COMPS350F Software Engineering

Testing Plan Report

Group 15 members:

Raja Arbab Mahmood (13275291)

Rejul Rai (13347518)

Wong Ngai Shing (12920028)

Ting Kenneth Charles (13338614)

Situ Tingzhen (13347101)

1. Introduction

The objective of this document is to provide the testing results of the functionality of the project.

The current status of the program fulfills most of the client requirements, with few still in the implementation process.

Client requirement

Functions	Status
Teachers/admin	
Student ID arranged in order	✔
Enter staff no. before updating record	✘
Update students' academic record	✔
Tutorial class division in a course	✘ *
Ranking of students based on scores/grading	✘ **
Accessing student's assessment results and grading	✔
Attendance (access and edit)	✔
Search bar/filter	✔
Create announcement	✔
Login / Logout	✔
Students	
Read own academic record	✔
Update personal information	✔
Authorization before accessing the academic record	✔
Access announcements	✔
Login / Logout	✔

Note:

* Similar function of add/drop student(s) from a course is implemented for admin staff

** Similar function of showing the highest grading and OCAS score is implemented for both teachers and admin

The report consists of two parts: validation testing and defect testing. Validation testing is to determine whether the program can comply with the requirements, which the testing objects include mainly the aforementioned functionalities. Defect testing is to figure out if the program runs with potential anomalies and the severity of issues to the system's overall performance under certain circumstances.

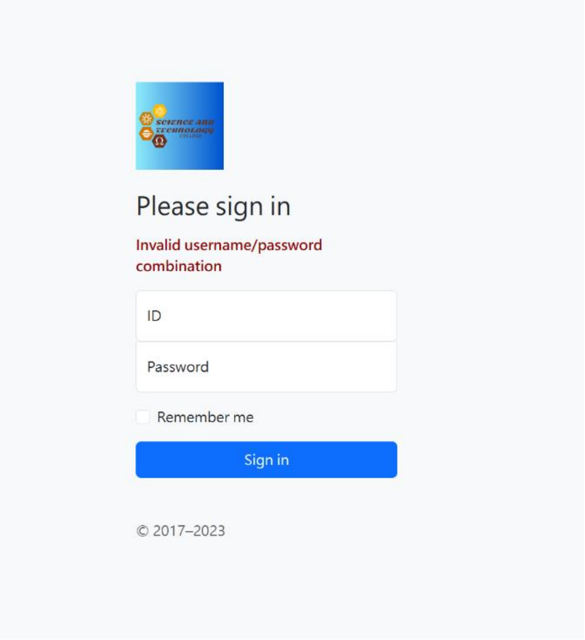
2. Validation Test

1. Login/Logout function

The login/logout determines if a guest is a registered user of the student record system. If the user is verified successfully, it will lead to webpages according to the user group (students, staff [teachers/admin]). Otherwise, an error message will be displayed and require a new login attempt for the user.

The following are the test cases as to whether the system can handle any input for login:

Test no.	Description	Result
1	Valid ID and password input (student)	Session created, redirect to student page
2	Valid ID and password input (teacher)	Session created, redirect to teacher page
3	Valid ID and password input (admin)	Session created, redirect to admin page
4	Invalid combination of ID and password input	Login page refreshed, error message displayed



Please sign in

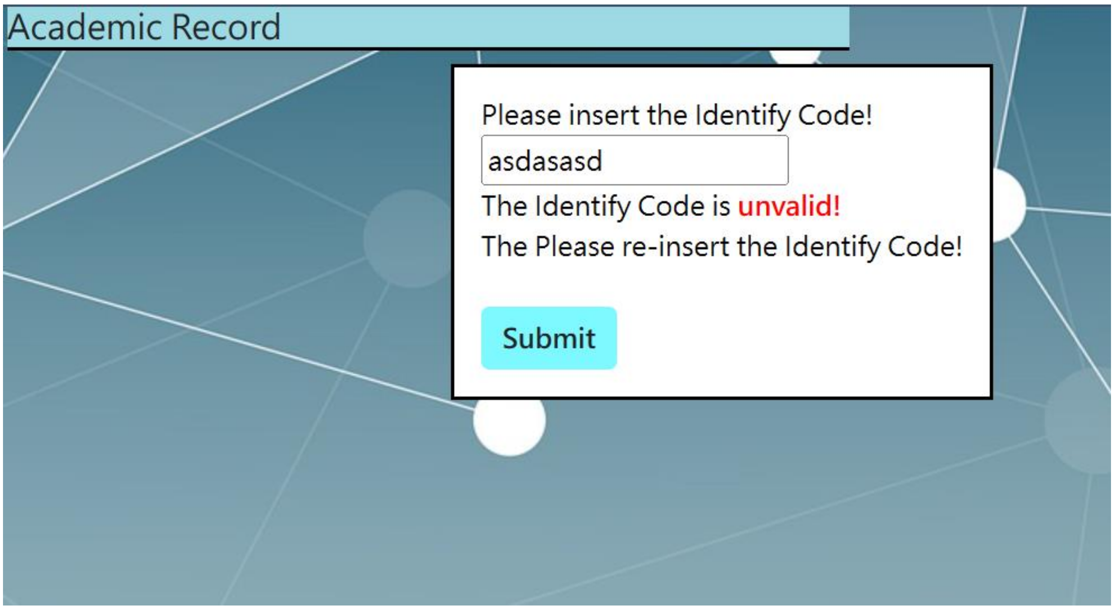
Invalid username/password combination

Figure 1: Screenshot of login page after unsuccessful login attempt

2. Accessing academic record for students

One of the inputs where anomalies would likely occur is when entering a Student ID before accessing his/her academic record. The following are the test cases of different input in the text field as well as after pressing "submit" button:

Test no.	Description	Result
1	Valid student ID and button pressed	Popup window for entering student ID disappeared, term list displayed
2	Invalid student ID and button pressed	Error message displayed
3	Inputting more than 8 characters	Input limited, cannot exceed more than 8 characters


Academic Record

Please insert the Identify Code!

asdasasd

The Identify Code is unvalid!

The Please re-insert the Identify Code!

Figure 2: Screenshot of invalid input of student ID when attempting to access academic record

3. Edit personal information for students

Another high anomaly possibility is editing personal information for students. Certain constraints are implemented to prevent irregularities. For instance, no special characters for student names and more than 8 digits for phone numbers. Both are required to input which the user cannot update with a null value.

Below are the test cases for editing personal information:

Test no.	Description	Result
1	Valid format of name and phone number	New input being updated to the database, profile page refreshed to display changes

2	Invalid format of name and phone number	New input cannot send to the server, prompting the user to input with the correct format
3	Null value of name and/or phone number	

4. Searching for students in academic records (teachers)

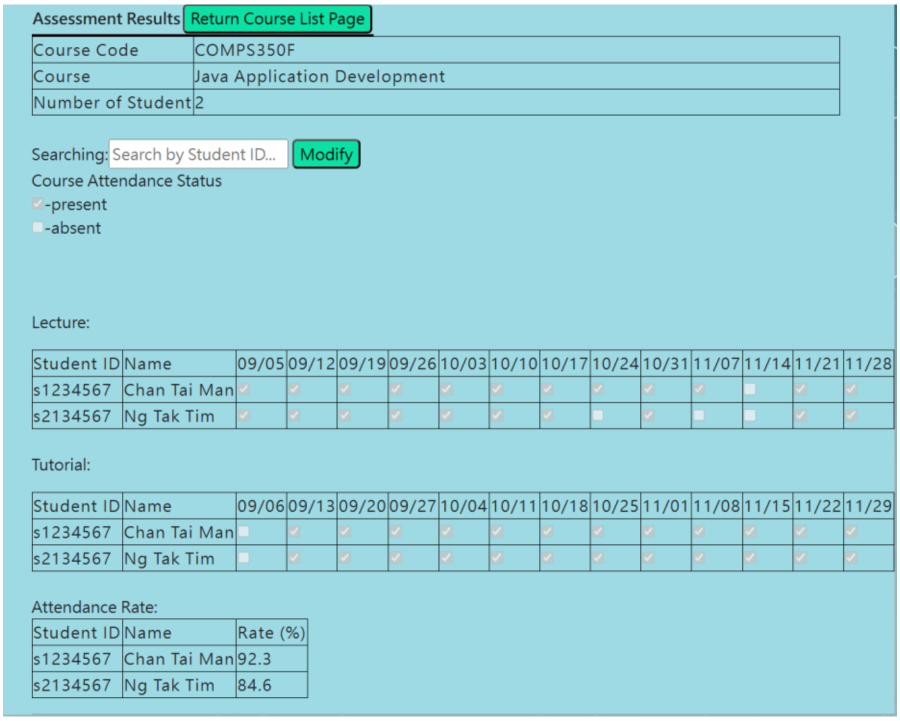
Only students which the student ID matches (fully or partially) in the search bar will be shown in the table.

The following are the test cases for searching students:

Test no.	Description	Result
1	Fully input a student's student ID	Specified student displayed
2	Partially input a student's student ID	Students with matching patterns with the value in the search bar displayed
3	Input does not match any student's student ID in the course	No student displayed in the table
4	A null value in the search bar	All students in the course displayed

5. Academic records displayed

Students academic record are correctly displayed according to particular course selected.



Course Code	COMPS350F
Course	Java Application Development
Number of Student	2

Searching: Search by Student ID...

Course Attendance Status

☑-present

□-absent

Lecture:

Student ID	Name	09/05	09/12	09/19	09/26	10/03	10/10	10/17	10/24	10/31	11/07	11/14	11/21	11/28
s1234567	Chan Tai Man	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	□	☑	☑
s2134567	Ng Tak Tim	☑	☑	☑	☑	☑	☑	☑	□	☑	□	□	☑	☑

Tutorial:

Student ID	Name	09/06	09/13	09/20	09/27	10/04	10/11	10/18	10/25	11/01	11/08	11/15	11/22	11/29
s1234567	Chan Tai Man	□	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
s2134567	Ng Tak Tim	□	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Attendance Rate:

Student ID	Name	Rate (%)
s1234567	Chan Tai Man	92.3
s2134567	Ng Tak Tim	84.6

Figure 3: Screenshot of the course assessment results of COMPS350F Java Application Development

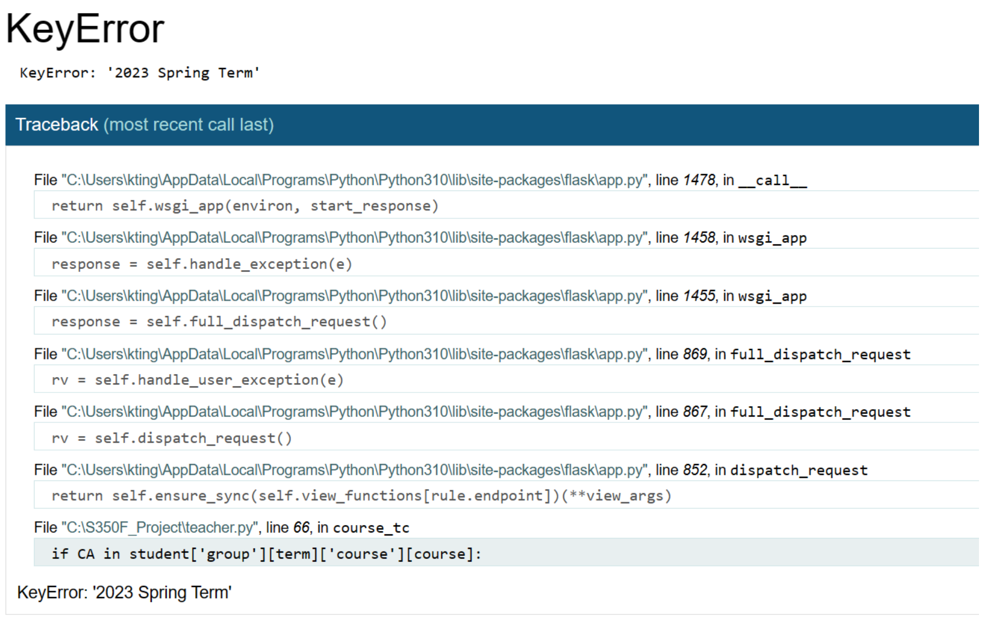
3. Defect Testing

During our validation tests, we have encountered the following issues which will be listed according to the severity of the defects to the overall performance of the system:

1. Inaccessible course data for teachers and administrators

The system cannot retrieve the data of that course anymore due to KeyError while trying to access the database. The issue is currently affecting teachers and administrators, whereas students are unaffected.

Note that only details of courses are not accessible, while other data and functions including course code modification, access and edit of attendance records as well as assessment scores, are still available.



```
KeyError
KeyError: '2023 Spring Term'

Traceback (most recent call last)

File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1478, in __call__
  return self.wsgi_app(environ, start_response)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1458, in wsgi_app
  response = self.handle_exception(e)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1455, in wsgi_app
  response = self.full_dispatch_request()
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 869, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 867, in full_dispatch_request
  rv = self.dispatch_request()
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 852, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "C:\S350F_Project\teacher.py", line 66, in course_tc
  if CA in student['group'][term]['course'][course]:
KeyError: '2023 Spring Term'
```

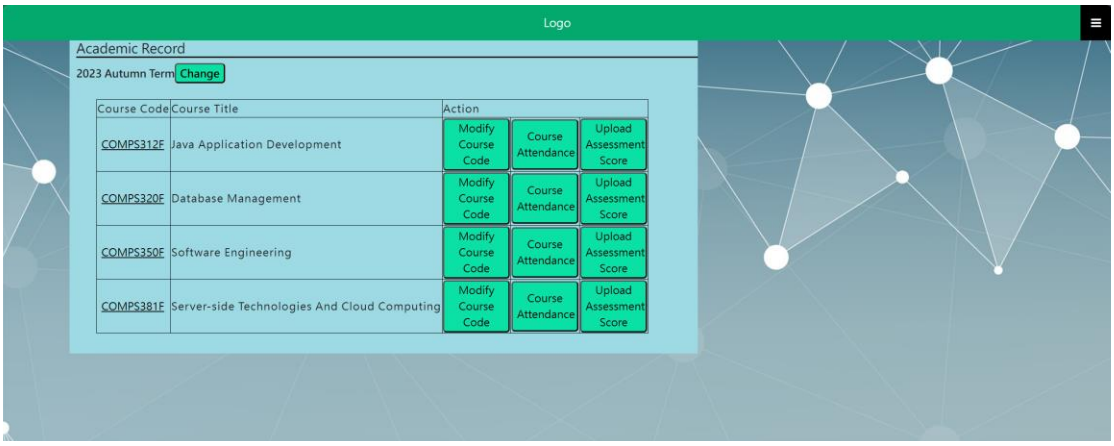
Figure 4: Debug interface of failed retrieval of course data for teachers and admins due to KeyError

2. System cannot prevent conflict of course code upon user edit

Users can edit the course code of a particular course, which also includes codes that are the same as the existing courses. As of the current stage, preventive measures are not being implemented yet, which may cause certain courses not to be accessible, including their academic and attendance records, as well as relevant announcements.

3. Page layout anomalies in course list for teachers and admins

It was discovered that the page layout for a course list of a specific term would be displayed under the sidebar. Although a temporary fix was implemented by using a navigation bar originally for screens with a vertical ratio, such will lead to difficulties for users when using the system.



Logo

Academic Record

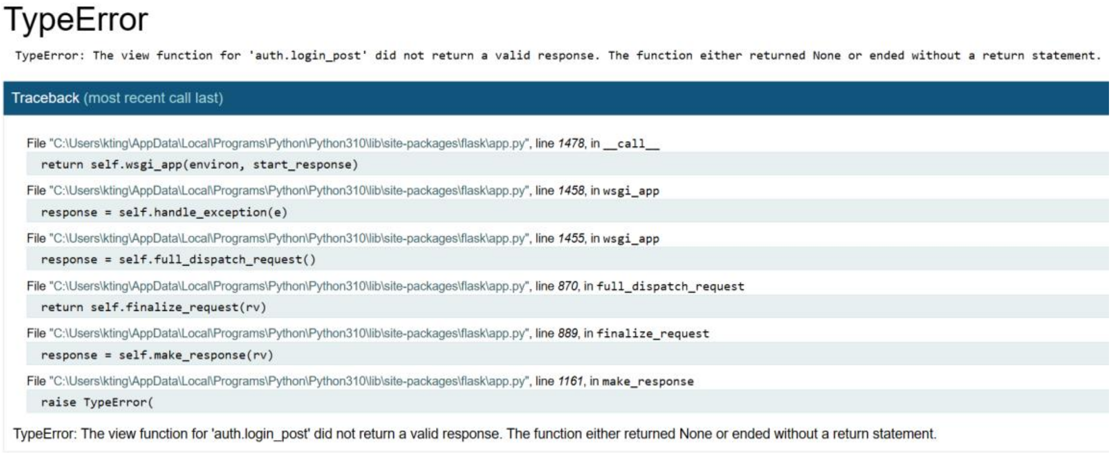
2023 Autumn Term Change

Course Code	Course Title	Action		
COMPS312F	Java Application Development	Modify Course Code	Course Attendance	Upload Assessment Score
COMPS320F	Database Management	Modify Course Code	Course Attendance	Upload Assessment Score
COMPS350F	Software Engineering	Modify Course Code	Course Attendance	Upload Assessment Score
COMPS381F	Server-side Technologies And Cloud Computing	Modify Course Code	Course Attendance	Upload Assessment Score

Figure 5: Affected page of course list for teachers and admins

4. Possible crashes for inputting wrong passwords in login page

It was tested that when inputting wrong passwords, typically in numeric or other users' passwords, a TypeError was displayed, indicating a possible lack of handling for such behavior.



```
TypeError

TypeError: The view function for 'auth.login_post' did not return a valid response. The function either returned None or ended without a return statement.

Traceback (most recent call last)

File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1478, in __call__
  return self.wsgi_app(environ, start_response)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1458, in wsgi_app
  response = self.handle_exception(e)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1455, in wsgi_app
  response = self.full_dispatch_request()
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 870, in full_dispatch_request
  return self.finalize_request(rv)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 889, in finalize_request
  response = self.make_response(rv)
File "C:\Users\kting\AppData\Local\Programs\Python\Python310\lib\site-packages\flask\app.py", line 1161, in make_response
  raise TypeError(

TypeError: The view function for 'auth.login_post' did not return a valid response. The function either returned None or ended without a return statement.
```

Figure 6: Debug interface of TypeError after inputting wrong passwords in login page

4. Development Test

a) unit test

- updating personal information for students

- creating announcement

- accessing student assessment result and grading

b) component test

- tutorial class division in a course

c) system test

- login and logout function

- accessing and editing the attendance

- search bar/filter function

-accessing announcements for students

5. Conclusion

In conclusion, the validation and defect testing procedure uncovered a number of areas for improvement and implementation. The program can be ready to use after fixing the detected flaws and resolving the issue.

While validation tests show that the program can run as most requirements specified, defect tests also show the potential anomalies the program might run. The tests suggest that further debugging and improvements are needed to fulfill the client's requirements.