**WIL Report Management System**

Test plan

­­­

By

**Ms. Phinthip Samutloiwon 552115050**

**Mr. Veerapat In-ongkarn 562115055**

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

**Dr. Prompong Sugunnasil**

**Document History**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **Documents** | | | | | | | |
| WRMS- TestPlan\_V.0.1.docx | Created document  - Add Chapter I  - Add Chapter II - Add Chapter III  - Add Chapter IV | | Draft | 27-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- TestPlan\_V.0.2.docx | - Edit Chapter II  - Edit Chapter III  - Edit Chapter IV | | Draft | 14-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- TestPlan\_V.0.3.docx | - Edit Chapter III  - Edit Chapter IV | | Draft | 15-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- TestPlan\_V.0.4.docx | - Edit Chapter I  - Edit Chapter II  - Edit Chapter III  - Edit Chapter IV | | Draft | 20-04-2017 | PS,VI | PS,VI | PS,VI |

**\*PS = Phinthip Samutloiwon  
\*VI = Veerapat In-ongkarn  
\*PSU= Prompong Sugunnasil**

Table of Contents

[Chapter I | Introduction 7](#_Toc480797572)

[1.1 Purpose 7](#_Toc480797573)

[1.2 Scope 7](#_Toc480797574)

[1.3 Acronyms and Definitions 7](#_Toc480797575)

[Chapter II | Test Procedure 8](#_Toc480797576)

[2.1 Test Objectives 8](#_Toc480797577)

[2.2 Test Duration 8](#_Toc480797578)

[2.3 Test Responsibility 8](#_Toc480797579)

[2.4 Test Strategy 8](#_Toc480797580)

[2.5 Result of Testing 9](#_Toc480797581)

[2.6 Test Environment 9](#_Toc480797582)

[2.6.1 Hardware 9](#_Toc480797583)

[2.6.2 Software 9](#_Toc480797584)

[Chapter III | Unit Test 10](#_Toc480797585)

[3.1 UserService Class 10](#_Toc480797586)

[UTC-01: getUserById(int id): User 10](#_Toc480797587)

[UTC-02: getNotifiesByUserId(int id): Notify[] 10](#_Toc480797588)

[UTC-03: getMyStatistic(int userId): Object 10](#_Toc480797589)

[UTC-04: getStatisticByUserID(int id): Object 10](#_Toc480797590)

[UTC-05: getUserByCode(String code): User 10](#_Toc480797591)

[UTC-06: getMentorsByStudentId(int id):User[] 11](#_Toc480797592)

[UTC-07: getSupervisprdByStudentId(int id):User[] 11](#_Toc480797593)

[UTC-08: setUserofStudent(int id, int sid): User 11](#_Toc480797594)

[UTC-09: deleteUserofStudent(int id, int sid): User 11](#_Toc480797595)

[UTC-10: create(object data): User 11](#_Toc480797596)

[UTC-11: createStudent(object data): User 11](#_Toc480797597)

[UTC-12: createSupervisor(object data): User 11](#_Toc480797598)

[UTC-13: createMentor(object data): User 11](#_Toc480797599)

[UTC-14: update(object data, int id): User 11](#_Toc480797600)

[UTC-15: getNotifiesByUserId(int id): Notify[] 11](#_Toc480797601)

[3.2 ProjectService Class 12](#_Toc480797602)

[UTC-16: show(int id): Project 12](#_Toc480797603)

[UTC-17: all(): Project[] 12](#_Toc480797604)

[UTC-18: getLogsByProject(int id): TaskLog[] 12](#_Toc480797605)

[UTC-19: store(Object request, int userId): Project 12](#_Toc480797606)

[UTC-20: update(Object request, int id): Project 12](#_Toc480797607)

[UTC-21: destroy(int id): Project 12](#_Toc480797608)

[UTC-22: getMyProjectsByUserId(int id): Project[] 12](#_Toc480797609)

[UTC-23: getProjectsByUserID(int id): Project[] 12](#_Toc480797610)

[3.3 TaskService Class 12](#_Toc480797611)

[UTC-24: store(Object request): Task 12](#_Toc480797612)

[UTC-25: getTaskLogsByUsers(User users): TaskLog[] 12](#_Toc480797613)

[UTC-26: getLogsByUserID(int userId): TaskLog[] 12](#_Toc480797614)

[UTC-27: getTasksByProjectID(int id): Task[] 12](#_Toc480797615)

[UTC-28: getTaskByTaskID(int id): Task 12](#_Toc480797616)

[UTC-29: getTasksByTaskID(int id): Task[] 12](#_Toc480797617)

[UTC-30: changeTo(request: Object, id: Integer): Task[] 12](#_Toc480797618)

[3.4 CommentService Class 12](#_Toc480797619)

[UTC-31: comment(Object request): Comment 12](#_Toc480797620)

[UTC-32: getComment(int id): Comment 12](#_Toc480797621)

[UTC-33: getCommentsByTask(int id): Comment 12](#_Toc480797622)

[UTC-34: destroy(int id): Comment 12](#_Toc480797623)

[3.5 UserController Class 12](#_Toc480797624)

[UTC-35: show(int id): User 12](#_Toc480797625)

[UTC-36: update(Object request, int id): User 12](#_Toc480797626)

[UTC-37: getUserByCode(String code): User 12](#_Toc480797627)

[UTC-38: getStatisticByUserID(int id): Object 12](#_Toc480797628)

[UTC-39: getNotifiesByUserId(Object request): Notify[] 12](#_Toc480797629)

[UTC-40: setUserOfStudent(Object request): User 12](#_Toc480797630)

[UTC-41: deleteUserOfStudent(Object request, int id): User 12](#_Toc480797631)

[UTC-42: getNotifiesByUserId(Object request): Notify[] 12](#_Toc480797632)

[3.6 AuthController Class 13](#_Toc480797633)

[UTC-43: authenticate(Object request): token 13](#_Toc480797634)

[UTC-44: getAuthenticatedUser(): User 13](#_Toc480797635)

[UTC-45: register(Object request): User 13](#_Toc480797636)

[3.7 ProjectController Class 13](#_Toc480797637)

[UTC-46: index(): Project[] 13](#_Toc480797638)

[UTC-47: logs(int id): TaskLog[] 13](#_Toc480797639)

[UTC-48: store(Object request): Project 13](#_Toc480797640)

[UTC-49: show(int id): Project 13](#_Toc480797641)

[UTC-50: getProjectByUserID(int id): Project[] 13](#_Toc480797642)

[UTC-51: myProject(Object request): Project[] 13](#_Toc480797643)

[UTC-52: update(Object request, int id): Project 13](#_Toc480797644)

[UTC-53: destroy(int id): Project 13](#_Toc480797645)

[3.8 TaskController Class 13](#_Toc480797646)

[UTC-54: store(Object request): Task 13](#_Toc480797647)

[UTC-55: index() : Task[] 13](#_Toc480797648)

[UTC-56: create(): Task 13](#_Toc480797649)

[UTC-57: getTasksByProject(int id): Task[] 13](#_Toc480797650)

[UTC-58: getTasksLogsByMe(Object request): TaskLog[] 13](#_Toc480797651)

[UTC-59: changeTo(Object request, int id): Task 13](#_Toc480797652)

[UTC-60: show(int id): Task 13](#_Toc480797653)

[UTC-61: update(Object request, int id): Task 13](#_Toc480797654)

[UTC-62: destroy(int id): Task 13](#_Toc480797655)

[3.9 CommentController Class 13](#_Toc480797656)

[UTC-63: comment(Object request): Comment 13](#_Toc480797657)

[UTC-64: getCommentByTask(int id): Comment[] 13](#_Toc480797658)

[Chapter IV | System Test 14](#_Toc480797659)

[4.1 Feature #1 14](#_Toc480797660)

[STC-01: Registration 14](#_Toc480797661)

[STC-02: Edit profile 15](#_Toc480797662)

[STC-03: Login 15](#_Toc480797663)

[STC-04: Logout 15](#_Toc480797664)

[STC-05: Add registration code 15](#_Toc480797665)

[4.2 Feature #2 15](#_Toc480797666)

[STC-06: View tasks 15](#_Toc480797667)

[STC-07: View Statistics 15](#_Toc480797668)

[STC-08: Add project 15](#_Toc480797669)

[STC-09: Delete project 15](#_Toc480797670)

[STC-10: Add task 15](#_Toc480797671)

[STC-11: Edit task 15](#_Toc480797672)

[STC-12: Delete task 15](#_Toc480797673)

[STC-13: Move task 15](#_Toc480797674)

[4.1 Feature #3 15](#_Toc480797675)

[STC-14: View comments 15](#_Toc480797676)

[STC-15: Add comment 15](#_Toc480797677)

[STC-16: Edit comment 15](#_Toc480797678)

[STC-17: Delete comment 15](#_Toc480797679)

[4.2 Feature #4 16](#_Toc480797680)

[STC-18: View weekly report 16](#_Toc480797681)

[STC-19: Generate report 16](#_Toc480797682)

[4.3 Feature #5 16](#_Toc480797683)

[STC-20: Receive web notification 16](#_Toc480797684)

[STC-21: Receive email notification 16](#_Toc480797685)

# **Chapter I | Introduction**

## **Purpose**

The purpose of test plan document is to ensure that all functional requirements are implemented as in SRS documentation as well as provide an approach of unit and system testing. This test plan also helps to manage changes in the early phase of the project since we got more information and revise the plan.

## **Scope**

The scope of this test document includes test objectives, test duration, Responsibility person, Test strategy, planning of unit and system testing with test data, and testing activities. WRMS implement both of white box testing and black box testing techniques. We implemented white box testing to control structure of testing design procedure to derive test cases. Black box testing uses for derive set of input data and possible conditions.

## **Acronyms and Definitions**

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| UTC | Unit Test Case |
| STC | System Test Case |
| SRS | Software Requirement Specification |
| WRMS | WIL Report Management System |

# **Chapter II | Test Procedure**

## **Test Objectives**

The objectives of testing WIL Report Management System are

1. Identify the faults – Finding the fault which cause the failure.
2. Correct and remove the faults – Fixing errors and remove faults.
3. Ensure that all functionalities are implemented correctly.

## **Test Duration**

|  |  |
| --- | --- |
| **Progress** | **Date and Duration** |
| Progress I | Perform date:  Duration: days |
| Final Progress | Perform date:  Duration: days |

## **Test Responsibility**

|  |  |
| --- | --- |
| **Item** | **Responsibility** |
| Unit test | VI |
| Record of unit test | VI |
| System test | VI, PS |
| Record of system test | VI, PS |

## **Test Strategy**

WRMS performed testing strategy by the following steps:

1. Design test cases
2. Prepare test data
3. Determine the expected result
4. Perform testing
5. Record all results

## **Result of Testing**

The results are separate into two parts:

1. Actual result: The actual results after testing
2. Pass/ Fail result:
   1. Pass: The actual result which match with the expected result.
   2. Fail: The actual result which does not match with the expected result.

## **Test Environment**

### **2.6.1 Hardware**

1. **DELL Vostro**

Processor Intel Core i5-5200U (2.20 GHz)

Memory 4 GB DDR3L 500 GB 5400 RPM

1. **MacBook Pro (Retina, Mid 2014)**

Processor 2.5 GHz Intel Core i7

Memory 16 GB 1600 MHz DDR 3

### **2.6.2 Software**

1. Operating system Windows 8.1
2. Operating system macOS Sierra
3. phpunit

# **Chapter III | Unit Test**

## UserService Class

### getUserById(int id): User

**Test description:** Test getUserById(int id) method by input id of user as a parameter.

**Test data:** refer to **Appendix A** in “User” content.

**Result:** The result will return in console.

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input user id as a request parameter | id = 1 | Get a student name `Veerapat In-ongkarn ` and id=1 |
| 2 | Test by input user id as a request parameter | id = 2 | Get a mentor name ` Albert ` and id=2 |
| 3 | Text by input wrong-id parameter | Id = 10 | Get an empty object |
| 4 | Test by input empty parameter | empty object | Response status 404 |

### getNotifiesByUserId(int id): Notify[]

### getMyStatistic(int userId): Object

### getStatisticByUserID(int id): Object

### getUserByCode(String code): User

**Test description:** Test getUserByCode(int code) method by input code of user as a parameter

**Test data:** refer to **Appendix A** in “User” content.

**Result:** The result will return in console.

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input user code as a request parameter | code = 12345 | Get a mentor name ` Albert Einstein`, id=2 and code=12345 |
| 2 | Text by input wrong-code parameter | Id = 22222 | Get an empty object |
| 3 | Test by input empty parameter | empty object | Response status 404 |

### getMentorsByStudentId(int id):User[]

**Test description:** Test getMentorsByStudentId (int id) method by input id of student as a parameter.

**Test data:** refer to **Appendix A** in “User” content.

**Result:** The result will return in console.

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input student id as a request parameter | id = 1 | Get a mentors of student id = 1 |
| 2 | Test by input mentor id as a request parameter | Id = 2 | Get a empty array |
| 3 | Test by input empty parameter | empty object | Response status 404 |

### getSupervisprdByStudentId(int id):User[]

### setUserofStudent(int id, int sid): User

### deleteUserofStudent(int id, int sid): User

### create(object data): User

### createStudent(object data): User

### createSupervisor(object data): User

### createMentor(object data): User

### update(object data, int id): User

### getNotifiesByUserId(int id): Notify[]

## ProjectService Class

### show(int id): Project

### all(): Project[]

### getLogsByProject(int id): TaskLog[]

### store(Object request, int userId): Project

### update(Object request, int id): Project

### destroy(int id): Project

### getMyProjectsByUserId(int id): Project[]

### getProjectsByUserID(int id): Project[]

## TaskService Class

### store(Object request): Task

### getTaskLogsByUsers(User users): TaskLog[]

### getLogsByUserID(int userId): TaskLog[]

### getTasksByProjectID(int id): Task[]

### getTaskByTaskID(int id): Task

### getTasksByTaskID(int id): Task[]

### changeTo(request: Object, id: Integer): Task[]

## CommentService Class

### comment(Object request): Comment

### getComment(int id): Comment

### getCommentsByTask(int id): Comment

### destroy(int id): Comment

## UserController Class

### show(int id): User

### update(Object request, int id): User

### getUserByCode(String code): User

### getStatisticByUserID(int id): Object

### getNotifiesByUserId(Object request): Notify[]

### setUserOfStudent(Object request): User

### deleteUserOfStudent(Object request, int id): User

### getNotifiesByUserId(Object request): Notify[]

## AuthController Class

### authenticate(Object request): token

### getAuthenticatedUser(): User

### register(Object request): User

## ProjectController Class

### index(): Project[]

### logs(int id): TaskLog[]

### store(Object request): Project

### show(int id): Project

### getProjectByUserID(int id): Project[]

### myProject(Object request): Project[]

### update(Object request, int id): Project

### destroy(int id): Project

## TaskController Class

### store(Object request): Task

### index() : Task[]

### create(): Task

### getTasksByProject(int id): Task[]

### getTasksLogsByMe(Object request): TaskLog[]

### changeTo(Object request, int id): Task

### show(int id): Task

### update(Object request, int id): Task

### destroy(int id): Task

## CommentController Class

### comment(Object request): Comment

### getCommentByTask(int id): Comment[]

# **Chapter IV | System Test**

## **Feature #1**

### **Registration**

**Description:** Test for UC-01. Visitor can register to use the services.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the Login page.

**Test script**:

1. Visitor clicks on “Register” button.
2. Visitor inputs first name, last name, description, company, position, email, password profile image, signature image, start date, and role of user.
3. Visitor clicks “Create” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Register with non-exist account as a student | first\_name: "Isaac"  last\_name: "Newton"  email: Newton@xmail.com  password: "newton1"  description: "I am a student"  company: "XYZ University"  position: "Student"  avatar: "1492663499.png"  sign: "1492663504.png"  start: "2017-04-01"  code: "null"  role: "student" | - The new account will be created.  - The system redirects to dashboard page. |
|  | 2 | Register with non-exist account as a mentor | first\_name:"Michael"  last\_name:"Faraday"  email:Faraday@xmail.com  password: "faraday1"  description:"I am a mentor"  company:"FaradayCompany"  position:"Owner"  avatar:"1492665000.jpg"  sign:"1492665004.png"  start: "null"  code:"16790"  role:"mentor" | - The new account will be created.  - The code of mentor will be created.  - The system redirects to dashboard page. |
|  | 2 | Register with non-exist account as a supervisor | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein@xmail.com  password: "einstein1"  description: "I am a Supervisor"  company: "XYZ University"  position: "Lecturer"  avatar: "1492663030.jpg"  sign: "1492662998.png"  start: “null”  code: "8f14e"  role: "supervisor" | - The new account will be created.  - The code of supervisor will be created.  - The system redirects to dashboard page. |
|  | 2 | Register with an account with existing email “Einstein@xmail.com” | first\_name: "XYZ"  last\_name: " XYZ "  email: Einstein@xmail.com  password: "XYZ"  description: " XYZ "  company: " XYZ "  position: " XYZ "  avatar: " XYZ "  sign: " XYZ "  start: “XYZ”  code: " XYZ "  role: " XYZ " | - The system displays an error message “The email has already been taken” |
|  | 2 | Register with incorrect email constraint | first\_name: "XYZ"  last\_name: " XYZ "  email: "Einstein"  password: "XYZ"  description: " XYZ "  company: " XYZ "  position: " XYZ "  avatar: " XYZ "  sign: " XYZ "  start: " XYZ "  code: " XYZ "  role: " XYZ " | - The system displays an error message “The email must be a valid email address”. |
|  | 2 | Register with empty first\_name field. (required) | first\_name: " " | - The system displays an error message “The first name field is required”. |
|  | 2 | Register with empty last\_name field. (required) | last\_name: " " | - The system displays an error message “The last name field is required”. |
|  | 2 | Register with empty email field. (required) | email: " " | - The system displays an error message “The email field is required”. |
|  | 2 | Register with empty password field. (required) | password: " " | - The system displays an error message “The password field is required”. |
|  | 2 | Register with empty start field. (required) | start: " " | - The system displays an error message “The start field is required”. |
|  | 2 | Register with empty description field. (required) | description: " " | - The system displays an error message “The description field is required”. |
|  | 2 | Register with empty company field. (required) | company: " " | - The system displays an error message “The company field is required”. |
|  | 2 | Register with empty position field. (required) | position: " " | - The system displays an error message “The position field is required”. |
|  | 2 | Register with empty avatar field. (required) | avatar: " " | - The system displays an error message “The avatar field is required”. |
|  | 2 | Register with empty sign field. (required) | sign: " " | - The system displays an error message “The sign field is required”. |

### Edit profile

**Description:** Test for UC-02. User can edit their own profile.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the Profile page.

**Test script**:

a.) Student role

1. Student clicks on “Edit” icon.
2. Student edits first name, last name, description, company, position, email, password, profile image, signature image, and start date.
3. Student clicks “update” button.

b.) Mentor or supervisor role

1. Mentor or supervisor clicks on “Edit” icon.
2. Mentor or supervisor edits first name, last name, description, company, position, email, password, profile image, and signature image.
3. Mentor or supervisor clicks “update” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | a.2 | Edit profile correct profile of student | first\_name: "nIsaac"  last\_name: "nNewton"  email: nNewton@xmail.com  password: "nnewton1"  description: "I am a new student"  company: "XYZ University"  position: "Student"  avatar: "1592663499.png"  sign: "1592663504.png"  start: "2017-04-02” | - The system displays profile page of student.  - The system will display the profile of nIsaac nNewton with his photo and the following info; Name: nIsaac nNewton Description: I am a new student Company: XYZ University Position: Student |
|  | b.2 | Edit profile with correct profile of Mentor | first\_name:"nMichael"  last\_name:"nFaraday"  email: "nFaraday@xmail.com"  password: "nfaraday1"  description:"I am a new mentor"  company:"FaradayCompany"  position:"Owner"  avatar:"1492665000.jpg"  sign:"1492665004.png" | - The system displays profile page of student.  - The system will display the profile of nMichael nFaraday with his photo and the following info; Name: nMichael nFaraday Description: I am a mentor  Company: FaradayCompany Position: Owner |
|  | a.2, b.2 | Edit profile with incorrect email constraint | email: "newton " | The system displays an error message “The email must be a valid email address.” |
|  | a.2, b.2 | Edit profile with empty first name field (required) | first\_name:" " | The system displays an error message “The first name field is required”. |
|  | a.2, b.2 | Edit profile with empty last name field (required) | last\_name:" " | The system displays an error message “The email field is required”. |
|  | a.2, b.2 | Edit profile with empty password field (required) | password: " " | The system displays an error message “The password field is required”. |
|  | a.2 | Edit profile with empty start field (required) | start: " ” | The system displays an error message “The start field is required”. |
|  | a.2, b.2 | Edit profile with empty description field (required) | description: " " | The system displays an error message “The description field is required”. |
|  | a.2, b.2 | Edit profile with empty company field (required) | company:" " | The system displays an error message “The company field is required”. |
|  | a.2, b.2 | Edit profile with empty company field (required) | position:" " | The system displays an error message “The position field is required”. |
|  | a.2, b.2 | Edit profile with empty avatar field (required) | avatar: " " | The system displays an error message “The position field is required”. |
|  | a.2, b.2 | Edit profile with empty sign field (required) | sign: " " | The system displays an error message “The position field is required”. |

### Login

**Description:** Test for UC-03. User can login into the WIL report management system.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the Login page.

**Test script**:

1. User inputs email and password.
2. User click “Login” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 1 | Login with an exist account as a student | email: "Newton@xmail.com"  password: "newton1" | - Login Successful.  - The system will redirects to Dashboard of a student. |
|  | 1 | Login with an exist account as a Mentor | email: "Faraday@xmail.com"  password: "faraday1" | - Login Successful.  - The system will redirects to Dashboard of a Mentor. |
|  | 1 | Login with an exist account as a Supervisor | email: "Einstein@xmail.com"  password: "einstein1" | - Login Successful.  - The system will redirects to Dashboard of a Supervisor. |
|  | 1 | Login with an incorrect email | email: "test"  password: "newton1" | - The system displays an error message “The email or password is incorrect” |
|  | 1 | Login with an incorrect password | email: "Newton@xmail.com"  password: "incorrect" | - The system displays an error message “The email or password is incorrect” |
|  | 1 | Login with incorrect email and password | email: " incorrect @xmail.com"  password: "incorrect" | - The system displays an error message “The email or password is incorrect” |

### Logout

**Description:** Test for UC-04. User can logout.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is login into the system.

**Test script**:

1. User is logging in.
2. User click “Logout” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Logout from the student account | email: "Newton@xmail.com"  password: "newton1" | - Logout successful.  - The system will redirects to Login page. |
|  | 2 | Logout from the mentor account | email: "Faraday@xmail.com"  password: "faraday1" | - Logout successful.  - The system will redirects to Login page. |
|  | 2 | Logout from the supervisor account | email: "Einstein@xmail.com"  password: "einstein1" | - Logout successful.  - The system will redirects to Login page. |
|  | 2 | Click ‘back’ on the browser after successfully logged out. | - | - The system will display Login page. |

### Add registration code

**Description:** Test for UC-05. Student can add registration code.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the profile page.

**Test script**:

1. User inputs the mentor or supervisor code.
2. User click “+” button.
3. User clicks “Confirm”

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Add an exist code | Code: "8f14e" | - The system display “Albert Einstein” as the supervisor in profile page. |
|  | 2 | Add the incorrect code which is shorter than 6 characters | code: "a12" | The system locks the “+” button. |
|  | 2 | Add the incorrect code which is longer than 6 characters | code: "aa1234567" | The system locks the “+” button. |

## **Feature #2**

### View tasks

**Description:** Test for UC-06. User can view tasks of a project.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the dashboard.

**Test script**:

a.) Student role

1. Student clicks on a project name.

b.) Mentor or supervisor role

1. Mentor or supervisor clicks the profile menu.
2. Mentor or supervisor clicks on a student name.
3. Mentor or supervisor clicks on a project name.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Note** | **Expected result** |
|  | a.1, b.3 | View the project detail which includes task list | Project id: 2 Project: "Gravity project" | The system displays the list of tasks as following; {Task1, Task2,Task3, Task4, Task5} |

### View Statistics

**Description:** Test for UC-07. User can view the statistics of tasks.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the dashboard.

**Test script**:

a.) Student role

1. Student is in the dashboard.

b.) Mentor or supervisor role

1. Mentor or supervisor clicks the profile menu.
2. Mentor or supervisor clicks on a student name.
3. Mentor or supervisor is in the dashboard of the selected student.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | a.1, b.3 | View the statistic of tasks. | Student’s name: "Isaac Newton" | The system displays the statistics of tasks as following; Projects: 2 Tasks: 7 Days: 24 |

### Create project

**Description:** Test for UC-08. Student can create a project.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the dashboard.

**Test script**:

1. Student clicks on “create new project” button.
2. Student inputs project name, description, and start date.
3. Student clicks on create button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Input correct project name, description, and start date | project name: "New project"  description: "This is a new project"  Start date: "2017-04-01" | The system displays the new project on the dashboard. |
|  | 2 | Empty the project name field (required) | project name: ""  description: "This is a new project"  Start date: "2017-04-01" | System locks the “create” button. |
|  | 2 | Empty the description field (required) | project name: " New project "  description: " "  Start date: "2017-04-01" | System locks the “create” button. |

### Edit project

**Description:** Test for UC-09. Student can edit a project.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the dashboard.

**Test script**:

1. Student clicks on “edit project” button.
2. Student inputs project name, description, and start date.
3. Student clicks on edit button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Input correct project name, description, and start date | project name: "Edit project"  description: "This is an edited project"  Start date: "2017-04-01" | The system displays the edited project on the dashboard. |
|  | 2 | Empty the project name field (required) | project name: ""  description: "This is an edited project"  Start date: "2017-04-01" | System locks the “edit” button. |
|  | 2 | Empty the description field (required) | project name: "Edit project  description: " "  Start date: "2017-04-01" | System locks the “edit” button. |

### Delete project

**Description:** Test for UC-10. Student can delete a project.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the dashboard.

**Test script**:

1. Student clicks on “delete project” button.
2. Student clicks “Confirm” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Test to delete a project | - | The system displays message “Delete a project success” |

### Create task

**Description:** Test for UC-11. Student can create a task

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the task page.

**Test script**:

1. Student clicks on “create new task” button.
2. Student inputs task name, description, and start date.
3. Student clicks on create button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Input correct task name, description, and start date | project name: "New task"  description: "This is a new task"  Start date: "2017-04-01" | - System displays the new task on the task page.  - System redirects to Task page |
|  | 2 | Empty the task name field (required) | project name: ""  description: "This is a new task"  Start date: "2017-04-01" | System displays an error message “The name field is required.” |
|  | 2 | Empty the description field (required) | project name: "New task"  description: " "  Start date: "2017-04-01" | System displays an error message “The description field is required.” |
|  | 2 | Empty the start date field (required) | project name: "New task"  description: "This is a new task"  Start date: " " | System displays an error message “The start field is required.” |

### Edit task

**Description:** Test for UC-12. Student can edit a task

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the task page.

**Test script**:

1. Student clicks on “edit task” button.
2. Student inputs task name, description.
3. Student clicks on edit button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Input correct task name, description | project name: "Edit task"  description: "This is an edited task" | - System displays the edited task on the task page.  - System redirects to Task page |
|  | 2 | Empty the task name field (required) | project name: ""  description: " This is an edited task" | System displays an error message “The name field is required.” |

### Delete task

**Description:** Test for UC-13. Student can delete a task

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the task page.

**Test script**:

1. Student clicks on “delete task” button.
2. Student click “Confirm” button

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Test to delete a task | - | - System displays a message “Delete success” |

### Change a status of task

**Description:** Test for UC-14. Student can change status of a task. There are three status which are “To-Do”, “Doing”, and “Done”

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in the task page.

**Test script**:

a.) Click status button in List tab

1. Student clicks on “LIST” tab in the Task page.
2. Student clicks on the status button.
3. Student selects start date.
4. Student click “Confirm” button.

b.) Drag and drop

1. Student clicks on “CARD” tab in the Task page.
2. Student drags a task card and drop it to another column.
3. Student selects start date.
4. Student click “Confirm” button.

c.) Click status button in task detail page

1. Student clicks on a task to see a task detail page.
2. Student clicks on the status button.
3. Student selects start date.
4. Student click “Confirm” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | a4 | Select the start date.  Test on LIST tab. | Start date: "2017-04-01" | - The status of task is changed to “Doing”.  - The start date is displayed on a task list. |
|  | b4 | Select the start date.  Test on CARD tab. | Start date: "2017-04-01" | - The status of task is changed to “Doing”.  - The start date is displayed on a task list. |
|  | c4 | Select the start date.  Test on task detail page. | Start date: "2017-04-01" | - The status of task is changed to “Doing”.  - The start date is displayed on a task list. |
|  | a4, b4, c4 | Empty the start date field | Start date: " " | System locks the “Confirm” button. |
|  | b2 | Drag a task card and does not drop in any column | - | - System displays forbidden sign as a mouse cursor.  - System locks for changing the status of a task. |
|  | b2 | Drag a task card and drop it to the same column | - | The task is still in the current column |

## **Feature #3**

### View comments

**Description:** Test for UC-15. User can view a comment.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the task detail page.

**Test script**:

1. User clicks on a task name.
2. User can view the comment(s).

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Note** | **Expected result** |
|  | 1 | View the comment(s) in a specific task. | Task id: 1 | The system displays the comment(s) as following; { Comment id:7 Text: Test comment by Supervisor User: Albert Einstein} , {Comment id:8 Text: Test comment 1 User: Isaac Newton} |

### Create comment

**Description:** Test for UC-16. User can create a comment.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in a task detail page.

**Test script**:

1. User fills the text(s) in a comment form.
2. User clicks “comment” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Input a comment text. | Comment text: "Test comment" | - System displays the content of comment “Test comment”  - System displays a message “Comment success” |
|  | 2 | Empty comment text | Comment text: " " | System locks comment button. |

### Delete comment

**Description:** Test for UC-17. User can delete their own comment.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the task detail page.

**Test script**:

1. User clicks on “delete comment” button.
2. User click “Confirm” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Test to delete a comment | - | - System displays a message “Delete success” |

### View activities

**Description:** Test for UC-18. User can view activities of projects.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. User is in the task page.

**Test script**:

1. User click “Activities” tab.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Test to view all tasks activities in a specific project. | - | - System displays list of activities; { Last Saturday at 5:38 PM - Isaac Newton doing Task5}, { Last Thursday at 12:01 PM - Isaac Newton to do Task5} |

## **Feature #4**

### Generate a report

**Description:** Test for UC-19. Student can generate a weekly report.

**Precondition:**

1. The test data has been prepared. (See Appendix A)

**Test script**:

1. User clicks on “report” menu.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 1 | Test to generate a report | - | - System displays the weekly report which is separated by the project. |

### Print a report

**Description:** Test for UC-20. Student can print a weekly report.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. Student is in a report page.

**Test script**:

1. Student clicks on “Print” button.
2. System provides UI to see a preview of a report with “Print” and “Cancel” button.
3. Student clicks “print” button.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 3 | Test to print a weekly report | - | System send a weekly report to a printer. |

## **Feature #5**

### Receive web notification

**Description:** Test for UC-21. User can receive an activity notification via Web application.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. A new activity is created (A new project is created, A project is edited, A new task is created, A task is edited. A new comment is created).

**Test script**:

1. User did some project or task activities.
2. User receives an activity notification.
   1. User receives an activity notification pop-up on the website.
   2. User receives an activity notification on “Today” notification tab.
   3. User receives an activity notification on “All” tab.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Note** | **Expected result** |
|  | 2.1 | Test to receive an activity notification pop-up | Add a comment  Text: Test comment Task id: 10 | System displays a notification message “You got a notification” |
|  | 2.2 | Test to receive an activity notification on “Today” notification tab | Add a comment  Text: Test comment Task id: 10 | System displays a notification as the following;  Isaac Newton 2017-04-26 15:24:31 TASK#10 Gravity project |
|  | 2.3 | Test to receive an activity notification on “All” tab | Add a comment  Text: Test comment Task id: 10 | System displays a notification as the following;  Isaac Newton 2017-04-26 15:24:31 TASK#10 Gravity project |

### Receive email notification

**Description:** Test for UC-22. User can receive an activity notification via Email.

**Precondition:**

1. The test data has been prepared. (See Appendix A)
2. A new activity is created (A new project is created, A project is edited, A new task is created, A task is edited. A new comment is created).

**Test script**:

1. User did some project or task activities.
2. User receives an email of activity notification.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Note** | **Expected result** |
|  | 2 | Test to receive an activity notification via email | Add a comment  Receiver: Isaac Newton Comment creator: Albert Einstein Text: Test comment Task id: 10 | System send an email notification as following;  Hi Issac Newton,  Albert Einstein has just comment on Task #10 Project: Gravity project --------------------------------- Comment: Test task ---------------------------------  Open this comment in Your Web browser. |