# TESTING METHODOLOGY

Team name: BB8

Project name: DukeTutor

## PART 1

### TEST PLAN OVERVIEW

This Test Plan we created is used to communicate the way of testing with team members and stakeholders. It includes objective, methodology and schedule three parts. In methodology, we want to introduce unit test, component test, and system test separately. However, we don't want to include specific test cases in this document.

### TEST METHODOLOGY

#### UNIT TEST

In unit test, we want to focus on testing each class and model. The unit test is divided into two parts: frontend and backend. In frontend part, we will make sure each object creation activity will send the correct json request to the server. In backend part, we will make sure the server will respond the correct form of json object or array. Unit Testing consists of both verification testing and validation testing. In verification testing, given an expected input, we should have an expected output. In validation testing, we use invalid inputs to test all the corner cases, to see if all exceptions are handled correctly. We should get the proper error message, and the system should not fail quietly.

#### COMPONENT TEST

In component test, we focus on testing each functionality of our project. So we combine frontend and backend together because each functionality has both parts. Each component should behave according to its specifications, assuming each unit has been tested properly. Notice that interface errors in the composite component may not be detected by this because these errors results from interactions between components.

#### SYSTEM TEST

System testing focusing on checks the compatibility and interaction among components, ensuring transferring valid data in the right time across component interfaces. We plan to go through our app as a user and test whether the interaction of components are working properly.

# PART 2

## ACTUAL PLAN

(**Note**: we are not dividing it into different sprint, but the start and finish time requires us to align with actual sprint time)

F: Frontend; B: Backend

| Test Type | Test Object | Owner | Start | Finish | Test Case |
|---|---|---|---|---|---|
| Unit Test | User | F:Yunjing Liu<br>B: Rui Zhang | 2/22 | 3/4 | F:<br>1. Build local unit tests using the JUnit 4 framework.<br>2. Run verification tests to check whether the user provided information are transformed in the correct JSON array format.<br>3. Run validation tests to check whether the username, password, email address and session information are valid and in the correct format. Try different corner case. If not, test whether an exception will be thrown.<br>4. Run verification tests to see whether each view can be properly displayed.<br>B:<br>Use Python's unit test module.<br>For each Django's view, test if the related HTTP request (GET, POST) is properly handled: accepts valid request, generates correct response, and properly modifies the database.<br>Given invalid requests (not supported request method, unauthorized operation, constraint violation), test if the views generate proper error messages. |
| | Session | F: Haozhe Wang<br>B: Jie Wang | 2/22 | 3/4 | |
| | Search | F: Haozhe Wang<br>B: Rui Zhang | 3/15 | 3/18 | |
| | Application | F: Yunjing Liu<br>B: Jie Wang | 3/5 | 3/15 | |
| | Feedback | F: Yunjing Liu<br>B: Rui Zhang | 4/3 | 4/13 | |
| Component Test | User Registration + Profile | Yunjing Liu | 2.22 | 3.4 | 1. Test component when input is empty<br>2. Test component when input data is invalid (including very short password)<br>3. Test component when input username is not unique<br>4. Test login part by input the wrong combination of password and username |
| | Session | Haozhe Wang | 3.15 | 4.3 | 1. Test component with empty input in different combination of fields required<br>2. Test component with wrong data type than required |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 3. Test component by trying to post tutoring session with overlapping times.<br>4. Test component by trying to delete a session<br>5. Test component by trying to edit a session |
| | Search | Rui Zhang | 3.15 | 4.3 | 1. Test component with input valid key words<br>2. Test component with input random combination of key words<br>3. Test component by trying to filter results by multiple constraints<br>4. Test component to check whether the system will automatically filter the unavailable session off for users. |
| | Application | Jie Wang | 4.3 | 4.10 | 1. Test component by sending an application to the owner of one session to see succeed or not.<br>2. As a tutor, test the component by trying to accept or decline one's application<br>3. As a tutor, test the component by trying to send the applicant a message back<br>4. Test the component by checking whether system will notify user the incoming message or not<br>5. Test the component by checking whether user will have a list of message where the unread ones will on top or not. |
| | Feedback | Jie Wang | 4.10 | 4.15 | 1. Test component by trying to add a feedback<br>2. Test component by checking whether tutees can see the rates of one session of the tutor at session detail page |
| System Test | Whole system | All team members | 4/14 | 4/15 | Use two android cellphones to work as a tutor and tutee. Test the functionality of register, login, create session, search, application and feedback consequently, using these two accounts. See if there is any incompatible features. |