# DEADLY SCIENCE

Progress Report


Information Technology Capstone Project

COMP5703

Group Members

1. Ao Zhang (470496806)
2. Chen Zhuge (480101147)
3. Haitian Huang (480263272)
4. Yadi Wei (470233234)
5. Yating Yao (480100058)

# TABLE OF CONTENTS

# 1. PROGRESS STATUS

| | |
|---|---|
| **Project Name** | Deadly Science |
| **Project Start Date** | August 14, 2019 |
| **Project Manager** | Yating Yao |

| | |
|---|---|
| Project Description | This is an online platform that links tutors/mentors with students. This platform aims to achieve the maximum relationship flexibility among individuals offering/accepting educational content and minimum administrative burden. |

| | | |
|---|---|---|
| Project Status Report | #9 | Date: October 23, 2019 |

| **Status Item** | **Status up to last week** | **Planned for this week** |
|---|---|---|
| **Major deliverables** | Proceed with the last stage of testing and bug fixing. | Delivery and presentation. |
| **Planned delivery date** | October 30, 2019 | November 6, 2019 |
| **Major issues** | | |
| **Major risks** | | |
| **External dependencies** | | |
| **Estimated effort (hr)** | 18 | 18 |
| **Recorded effort (hr)** | 23 (21 individual + 2 group) | |
| **Status (R, Y, G)** | G | G |

## 2. ROLES & RESPONSIBILITIES

**Technical support:**

1. System administration. Uploaded the project into AWS cloud server, and made it public accessible online.
2. Continued to refine the website's functionalities. Added several improvement features.
3. Started to code the offline assistance tool using Python, which would help the root user to manage the platform.
4. Database management. Regenerated the content of the entire database, using the script I wrote, to get rid of the corrupted data entries caused by the bugs in the previous version's code base.

## 3. INDIVIDUAL ACHIEVEMENTS

There were four things (5 + 10 + 5 + 1 = 21 hours) I have done this week:

1. (5 hours) System administration. After a serial of troubleshooting, I eventually made the website publicly accessible with a public IP address. This IP address, along with three testing accounts, were then sent to the customers for testing.

Using the remote SSH connection, I pulled the code base from University of Sydney's GitHub to our AWS cloud server based in Sydney, NSW. Then I tried to make the code run in the remote server. During this process, I encountered a severe compatibility issue: a specific node.js library named sass-loader (a front-end toolkit) did not work properly in Linux environment. As the result, the home page is blank without any visible content, while all libraries worked fine in Windows environment.

The technical team from the Innovation Center of University of Sydney provided two feasible approaches to resolve this issue:

A. Instead of Linux environment, deploy the project in Windows environment, as there are also many decent and reliable cloud instance options running the Windows operating system.

B. Figure out the reason behind the sass-loader library's compatibility issue and fix it, continuing to using the current Linux environment.

After the consultation in the Innovation Center, I decided to dig into the sass-loader library's compatibility issue at first (the approach B above), and switch to Windows environment only if this issue can not be solved in Linux environment, since I have experience in AWS Linux environment deployment and I may possibly encounter other troublesome issues if I attempt to deploy the code base in Windows.

With four hours' information collecting, analyzing and testing, I eventually found the correct way to address this compatibility issue [1], and successfully updated the local environments in AWS cloud server to a compatible condition. After confirmed the healthy status of the web server, I compressed the local environment into zip file and transferred it to my local computer, then I uploaded this zip file to the team's shared Google Drive.

A few days' later, the front-end team have introduced another two front-end libraries, and I did the corresponding system administration - updated the two libraries in the AWS cloud server and restarted the server, then compressed the local library environment and saved it to Google Drive.

As the result, our server has been running for more than five days and it is in a healthy condition along with the MongoDB server with resides in the same instance.

2. (10 hours) I Continued to refine the functionalities of the project, most of which are minor improvements. A comprehensive list of this refinements are listed below [2]-[27]:

I updated the name display of the users. I appended the account name or unikey with the user's name, to make the user more identiable in the admin's approval panel and user's notification center. The previous version of the website can not distinguish two users with the same name. In the future, we would further improve the usability by making every user name on the website clickable, which then leads to the user's information pop-up window. In that condition, the current name-displaying approach would be rolled back to the initial pure name strategy to create a clear user interface.

I participated a major usability improvement in admin panel - tutorial appointments. Collaborating with another back-end team member, we added another important field into the tutorial appointments: the tutorial's start time's distance from current now (e.g. 3 days 20 hours 11 minutes). This feature could greatly help the admin to make the decision, since our team have identified a notable issue that not all the tutorial appointments have sufficient time-gap/time-buffer to be carried out after the admin's approval. In worse cases, some of the tutorial appointments' start times are already past when the admin begins to consider its approval. Hence, we decided to calculate the time distance every time the admin requests a new batch of tutorial appointment applications, in order to help the admin make reasonable decisions (i.e. declining the tutorial appointments that does not have sufficient reaction time in real world for the students and tutors).

I participated in fixing the bug of timezone difference. The MongoDB database system persists to store all the time points in UTC format, which has 11 hours difference with the Eastern Australia Timezone. With two hours' coding and testing, an other back-end team member and I together addressed all the time stamp displaying and calculation issues that were prone to the timezone different. Now the website has a flawless time displaying and calculation experience, for both the front-

end developers and the website's real users. In their use cases, they don't have to concern about the timezone issue. In other words, we made this issue 'transparent' to them via re-programming the back-end side codes.

I refined the mentor matching algorithm. In previous versions of the project, the mentor matching algorithm was not mature enough, in which we could not guarantee the matching result was one of the matches among the best mentor candidates. I have refined the algorithm to fully consider the students inputs (faculty preference, campus preference and personal statement) and search the entire mentor candidate database to filter the best ones out. Along with the trustworthy candidate, the current algorithm also gives out the reason why the system did this recommendation, which could help the admin in his/her final decision.

3. (5 hours) Started to implement the offline assistance tool for the root user.

The aim of the offline assistance tool is to help the root user to proceed the daily administration in a secure way. I coded with Python and its Tkinter library, with the assistance of Python MongoDB driver. The major functionalities of this tool includes:

1. User search via user name.
2. User tag granting/removing.
3. Report generation (individual report or overall report).
4. Abnormal tutorial appointments archive.
5. Tutor/mentor removal (useful in the end of one semester).

Points 1 and 2 of the above list have been finished with graphical user interface. The rest points are realized using command line prompts, and will be carried out with graphical user interface in next week.

4. (1 hour) Database management. In this week, I erased and regenerated the entire database content. This work was carried out via the automatic script I wrote last week. This data re-generation was to erase the corrupted data samples which were produced by inferior codes in the previous versions of the project. After this time's data re-generation, I picked out three testing accounts that are able to be logged in on the public accessible IP address. These three testing accounts has different previledges in the system, and were sent to the customer for the testing.

## 4. GROUP COLLABORATION

I attended and participated the following three group activities (1 + 1 = 2hours) in this week:

1. (1 hour) Customer meeting on October 21, 15:30 - 16:30, in Innovation Center, Level 3, USYD.

In this meeting we introduced the current status of the project to our customers, and got their valuable feedbacks. The technical team in the center helped us in the compatibility issues and user interface optimization.

2. (1 hour) Team meeting on October 24, 15:30 - 16:00, in ABS Building Level 2, USYD.

The team got together and merged individual's code, then we tested and verified the functionalities.

## REFERENCES


[1] https://github.com/sass/node-sass/issues/2536

[2] https://stackoverflow.com/questions/2647867/how-to-determine-if-variable-is-undefined-or-null

[3] https://www.reddit.com/r/javascript/comments/80whdr/addeventlistener_is_not_a_function/

[4] https://stackoverflow.com/questions/4421207/how-to-get-the-last-n-records-in-mongodb

[5] https://flaviocopes.com/file-upload-using-ajax/

[6] https://stackoverflow.com/questions/41497674/access-control-allow-origin-issue-when-api-call-made-from-react-isomorphic-ap

[7] https://stackoverflow.com/questions/43262121/trying-to-use-fetch-and-pass-in-mode-no-cors

[8] https://stackoverflow.com/questions/22087076/how-to-make-a-simple-image-upload-using-javascript-html

[9] https://stackoverflow.com/questions/4796914/store-images-in-a-mongodb-database

[10] https://stackoverflow.com/questions/3794919/replace-all-spaces-in-a-string-with

[11] https://stackoverflow.com/questions/3794919/replace-all-spaces-in-a-string-with

[12] https://github.com/babel/babel/issues/8599

[13] https://github.com/webpack/webpack/issues/8082

[14] https://github.com/webpack/webpack-cli/issues/436

[15] https://stackoverflow.com/questions/38788166/webpack-command-not-working

[16] https://stackoverflow.com/questions/46541371/npm-err-refusing-to-delete-code-eexist/46541546

[17] https://github.com/sass/node-sass/issues/2536

[18] https://stackoverflow.com/questions/39404267/module-not-found-error-cant-resolve-react-bootstrap-validation

[19] https://stackoverflow.com/questions/53914013/failed-to-compile-module-not-found-cant-resolve-react-router-dom

[20] https://stackoverflow.com/questions/34819473/error-couldnt-find-preset-es2015-relative-to-directory-users-username

[21] https://stackoverflow.com/questions/44751216/module-build-failed-error-couldnt-find-preset-stage-0-relative-to-directory

[22] https://stackoverflow.com/questions/50100430/referenceerror-unknown-plugin-transform-runtime-specified-in-package-json-at

[23] https://stackoverflow.com/questions/22137564/node-how-to-run-app-js

[24] https://serverfault.com/questions/264595/can-scp-copy-directories-recursively

[25] https://stackoverflow.com/questions/6558080/scp-secure-copy-to-ec2-instance-without-password

[26] https://github.com/babel/babel/issues/8599

[27] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/toLocaleString