TEAM #18

Project Requirements

TEAM MEMBERS

- Naveen Muralidhar Prakash - Person# 50208032
- Barath Eswer Nagasubramaniyan - Person # 50207356
- Khushbu Mittal- Person# 50169099
- Jiawei Zhao -Person# 50207069
- Vanshika Nigam- Person# 50208031

Outline:

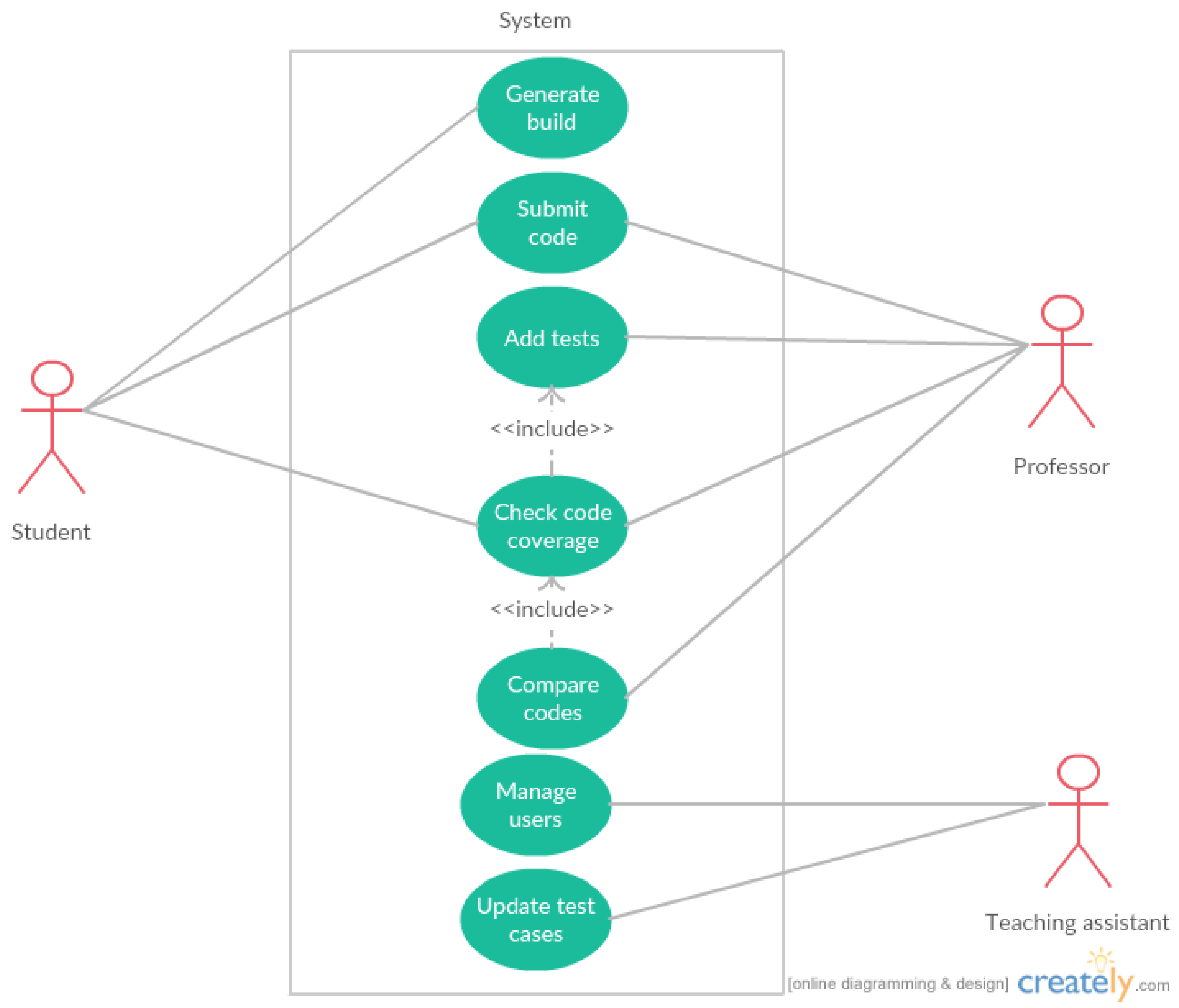
Use case: Evaluate Code Coverage

Goal	To evaluate the percentage of code covered by the classes and methods passed to a library built on top of JaCoCo
Users	Students, Professors, Teaching Assistants
Brief Description	The user passes the name of a method and class to a library, and it returns a report with each of its code coverage in percentage
Rationale	Writing efficient code is an essential skill in software development. Recruiters are constantly looking for professionals, who are not only proficient, but also capable of writing efficient scalable code, thereby reducing overhead costs. This library, which acts as a diagnostic tool, helps in finding areas of a program that are exercised by a set of test cases, thus providing a quantitative measure of the code coverage.
Preconditions	1. All the students must have registered for the course. 2. The library and test cases must be runnable from Java application 3. The system should have been built upon the existing JaCoCo library.

	4. Code coverage must be returned as a percentage of class and method coverage.
Outline Representation	1.The user compiles and generates a build for the Java code. 2. The user commits the code to the branch as per the deadline mentioned by the professor. 3. The professor adds the specific test cases into the library for a specific program 4. This will build and test against the JUnit test cases to find the class and method coverage as per the number of test cases returned. 5. A report containing the results is generated in HTML / XML / CSV format as supported by JaCoCo
Alternate Flow	1.Mismatch or invalid class and method names will be handled by the exception handlers and prompts the user with an error message. 2. Unauthorized access to modify the library or test cases will be reported to admin. 3. Files size exceeding limit will be reported to users by the error handlers
Postconditions	1. The code coverage percentage for the respective methods / classes are

	generated and reported in the format supported by JaCoCo. 2. A technical document describing the library concepts and deliverables will be given for future developments / open source
Risk analysis	As it is a complete development project the possible risks involved can be minimized by proper design patterns and continuous integration in the production environment

Use case diagram:



User Stories:

As a professor I want a portal to compare the code coverage among various students so that I can grade relatively.
This helps in assigning a priority level to each test case and in calculating the class average over a particular test case.

As a teaching assistant, I want to have the ability to change access specifications of users, so that I can maintain log activities of current users .
This helps in maintaining fixed set of users and evaluating their build submissions.

As a student I want a portal to upload my code so that my build gets evaluated and reports are generated.
This helps in fixing the issues during the dry run.

As a teaching assistant I want all the library files in a remote repository so that the logic can be abstracted from the end user.
This prevents the users from modifying program logic as per test cases.

Discarded Roles:

Database administrator: All activities are performed within the existing grading system and it is handled by a database administrator. Hence no administrator is required to take care of new inclusions.

Network administrator: The developed system will be utilising the infrastructure provided to the existing grading system. Hence we do not need the network admin to set up new infrastructure.

Software developer: Once the system is deployed in production environment there will be a less maintenance work from the developers.

Head of the department: A head of the department can directly communicate with the professor for the student performance. He may not be required for the system.

Quality Assurance Engineer:
Although a quality engineer is responsible for ensuring the quality of the system we may not need a constant support after the deployment.