**COMPS350F**

**Software Engineering**

**(Autumn Term 2023)**

# **Student Record Management System**

# **Testing Report**

**Group 11**

| **Member** | |
| --- | --- |
| **Name** | **Student ID** |
| **Tse Kai Chun** | **12755669** |
| **Mak Ho Ying** | **12755670** |
| **Chow Hong Nin** | **13032510** |
| **Lo Pak Sum** | **13091902** |
| **Wong Ka Ho** | **13251975** |

## Table of Content

[**Student Record Management System 1**](#_i0dl6zcgactk)

[**Testing Report 1**](#_1ra66k2qku5u)

[Table of Content 2](#_rq2s33boenar)

[Introduction 3](#_eva7nyz4gjia)

[1. Background 3](#_82b7dkilehv1)

[2. Purpose 3](#_jl98t17lmy18)

[3. Scope 3](#_fhdfu84fibqm)

[Testing Overview 4](#_v60t19edzpsy)

[Validation Test 5](#_jatq752el9i2)

[Objective 5](#_w9g8odel2rej)

[Test Cases and Procedures 7](#_j2ah3wu7lt55)

[Defect Testing 25](#_necvf8cfdx8g)

[Objective 25](#_1szqi6a30q74)

[Test Strategy 25](#_o3hynxnm0lqg)

[Test Cases and Procedures 26](#_ppwf0wd5iqy1)

[Conclusion 41](#_7tcwfoe0thnb)

[Appendix(if we have) 42](#_h0toe5qbzcst)

## Introduction

### Background

The School of Science and Technology is a part of HKMU that aims to advance learning and knowledge through teaching and research. Now they require a system to manage student records. The main purpose of this system is to manage and record student academic records and student personal information. This requirement documentation is to document the requirement for the development of the system.

### Purpose

The student record management system is designed to cater to the needs of students, teachers, and administrators within the School of Science and Technology. It will provide teachers with the ability to access and update student academic records, while students will have secure access to their academic records and the capability to view and update their personal information. Additionally, the system will incorporate an administrative component, granting specific privileges to authorized personnel.

By implementing this system, the School of Science and Technology aims to streamline administrative processes, improve data accuracy, and enhance the overall student experience. Efficient management of student records will contribute to a more effective teaching and learning environment, ensuring that all stakeholders have access to up-to-date and reliable information.

### Scope

The Student Record Management System is a web-based application built on the Node.js platform and uses the NoSQL database management system. Node.js, with its event-driven, non-blocking I/O model, enables efficient handling of concurrent requests and ensures high performance for data processing and retrieval. It seamlessly integrates with NoSQL databases, allowing for smooth interaction and leveraging the advantages of a NoSQL DBMS. Additionally, Node.js offers a vast ecosystem of modules and libraries, facilitating rapid development, easy scalability, and cross-platform compatibility for your Student Record Management System.

## 

## Testing Overview

The Validation Test aims to demonstrate that the student academic record system meets its requirements, while defect testing focuses on discovering faults or defects in the software. The validation test ensures the system operates as intended and provides accurate results, while Defect Testing aims to identify any issues or inconsistencies. Both types of testing are essential for ensuring the reliability and functionality of the system.

## Validation Test

### Objective

Validation testing is a type of testing that focuses on the user-visible aspects of the software, including inputs, outputs, and actions. Its primary objective is to verify that these user-visible portions meet the specified software requirements. This type of testing ensures that the software behaves as intended from a user's perspective.

During validation testing, various techniques and methods are employed to examine the user-visible components of the software. This may involve conducting functional tests to validate the expected behavior of the system, performing user acceptance tests to ensure that the software meets user expectations, and conducting usability tests to assess the ease of use and user-friendliness of the software.

In addition to testing the software itself, validation testing also involves examining the documentation associated with the software. This includes user manuals, help guides, and any other relevant documentation. The purpose is to ensure that the documentation aligns with the software's functionality and accurately reflects how to use the software.

By conducting thorough validation testing, software development teams can gain confidence that the software meets the intended requirements and functions correctly from a user's perspective. It helps identify any discrepancies or deviations from the expected behavior and provides an opportunity to address them before the software is released to end-users.

Test Strategy

**Requirements Analysis:**

Thoroughly analyze the software requirements to gain a clear understanding of the expected behavior and functionality. Identify any ambiguities or gaps in the requirements that need to be clarified before testing. This will ensure that the validation testing is aligned with the intended purpose of the software.

**Test Coverage:**

Define the scope and coverage criteria for validation testing. Determine which aspects of the software will be tested, such as specific features, modules, or user workflows. Ensure that the selected test cases cover a wide range of scenarios and adequately represent the intended usage of the software. This will help ensure comprehensive coverage of the software's functionality.

**Test Data Preparation:**

Prepare the necessary test data to execute the validation test cases. This may involve creating realistic and representative data sets that cover different scenarios and edge cases. The test data should be designed to thoroughly exercise the software's functionality and validate its behavior in various scenarios.

**Test Oracles:**

Define the expected outcomes and criteria for determining whether the software's behavior aligns with the requirements. Establish clear criteria for pass/fail decisions and ensure that the test oracles are well-defined and documented. This will provide a basis for evaluating the software's behavior and determining the success or failure of the validation tests.

### Test Cases and Procedures

1. **User Authentication and Authorization:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC001 | Verify that teachers can log in with valid credentials and access the system. |  | 1. Go to the Login Page  2. Input userId  3. Input password  4. Click the sign-in button | userId: teacher1  password: TeacherPass!123 | Login successfully and jump to the homepage | Login successfully and jump to the homepage | Pass |
|  | | | | | | |
| TC002 | Verify that students can log in with valid credentials and access their records. |  | 1. Go to the Login Page  2. Input userId  3. Input password  4. Click the sign-in button | userId: s123456  password: pT3kD1Ri | Login successfully and jump to the homepage | Login successfully and jump to the homepage | Pass |
|  | | | | | | |
| TC003 | Verify that administrators can log in with valid credentials and have full access to all student records. |  | 1. Go to the Login Page  2. Input userId  3. Input password  4. Click the sign-in button | userId:  admin1  password: admin123 | Login successfully and jump to the homepage | Login successfully and jump to the homepage | Pass |
|  | | | | | | |
| TC004 | Verify that unauthorized users are unable to log in and access the system. |  | 1. Go to the Login Page  2. Input userId  3. Input password  4. Click the sign-in button | userId:  whoiam  password: whoami | Login failed due to wrong information provided by users | Login failed due to wrong information provided by users | Pass |
|  | | | | | | |

1. **Personal Information Update:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC005 | Verify that users can update their contact details and emergency contact information and the changes are reflected in the system. | Login | 1. Go to the Profile  2. Press Edit Button  3. Input updated info  4. Click the update button | Email:  jsmith@school.com  Phone Number:  735-560-8888  Emergency Phone Number:  888-806-5537 | Update Personal Info Success and email, phone number, emergency phone number are changed. | Update Personal Info Success and email, phone number, emergency phone number are changed. | Pass |
|  | | | | | | |

1. **Academic Record Management:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC006 | Verify that teachers can search and retrieve student records based on different criteria such as student ID or name. | Login to teacher role account | 1. Go to Dashboard  2. Click Detail Button  3. Press Student Records Button | None | Teacher are allowed to retrieve student records | Teacher are allowed to retrieve student records | Pass |
|  | | | | | | |
| TC007 | Verify that teachers can update student academic records | Login to teacher role account | 1. Go to Dashboard  2. Click Detail Button  3. Press  Student Records Button  4. Press Edit button  5. Choose Final Grade and Submit | Final Grade:  In Process | update student academic records success | update student academic records success | Pass |
|  |  | | | | | | |
| TC008 | Verify that students can view their academic records | Login to teacher role account | 1. Go to Academic Page | None | Student can see their academic records | Student can see their academic records | Pass |
|  |  | | | | | | |
| TC009 | Verify that administrators can update academic records | Login to teacher role account | 1. Go to Dashboard  2. Click Detail Button  3. Press Student Records Button  4. Press Edit button  5. Choose Final Grade and Submit | Final Grade: A+ | Admins can update academic records | Admins update academic records successfully | Pass |
|  |  | | | | | | |

1. **Course Management:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC010 | Verify that Administrators can add/edit programs they are teaching. | Login to the admin account | 1. Go to Dashboard  2. Click Program Management Button  3. Click Add Button/Details Button(Edit)  4. Input/Update Program Info and submit | ID: C133 Name: Test1 Intro: test1 | Add Program Success/  Edit ProgramSu-ccess | Add Program Success/  Edit ProgramSu-ccess | Pass |
|  | | | | | | |
| TC011 | Verify that Administrators add courses under their program. | Login to the admin account | 1. Go to Couse Page  2. Click Add button  3.Input Course Info and Submit | ID: C1305  Name: TEST  Intro: TEST  credit: 3 | Success and show in Course Table | Success and show in Course Table | Pass |
|  |  | | | | | | |
| TC012 | Verify that Administrators/Teachers can modify course information | Login to the admin /teacher account | 1. Go to Couse Page  2. Click C001 ID  3. Click Edit Button  4. Update info and Submit | Name: Introduction to Programming 2  Credit: 4 | Course Edit Success | Course Edit Success |  |
|  |  | | | | | | |

1. **Course Statistics Report:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC013 | Verify that teachers can generate a report about the statistics of the course they are assigned to. | Login to the teacher account | 1. Go to Dashboard  2. Click Course Detail Button  3. Click Student  Records Button | None | Student Status Table should be generated | Student Status Table generated | Pass |
|  | | | | | | |

1. **Administrator Privileges:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC014A | Verify that administrators can create a user account | Login to the admin account | 1. Go to Dashboard  2. Click User Management Button  3. Click Add Button  4. Input User Info and submit | userId:  s789012  Password:  17722929  Role:  student  Name:  Peter Peter  Major:  P005  Program year:3  Date from:  2021-09-01  Date to:  2024-08-30  Email:  peter@gmail.com  Phone number:  12345678  Emergency Phone number:  12345678 | Add user Success | Add user Success | Pass |
|  |  | | | | | | |
| TC014B | Verify that administrators can edit user account | Login to the admin account | 1. Go to Dashboard  2. Click User Management Button  3. Click Details Button  4. Click Edit  Button  5. Input User Info and submit | Phone Number:  45678900 | Edit user information Success | Edit user information Success | Pass |
|  |  | | | | | | |
| TC015 | Verify that administrators can assign roles and permissions to users. | Login to the admin account | 1. Go to Dashboard  2. Click User Management Button  3. Click Add Button  4. Select the role of the user | None | Admin can assign student, teacher and admin role for user | Admin can assign student, teacher and admin role for user | Pass |
|  |  | | | | | | |
| TC016 | Verify that administrators can access all system functionalities and student records. | Login to the admin account | 1. Go to Dashboard  2. Select user management with/without detail | None | Admin can achieve user management page | Admin can achieve user management page | Pass |
|  |  | | | | | | |
| TC017 | Verify that administrators can access all system functionalities and student records. | Login to the admin account | 1. Go to Dashboard  2. Select program management with/without detail | None | Admin can achieve program management page | Admin can achieve program management page | Pass |
|  |  | | | | | | |

## 

## Defect Testing

### Objective

Defect testing aims to identify and uncover faults or defects in the software. The primary goal is to discover any discrepancies or inconsistencies between the actual behavior of the software and its expected behavior as defined by the requirements or specifications. Defect testing aims to expose issues that may impact the functionality, reliability, performance, or usability of the software.

### Test Strategy

**Defect Identification Techniques:**

Specify the techniques and methods to be used for identifying defects. This may include techniques such as boundary value analysis, equivalence partitioning, exploratory testing, and error guessing.

**Test Environment Setup:**

Define the necessary test environment, including hardware, software, and network configurations, to effectively execute defect testing. Ensure that the test environment closely resembles the production environment to simulate real-world conditions.

**Test Data Preparation:**

Determine the test data required to expose potential defects. This may involve creating test data sets that cover various scenarios, including both valid and invalid inputs, edge cases, and boundary conditions.

### Test Cases and Procedures

1. **User Authentication and Authorization:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC001 | Attempt to login with an invalid username and password combination. Verify that the system displays an appropriate error message and does not grant access. |  | 1. Input invalid ID  2. Input invalid Password | Valid userId: teacher1  Valid  Password: TeacherPass!123  Fake userID: haha  Fake password:nice  Combination:  1.Valid ID, Wrong Password  2. Invalid ID, Valid Password  3. Both Invalid | Login Failed | Login Failed | Pass |
|  | | | | | | |
| TC002 | Login as a student and verify that the system only allows read-only access to the student's records. Attempt to modify the records and ensure that the system restricts the user from making changes. | Login | 1. Log in to System  2. Force to jump admin edit record page i.e.  http://localhost:3000/course/C001/2023/A/s123456 | None | Student Cannot edit his records | Student Cannot edit his records | Pass |
|  |  | | | | | | |
| TC003 | Login as an administrator and verify that the system grants full access to all student records. Attempt to modify student records to ensure that the system allows these actions. |  | 1. Login admin account  2. Go to Dashboard  3. Go to User Management | None | Admins Can modify Student Records and access all student records | Admins Can modify Student Records and access all student records | Pass |
|  |  | | | | | | |

1. **Personal Information Update:**

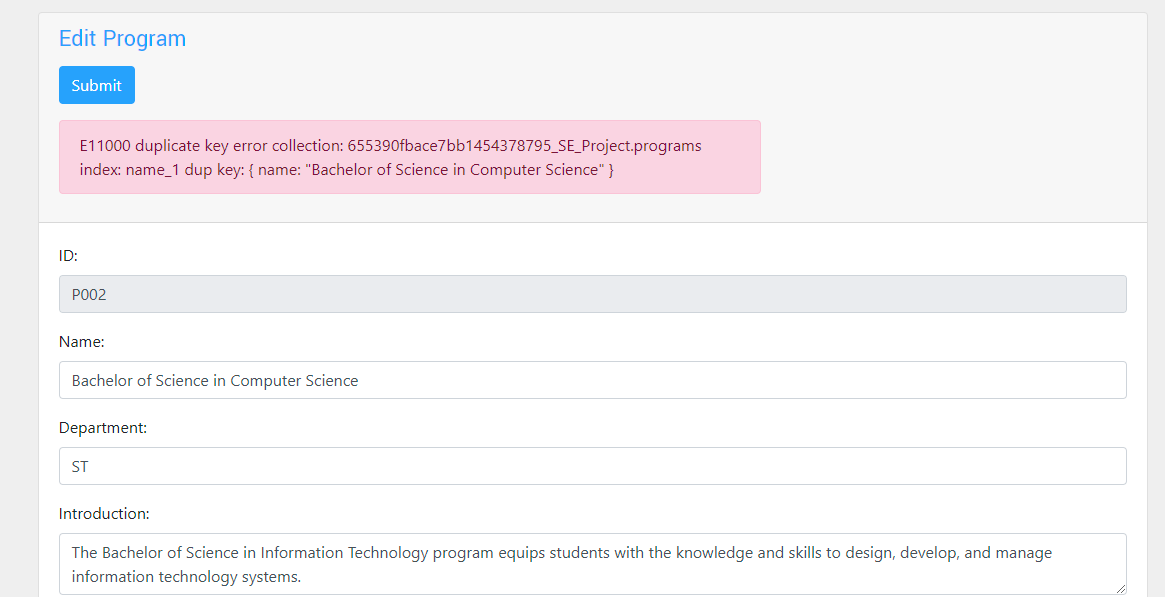
| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC004 | Update personal information as a student and verify that the changes are reflected in the system and visible to authorized users. | Login | 1. Login student account  2. Edit personal info  3. Login admin account  4. Go to dashboard  5. Click into User Management | **edit data account:**  userId: s123456  Password: pT3kD1Ri  edit email:  jsmith@wwwschool.com  **Check updated account:**  userId:  admin1  Password:  admin123 | John Smith Update Email, admin can see it was updated | John Smith Update Email, admin see it was updated | Pass |
| **Student Pov before Update:**    **admin pov:** | | | | | | |
| TC005 | Attempt to update personal information without providing required fields. Verify that the system displays appropriate error messages and prevents the update from being saved. | Login | 1. Login student account  2. Edit personal info in wrong type  3. Click Update button | Email:jsmith  Phone number:abc  Emergency Phone number:abc | Student can’t update due to wrong type of data | Student can update due to wrong type of data | fail  (need to restrict the type of data when user input) |
|  |  | | | | | | |

1. **Academic Record Management:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC006 | Add student using different criteria (e.g., name, student ID) and verify that the student was registered in course | Login to teacher/admin account | 1. Go to Dashboard  2. Click Details Button of Course  3. Add student by searching id or name  4. Click Add button | Search s123456 to add student to the course | Students should be added to Student Records and status should be in progress. | Students fail to be added to Student Records | Fail |
|  | | | | | | |
| TC007 | Update student academic records as a teacher or administrator and verify that the changes are saved correctly. | Login to teacher/admin account | 1. Login teacher account  2. Go to dashboard  3. Choose a course details button  4. Click student records button  5. Edit Student Grade and Submit it  3. Login student account  4. Go to academic | **Edit Records account:**  userId: teacher1  Password: TeacherPass!123  **Check Records account:**  userId: s123456  Password: pT3kD1Ri | Original Grade is “in progress” After Submit Student Grade should Be A+ | Student Course Grade Be A+ | Pass |
|  | **Teacher Edit:**    **Student Pov:** | | | | | | |

1. **Course Management:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC009 | Verify that admins can create programs. Attempt to create duplicate programs and ensure that the system handles these scenarios correctly. | Login to admin account | 1. Login to admin  2. Go to Dashboard  3. Click Program Management Button  4. Click Add Button and input same program info  5. Click Submit | ID: P001  Name: Bachelor of Science in Computer Science  Department:ST | System should not allow to add same name and same ID program | System Show program id is exist | Pass |
|  | | | | | | |
| TC010 | Verify that admins can edit programs. Attempt to edit existing programs and ensure that the system handles these scenarios correctly. | Login to admin account | 1. Login to admin  2. Go to Dashboard  3. Click Program Management Button  4. Click Details  5. Click Edit button  6. Summit | Input Name: Bachelor of Science in Computer Science to P002(Bachelor of Science in Information Technology) | System should not allow to edit other program to same name program | System Show program name is existed | Pass |

****

1. **Course Statistics Report:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC012 | Generate a course statistics report as a teacher and verify that the report provides accurate information based on the licensed information of the teacher. | Login to teacher account | 1. Login to the teacher account and go to the dashboard to see that teacher’s teaching course.  2. Login to the admin account and go to the dashboard to see the exact teacher’s teaching course. | none | Teacher only can see what courses his teaching, admin can see through report and see all teacher’s teaching courses | Teacher see what courses his teaching, admin see through report and see all teacher’s teaching courses | Pass |
| Teacher 1 Pov:    Admin’s Pov: | | | | | | |

1. **Administrator Privileges:**

| Test Case ID | Test Case Objective | Pre- requisite | Steps | Input Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC013 | Verify that administrators can create, edit user accounts. Attempt to create duplicate accounts or disable existing accounts and ensure that the system handles these scenarios correctly. |  | 1. Login to admin account  2. Go to Dashboard  3. Click User Management  4. Click Add button/ Details(Edit)  5. Input data and Submit | ID: s123456  (s123456 John Smith already registered) | Add new user failed due to existing user  Edit existing account to change same ID is not allow due to it is unique | Add new user failed due to existing user  Edit existing account to change same ID is not allow due to it is unique | Pass |
| Edit ID is read only state: | | | | | | |
| TC014 | Verify that administrators can assign roles and permissions to users. Ensure that the assigned roles and permissions are applied correctly and reflected in the system. |  | 1. Login to admin account  2. Go to Dashboard  3. Click User Management  4. Click Details and Edit Button/Add button | Try to select the role of user | System should allow admin to assign roles to user | System should allow admin to assign roles to user, but edit is not allow/ add is allow | Half-Pass |
|  | Edit Role failed:    Add role allow: | | | | | | |
| TC015 | Verify that administrators have access to all system functionalities and student records. Attempt to perform various actions and verify that the system grants the appropriate access and permissions. |  | 1. Login to admin account  2. Go through system to check all edit/add function page is allow to admin | None | Admin Can Go through all edit/add pages | Admin go through all edit/add pages successfully except Add student to course | Mostly Pass |
|  | Except Case: | | | | | | |

## 

## Conclusion

In conclusion, validation testing for the Student Academic Record System aimed to ensure that the system met the specified requirements and functioned as intended. This involved analyzing requirements, defining test coverage, preparing test data, and establishing clear criteria for evaluating the system's behavior.

Defect testing focuses on identifying and resolving faults or defects in the software. Through a comprehensive test strategy, defects were identified, prioritized, and reported. Collaboration between testers, developers, and stakeholders facilitated prompt resolution of identified issues.

Before discussing the overall conclusion, it is important to note that the current system was developed based on the requirements documents of the core project. While the validation testing and defect testing processes have ensured the system's functionality and reliability to a satisfactory level, there is still room for expansion and improvement in the future.

As the system evolves, there may be opportunities to enhance its features, usability, and performance. This could involve incorporating additional functionalities, improving user interfaces, optimizing system performance, or integrating with other relevant systems. By continuously expanding and refining the system, it can better meet the evolving needs of users and stakeholders.

With ongoing development and expansion, it is essential to continue employing effective testing strategies, including validation testing and defect testing, to ensure that any new additions or modifications align with the requirements and maintain the system's overall quality and reliability.

Overall, while the current system has undergone thorough validation testing and defect testing, it is important to recognize that future efforts will focus on fully expanding and refining the system to further enhance its capabilities and meet the evolving needs of the users and stakeholders.