1. **Team Name:** !Failing
2. **Team Leader for this deliverable:** Kevin Cadavillo
3. **Team Members:** Jack Wu, Feiying Zheng, Jake Zhou
4. **Meetings:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Time-date** | **Attendees** | **Agenda** | **Action Items (who will do what)** |
| 11/19/19 | All | Understand what is expected of us for this deliverable, divide up work for this deliverable, and identify the framework we will use for unit tests | **Jake** to write test cases for IN: 001-004; DB: 101; OT: 101-102; NonFunc: 2, 3, 5.  **Kevin** to write test cases for IN: 005, 101-102, 302; TK: 101; DB: 102, 202; OT: 103, 105, 301; NonFunc: 1, 6.  **Feiying** to write test cases for IN: 201-203, 401; TK: 201, 401; DB: 201; OT: 201-202, 401; NonFunc: 4, 7.  **Jack** to write test cases for IN: 301; TK: 202, 301-302; OT: 104, 302. |
| 11/20/19 | All | Finish writing test cases using the template; clear up any confusion about how we plan to implement that might be relevant to writing these tests; discuss questions to ask professor in lecture. | **Feiying** to write JUnit tests for Chat, Message.  **Kevin** to write JUnit tests for Course.  **Jack** to write JUnit tests for Session.  **Jake** to write JUnit tests for Student.  All members to write skeleton methods for the major entity classes as needed so that project will compile (since unit tests will call the entities’ methods). |

1. **Weekly Time Logs:**

|  |  |  |
| --- | --- | --- |
| **Person** | **Total Time in minutes** | **Tasks** |
| Kevin | 280 | -Worked on test cases for IN: 005, 101-102, 302; TK: 101; DB: 102, 202; OT: 103, 105, 301; NonFunc: 1, 6.  -Implemented entity class Course.  -Created JUnit tests for entity class Course.  -Wrote skeleton methods for Student classes so that project would compile after writing unit tests calling these methods. |
| Jack | 240 | -Worked on test cases for IN: 301; TK: 202, 301-302; OT: 104, 302.  -Created JUnit tests for entity class Session.  -Set up test class for Student. |
| Feiying | 250 | -Worked on test cases for IN: 201-203, 401; TK: 201, 401; DB: 201; OT: 201-202, 401; NonFunc: 4, 7.  -Created JUnit tests for entity classes Chat and Message. |
| Jake | 265 | -Worked on test cases for IN: 001-004; DB: 101; OT: 101-102; NonFunc: 2, 3, 5.  -Created JUnit Tests for entity class Student. |
| **Total:** | 1035 |  |

1. **Issues:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue Number** | **Discovery Date** | **Resolution Date ( Est. – Act. )** | **Responsible Person** | **Description (Prob / Resolution)** |
| 01 | 11/21/19 | 11/21/19 | Jake | With the better understanding of communication cohesion in mind, I moved the addCourse and removeCourse methods out of the EditCourseActivity boundary class and into the Student entity class instead. This should make our code more cohesive and reasonable going forward. |

1. **Files and repository locations:**

|  |  |  |
| --- | --- | --- |
| **Filename** | **Location** | **Contents** |
| TestDesignDoc.docx | Deliverables/TestDesign | Test Design Document |
| TeamReport.docx | Deliverables/TestDesign | Team Report |
| (all files in this location) | app/src/test/java/model | JUnit tests |

1. **Plans for Coming Week:**

The team will have a digital meeting on Sunday to create a work plan for implementation. Kevin has set up a MongoDB cluster for the project, so we can discuss further how we plan to structure and access our data.

1. **Comments:** *a paragraph from each engineer describing what they have done/learned from this deliverable*

**Engineer 1:** Kevin

For this deliverable, I was team lead. In addition to writing the Team Report, I organized our meetings, figuring out the time we should meet and the agenda for the meetings. I made sure everyone understood what they had to do after these meetings and that we had the resources we needed to accomplish our tasks. I wrote several test cases and wrote the JUnit tests for the Course class (for this, I ended up implementing Course in its entirety, since it’s a very simple class), and I set up skeleton methods for the Student class so that the unit tests for Student Jake wrote wouldn’t cause compilation issues. It was interesting writing and executing these unit tests, and in fact it revealed an issue with my implementation of Course, so it really showed me how useful it is to write unit tests, which was a good learning experience.

**Engineer 2:** Jack

This week, I was responsible for writing the test cases for the Session class. Throughout this process, I learned a

lot about the importance of unit testing and testing in general. When we write these tests, we are predetermining how we would like our software to operate. However, I don’t expect our system to pass all of the tests immediately. Furthermore, I think that the testing phase of this project will be even more tedious than the implementation phase because of the nature of how the tests are formed. If and when an error occurs in the testing phase, not only do we need to check that the code for the program is not at fault, but we must also check that our implementation of the tests is free of any bugs. This seems like one of the biggest downsides of test driven development. However, I also expect that this process will result in our application performing at a much better level.

**Engineer 3:** Jake

This week was probably the first time that I felt unsure of how to proceed for the deliverable. Identifying and documenting tests that we needed was an informative experience, and was not hard to pick up. But I’ve never had experience doing formal unit tests before. I was able to learn how to do so from my teammates and from my own research online. With the help of these resources, I was able to complete the many methods contained in our main entity class. I also started writing some basic implementation for some of the methods for the Student entity. I also incorporated some of the stuff that I learned in class since the last deliverable to move some methods so that our code made more sense. With this deliverable done, I think it is time that I start to incorporate MongoDB into our app and start playing around with code involving that so I can be ready to implement all the database related features soon.

**Engineer 4:** Feiying

I mostly worked on test cases that covered the chats list, chatbox messaging, set up study sessions, and blocking requirements. Additionally, I created the unit tests for the Chat and Message classes. I tried to think of the different possible inputs for a particular function, and learned how to divide those inputs into different classes and create test cases. From this, I realized that the user could potentially make a lot of errors, or give invalid inputs. Creating these test cases made me think of what the outputs should be when these errors occur, and we need to make sure that we handle these errors when we implement so that there will be no unexpected outputs. From this deliverable, I learned that it is important to have test cases that cover all of our system requirements. I definitely think having these test cases will help guide us during implementation since we now know what the expected results should be. I also realized how useful the unit tests can be since it helped catch some mistakes we made in our entity classes that we did not even realize we had.