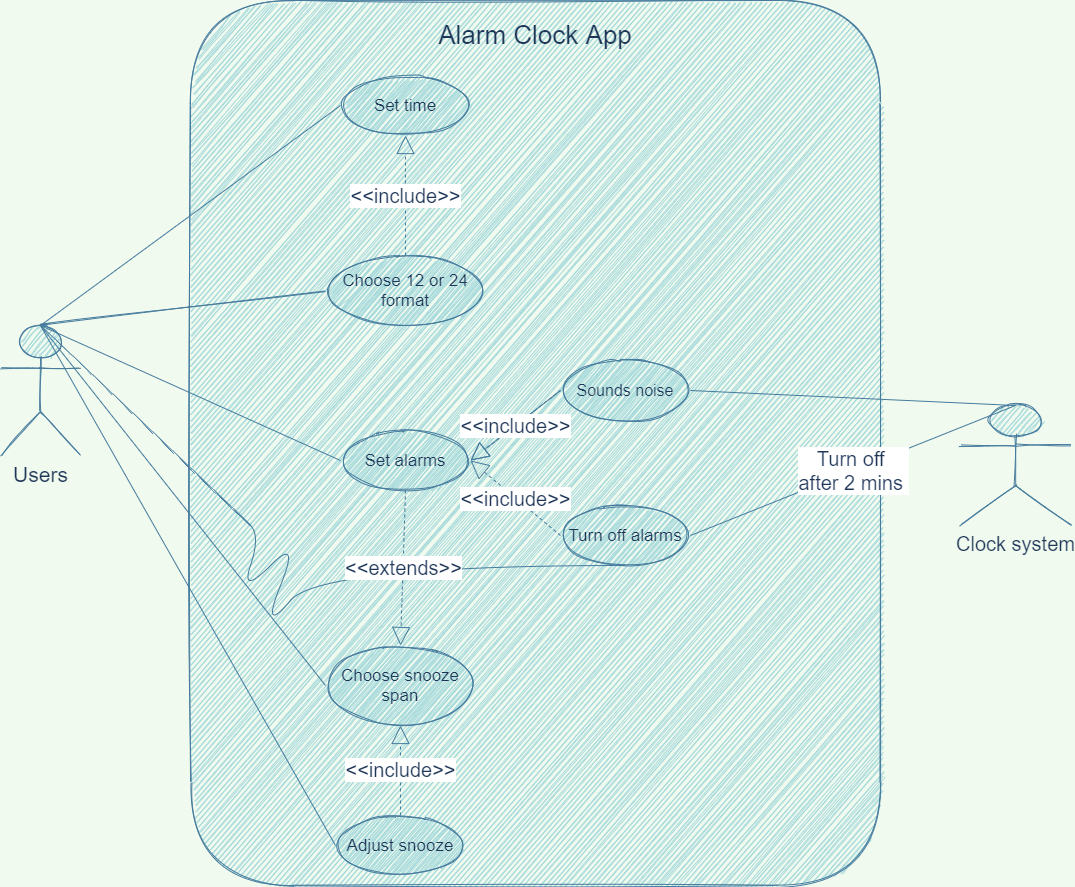
**Functional requirements:**

**Process – oriented**

* DTSquad allows new users to create new accounts.
* DTSquad allows users who have already created their accounts to forget their password and get it back by sending a private email to their TDTU emails and it is automatically
* DTSquad will send you friend suggestion list who will be your classmates and lecturers in new classes
* DTSquad allows registered users to add more friends on their Friends lists or unfriend
* DTSquad allows registered users report groups, clubs, status or even accounts that are against TDTU’s rules
* DTSquad allows users to update their information from description, changing status content, comment content
* DTSquad allows users to share picture, audio, video, documentation files
* All users will receive notifications if there are news, friend acceptation, information updating successfully, etc.

**Information – oriented**

* DtSquad must contain all information of students, lecturers and parents in its database
* DTSquad includes friend tree of every students to make suggestions
* DTSquad will save email address marked as black list mails who are not parents but strangers outside TDTU
* DTSquad will save user’s online history and who were online for 1 hour and will delete the history if the user comes back within 1 hour or after 1 hour

****

**Exercise 3:**

Using your knowledge of how an ATM is used, develop a set of use cases that could serve as a basis for understanding the requirements for an ATM system.