# Review and Retrospective

Sprint 1 Review and Retrospective for Demo at SAL 322 on April 3rd 2018 6pm.

## Verbal feedback

The grader asked us to run the white-box tests. While we were running it, he asked us which features did we implement, and asked us to show white-box tests codes that handles those features that we implemented.

He also asked us to show the code coverage after we ran the white-box tests. Our coverage was a little bit above 80%. He said it should be over 90%, and that we should comment out or delete all the codes that we don’t actually use.

Then the grader asked us to run the black-box tests. We had some cucumber errors so we couldn’t get it running. The grader told us we have to run the black-box tests before we show him the actual features we implemented.

**General**

1. All the tests and features should be on one branch and be able to run on one machine
2. Tests need to be able to run, otherwise the grader won’t even let us demo the features
3. The grader asked for what features we want to demo and asks for black & white box tests that handle those features
4. The points we lost for this sprint, we can get them back in future sprints

**Black box**

1. More black box tests for input/collage options on Main.jsp  
   “I’m looking for a test for, for example, when I am at web page, then I should see an input box and collage options (i.e. what options there are) and save history button, placeholder text”

**White box**

1. More white box tests that test letter shape, filters, collage options, etc.
2. The coverage for whitebox should be higher than 80% (which was the coverage we had). Code we don’t actually use should be commented out.
3. Feature 3 should have the most whitebox tests

**Missing Requirement**

* HTTPS connection when logging in

## Written feedback

Grader comments for Sprint 1 Process Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Category** | **Comment** | **Deduction** |
| First Page | Professionalism | No headers or footers on the title page | -1 |
| Section 1.1-1.2 | Professionalism | Introduction does not address the purpose and scope of the software system | -1 |
| Section 2.2 | Planning | Missing justifications which include discussion of customer priorities, the need for always-working code, and the group's internal dynamics. Also, missing a description of the activities of the meeting | -2 |
| Section 3.3.1-3.3.5 | Scrums | Scrum 1-5 show only 5 people at scrum meeting but there are 6 responses and no explanation for the discrepancy | -2.5 |
| Section 4.1 | Pair Programming | Missing details of when did each pair do pair programming | -1.5 |
| Section 4.1 | Pair Programming | Missing specification of which of all commits belongs to which pair during its pair programming session (along with each session’s photo proof) | -2 |
| Section 4.1 | Pair Programming | Need to share /github repo with TA’s git account for checking commit history | 0 |
| Section 4.2 | Simple Design | Does not include designs for simple design | -3 |
| Section 4.4 | Develop tests first | Lacking screenshots or link to the screenshots to show you create test first | -2 |

## 

# **Grading Summary**

Professionalism: -2

Backlog: 2/2

Planning: 2/4

Scrums: 15.5/18

Pair Programming: 4.5/8

Simple Design: 1/4

Sustainable pace: 4/4

Develop tests first : 6/8

## Retrospective

A retrospective meeting was organized on April 4th, 2018 at JFF 333. All group members were present.

**Improvement Suggestions**

Ryan:

* I believe backend can be better organized and separated into different components. Right now server.java contains all the functionality from sending requests to the Google API, processing image filters, and adding borders. Server.java is a “God” class right now and it would really benefit our team in the future if we made separate classes for separate functionalities.

Tuling:

* I believe frontend code should only be pushed when all features are checked to exist and the code do not contain errors. The errors with frontend code dragged down the process a lot when we tried to connect front with back.

Yisa:

* I believe we should summarize the activities we did in each scrum meeting
* I believe we should add a short description of what we did by the end of each pair programming event

Peixuan:

* I believe we should write all the tests and get them working on one branch before we implement anything else
* I believe we should find a way to make sure Credentials.java and config.java don’t have to be added/modified everytime after we pull from github

Gong:

* We can focus on the connection between the front end and the back end before we start implementing the other features because if the connection is missing other features will not be seen by the stakeholders and thus useless.
* After the document is finished, we should verify it with instructors to make sure it contains all the required content.

Joann:

* I believe we can allocate more resources to testing because that is a key portion of the project. Our grader’s feedback showed us that we are missing tests. Thus, I think we need to spend more time and have more people writing tests.
* I believe we can merge branches more frequently to ensure that our master branch has everything we need for the program and for the tests

**Summary**

Our retrospective discussion highlighted that we need to spend more time on the planning phase for documentation, testing, and coding. With documentation, we need to make sure we prioritize creating the sprint backlog and pushing it to GitHub at the beginning of the sprint. Additionally, we need to consider how our customer’s priorities affect the order of the items in our backlog. By taking these actions, we as a team will have a clearer understanding of what needs to be accomplished during each sprint and the purpose behind each item.

Additionally, we need to spend more time focusing on the process of our sprints. This means ensuring we have written our tests and designed our system before beginning to implement features. Taking the extra time to do this will help prevent problems we experienced during this sprint, like our back-end and front-ends not fitting together and code containing errors being pushed. By investing more time in simple design, we can avoid the accidental creation of “God” classes. We will also have a more coherent understanding of how our code interacts and fits together.

In this next sprint, we will take care to make sure that we adhere to the correct process and that we invest sufficient amount of time in planning and testing. We will attend the office hours of the Professor, TAs, and CPs to review our documents and ask questions. We will focus on developing tests and designs before we start implementing. Lastly, we will spend more time discussing and documenting our discussions so that we can ensure we are all on the same page.