**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, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- TestPlan\_V.0.2.docx | - Edit Chapter II  - Edit Chapter III  - Edit Chapter IV | | Draft | 14-02-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- TestPlan\_V.0.3.docx | - Edit Chapter III  - Edit Chapter IV | | Draft | 15-02-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- TestPlan\_V.0.4.docx | - Edit Chapter III  - Edit Chapter IV | | Draft | 20-04-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- TestPlan\_V.0.5.docx | - Edit Chapter III  - Edit Chapter IV | | Draft | 28-04-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- TestPlan\_V.0.6.docx | Release WRMS test plan | | Release | 09-05-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |

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

Table of Contents

[Chapter I | Introduction 6](#_Toc481179421)

[1.1 Purpose 6](#_Toc481179422)

[1.2 Scope 6](#_Toc481179423)

[1.3 Acronyms and Definitions 6](#_Toc481179424)

[Chapter II | Test Procedure 7](#_Toc481179425)

[2.1 Test Objectives 7](#_Toc481179426)

[2.2 Test Duration 7](#_Toc481179427)

[2.3 Test Responsibility 7](#_Toc481179428)

[2.4 Test Strategy 7](#_Toc481179429)

[2.5 Result of Testing 8](#_Toc481179430)

[2.6 Test Environment 8](#_Toc481179431)

[2.6.1 Hardware 8](#_Toc481179432)

[2.6.2 Software 8](#_Toc481179433)

[Chapter III | Unit Test 9](#_Toc481179434)

[3.1 UserService Class 9](#_Toc481179435)

[UTC-01: getUserById(int id): User 9](#_Toc481179436)

[UTC-02: getNotifiesByUserId(int id): Notify[] 9](#_Toc481179437)

[UTC-03: getMyStatistic(int userId): Object 10](#_Toc481179438)

[UTC-04: getStatisticByUserID(int id): Object 10](#_Toc481179439)

[UTC-05: getUserByCode(String code): User 11](#_Toc481179440)

[UTC-06: getMentorsByStudentId(int id):User[] 11](#_Toc481179441)

[UTC-07: getSupervisprdByStudentId(int id):User[] 12](#_Toc481179442)

[UTC-08: setUserofStudent(int id, int sid): User 12](#_Toc481179443)

[UTC-09: deleteUserofStudent(int id, int sid): User 12](#_Toc481179444)

[UTC-10: create(object data): User 13](#_Toc481179445)

[UTC-11: createStudent(object data): User 14](#_Toc481179446)

[UTC-12: createSupervisor(object data): User 15](#_Toc481179447)

[UTC-13: createMentor(object data): User 16](#_Toc481179448)

[UTC-14: update(object data, int id): User 16](#_Toc481179449)

[UTC-15: getReportsByUserId(int id): Project[] 17](#_Toc481179450)

[UTC-16: getStatisticByUsers(User[] users): Object 18](#_Toc481179451)

[3.2 ProjectService Class 18](#_Toc481179452)

[UTC-17: show(int id): Project 18](#_Toc481179453)

[UTC-18: all(): Project[] 19](#_Toc481179454)

[UTC-19: getLogsByProject(int id): TaskLog[] 19](#_Toc481179455)

[UTC-20: store(Object request, int userId): Project 19](#_Toc481179456)

[UTC-21: update(Object request, int id): Project 20](#_Toc481179457)

[UTC-22: destroy(int id): Project 21](#_Toc481179458)

[UTC-23: getMyProjectsByUserId(int id): Project[] 21](#_Toc481179459)

[UTC-24: getProjectsByUserID(int id): Project[] 21](#_Toc481179460)

[3.3 TaskService Class 22](#_Toc481179461)

[UTC-25: store(Object request): Task 22](#_Toc481179462)

[UTC-26: update(Object request, int taskId): Task 23](#_Toc481179463)

[UTC-27: getTaskLogsByProjectIdAndDates(int projectId, Date start, Date end) 23](#_Toc481179464)

[UTC-28: getTaskLogsByUsers(User users): TaskLog[] 24](#_Toc481179465)

[UTC-29: getLogsByUserID(int userId): TaskLog[] 24](#_Toc481179466)

[UTC-30: getTasksByProjectID(int id): Task[] 25](#_Toc481179467)

[UTC-31: getTaskByTaskID(int id): Task 25](#_Toc481179468)

[UTC-32: getTasksByUserID(int id): Task[] 26](#_Toc481179469)

[UTC-33: changeTo(request: Object, id: int): Task 26](#_Toc481179470)

[3.4 CommentService Class 27](#_Toc481179471)

[UTC-34: comment(Object request): Comment 27](#_Toc481179472)

[UTC-35: getComment(int id): Comment 27](#_Toc481179473)

[UTC-36: getCommentsByTask(int id): Comment 28](#_Toc481179474)

[UTC-37: destroy(int id): Comment 28](#_Toc481179475)

[Chapter IV | System Test 29](#_Toc481179476)

[4.1 Feature #1 29](#_Toc481179477)

[STC-01: Registration 29](#_Toc481179478)

[STC-02: Edit profile 33](#_Toc481179479)

[STC-03: Login 36](#_Toc481179480)

[STC-04: Logout 37](#_Toc481179481)

[STC-05: Add registration code 38](#_Toc481179482)

[4.2 Feature #2 39](#_Toc481179483)

[STC-06: View tasks 39](#_Toc481179484)

[STC-07: View Statistics 39](#_Toc481179485)

[STC-08: Create project 40](#_Toc481179486)

[STC-09: Edit project 41](#_Toc481179487)

[STC-10: Delete project 42](#_Toc481179488)

[STC-11: Create task 42](#_Toc481179489)

[STC-12: Edit task 43](#_Toc481179490)

[STC-13: Delete task 44](#_Toc481179491)

[STC-14: Change a status of task 44](#_Toc481179492)

[4.1 Feature #3 46](#_Toc481179493)

[STC-15: View comments 46](#_Toc481179494)

[STC-16: Create comment 46](#_Toc481179495)

[STC-17: Delete comment 47](#_Toc481179496)

[STC-18: View activities 47](#_Toc481179497)

[4.2 Feature #4 48](#_Toc481179498)

[STC-19: Generate a report 48](#_Toc481179499)

[STC-20: Print a report 48](#_Toc481179500)

[4.3 Feature #5 49](#_Toc481179501)

[STC-21: Receive web notification 49](#_Toc481179502)

[STC-22: Receive email notification 50](#_Toc481179503)

# **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 |

**Definition**

|  |  |
| --- | --- |
| Unit test | A test of individual programs or modules in order to remove design or programming errors. [IEEE90] |
| System test | System test is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. |
| Validation | Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use a fulfilled (“doing the right thing”).Part of quality control. |
| Verification | Confirmation at the end of the process by examination and provision of objective evidence that specified requirements to the process has been fulfilled (“doing things right”). Part of quality control. [IEEE90 |

# **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 | User  {  "id": 1,  "first\_name": "Veerapat",  "last\_name": "In-ongkarn",  "email": "karjkeng@hotmail.com",  "description": "Hi!....",  "company": "Buzzwoo!",  "position": "Junior Frontend Developer",  "avatar": ‘avatar.jpg’,  "sign": ‘sign.jpg’,  "start": "09/03/2016",  "role": "student",  "code": null,  "created\_at": null,  "updated\_at": null  } |
| 2 | Test by input id that is not existed in the database. | Id = 99 | null |
| 3 | Test by input empty parameter | Id = null | null |

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

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

**Test data:** refer to **Appendix A** in “User” and “Notify” 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 | Notify[]  [{  "id": 6,  "obj\_id": 1,  "type": "COMMENT",  "created\_at": "2017-04-20 12:13:56",  "updated\_at": "2017-04-20 12:13:56",  “comment” : {…},  “task\_log” : {…}  }, … ] |
| 2 | Test by input id that is not existed in the database. | Id = 99 | null |
| 3 | Test by input empty parameter | Id = null | null |

### getMyStatistic(int userId): Object

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

**Test data:** refer to **Appendix A** in “User”, “Project” and “Task” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input my user id as a request parameter | id = 8 | {  total\_projects = 1,  total\_tasks = 7  } |
| 2 | Test by input user id as a request parameter | Id = 1 | {  total\_projects = 1,  total\_tasks = 0  } |
| 3 | Test by input id that is not existed in the database. | Id = 99 | {  total\_projects = 0,  total\_tasks = 0  } |

### getStatisticByUserID(int id): Object

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

**Test data:** refer to **Appendix A** in “User”, “Project” and “Task” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input my user id as a request parameter | id = 8 | {  total\_projects = 1,  total\_tasks = 7  } |
| 2 | Test by input user id as a request parameter | Id = 1 | {  total\_projects = 1,  total\_tasks = 0  } |
| 3 | Test by input id that is not existed in the database. | Id = 99 | {  total\_projects = 0,  total\_tasks = 0  } |

### 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 = ‘16790’ | User  {  "id": 6,  "first\_name": "Michael",  "last\_name": "Faraday",  "email": "Faraday@xmail.com",  "description": "I am a mentor",  "company": "FaradayCompany",  "position": "Owner",  "avatar": "1492665000.jpg",  "sign": "1492665004.png",  "start": null,  "role": "mentor",  "code": "16790",  "created\_at": "2017-04-20 11:31:35",  "updated\_at": "2017-04-20 12:10:05"  } |
| 2 | Text by input code that is not existed in the database. | code = ‘11111’ | null |
| 3 | Test by input empty parameter | code = null | null |

### 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” and “Students\_has\_Users” 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 = 8 | User[]  [  {  "id": 6,  "first\_name": "Michael",  "last\_name": "Faraday",  "email": "Faraday@xmail.com",  "description": "I am a mentor",  "company": "FaradayCompany",  "position": "Owner",  "avatar": "1492665000.jpg",  "sign": "1492665004.png",  "start": null,  "role": "mentor",  "code": "16790",  "created\_at": "2017-04-20 11:31:35",  "updated\_at": "2017-04-20 12:10:05",  "pivot": {  "student\_id": 8,  "user\_id": 6  }  }  ] |
| 2 | Test by input student id as a request parameter | Id = 1 | null |
| 3 | Test by input id that is not existed in the database. | Id = 99 | null |

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

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

**Test data:** refer to **Appendix A** in “User” and “Students\_has\_Users” 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 = 8 | User[]  [  {  "id": 7,  "first\_name": "Albert",  "last\_name": "Einstein",  "email": "Einstein@xmail.com",  "description": "I am a Supervisor",  "company": "XYZ University",  "position": "Leuturer",  "avatar": "1492663030.jpg",  "sign": "1492662998.png",  "start": null,  "role": "supervisor",  "code": "8f14e",  "created\_at": "2017-04-20 11:37:38",  "updated\_at": "2017-04-20 14:12:33",  "pivot": {  "student\_id": 8,  "user\_id": 7  }  }  ] |
| 2 | Test by input student id as a request parameter | Id = 1 | null |
| 3 | Test by input id that is not existed in the database. | Id = 99 | null |

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

**Test description:** Test setUserOfStudent(int id, int sid) method by input id of mentor or supervisor as a parameter and 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 id of mentor or supervisor as a parameter and input id of student as a parameter. | Id: int = 12  sid: int = 8 | User  {  "id": 8,  "first\_name": "Isaac",  "last\_name": "Newton",  "email": "Newton@xmail.com",  "description": "I am a student",  "company": "XYZ University",  "position": "Student",  "avatar": "1492663499.png",  "sign": "1492663504.png",  "start": "2017-04-01",  "role": "student",  "code": null,  "created\_at": "2017-04-20 11:43:19",  "updated\_at": "2017-04-20 11:45:08",  "users": [  {  "id": 7,  "first\_name": "Albert",  "last\_name": "Einstein",  "email": "Einstein@xmail.com",  "description": "I am a Supervisor",  "company": "XYZ University",  "position": "Leuturer",  "avatar": "1492663030.jpg",  "sign": "1492662998.png",  "start": null,  "role": "supervisor",  "code": "8f14e",  "created\_at": "2017-04-20 11:37:38",  "updated\_at": "2017-04-20 14:12:33",  "pivot": {  "student\_id": 8,  "user\_id": 7  }  },  {  "id": 6,  "first\_name": "Michael",  "last\_name": "Faraday",  "email": "Faraday@xmail.com",  "description": "I am a mentor",  "company": "FaradayCompany",  "position": "Owner",  "avatar": "1492665000.jpg",  "sign": "1492665004.png",  "start": null,  "role": "mentor",  "code": "16790",  "created\_at": "2017-04-20 11:31:35",  "updated\_at": "2017-04-20 12:10:05",  "pivot": {  "student\_id": 8,  "user\_id": 6  }  },  {  "id": 12,  "first\_name": "Alexander",  "last\_name": "Graham Bell",  "email": "Graham@xmail.com",  "description": "I am a Mentor",  "company": "ABCCompany",  "position": "Owner",  "avatar": "1492672181.jpg",  "sign": "1492672154.png",  "start": null,  "role": "mentor",  "code": "c20ad",  "created\_at": "2017-04-20 14:08:21",  "updated\_at": "2017-04-20 14:09:49",  "pivot": {  "student\_id": 8,  "user\_id": 12  }  }  ],  "students": [],  "mentors": [  {  "id": 6,  "first\_name": "Michael",  "last\_name": "Faraday",  "email": "Faraday@xmail.com",  "description": "I am a mentor",  "company": "FaradayCompany",  "position": "Owner",  "avatar": "1492665000.jpg",  "sign": "1492665004.png",  "start": null,  "role": "mentor",  "code": "16790",  "created\_at": "2017-04-20 11:31:35",  "updated\_at": "2017-04-20 12:10:05",  "pivot": {  "student\_id": 8,  "user\_id": 6  }  },  {  "id": 12,  "first\_name": "Alexander",  "last\_name": "Graham Bell",  "email": "Graham@xmail.com",  "description": "I am a Mentor",  "company": "ABCCompany",  "position": "Owner",  "avatar": "1492672181.jpg",  "sign": "1492672154.png",  "start": null,  "role": "mentor",  "code": "c20ad",  "created\_at": "2017-04-20 14:08:21",  "updated\_at": "2017-04-20 14:09:49",  "pivot": {  "student\_id": 8,  "user\_id": 12  }  }  ],  "supervisors": [  {  "id": 7,  "first\_name": "Albert",  "last\_name": "Einstein",  "email": "Einstein@xmail.com",  "description": "I am a Supervisor",  "company": "XYZ University",  "position": "Leuturer",  "avatar": "1492663030.jpg",  "sign": "1492662998.png",  "start": null,  "role": "supervisor",  "code": "8f14e",  "created\_at": "2017-04-20 11:37:38",  "updated\_at": "2017-04-20 14:12:33",  "pivot": {  "student\_id": 8,  "user\_id": 7  }  }  ]  } |
| 2 | Test by sid is not exist in database | sid: int=999 | null |

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

**Test description:** Test deleteUserOfStudent(int id, int sid) method by input id of mentor or supervisor as a parameter and 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 id of mentor or supervisor as a parameter and input id of student as a parameter. | Id: int = 12  sid: int = 8 | User  {  "id": 8,  "first\_name": "Isaac",  "last\_name": "Newton",  "email": "Newton@xmail.com",  "description": "I am a student",  "company": "XYZ University",  "position": "Student",  "avatar": "1492663499.png",  "sign": "1492663504.png",  "start": "2017-04-01",  "role": "student",  "code": null,  "created\_at": "2017-04-20 11:43:19",  "updated\_at": "2017-04-20 11:45:08",  "users": [  {  "id": 7,  "first\_name": "Albert",  "last\_name": "Einstein",  "email": "Einstein@xmail.com",  "description": "I am a Supervisor",  "company": "XYZ University",  "position": "Leuturer",  "avatar": "1492663030.jpg",  "sign": "1492662998.png",  "start": null,  "role": "supervisor",  "code": "8f14e",  "created\_at": "2017-04-20 11:37:38",  "updated\_at": "2017-04-20 14:12:33",  "pivot": {  "student\_id": 8,  "user\_id": 7  }  },  {  "id": 6,  "first\_name": "Michael",  "last\_name": "Faraday",  "email": "Faraday@xmail.com",  "description": "I am a mentor",  "company": "FaradayCompany",  "position": "Owner",  "avatar": "1492665000.jpg",  "sign": "1492665004.png",  "start": null,  "role": "mentor",  "code": "16790",  "created\_at": "2017-04-20 11:31:35",  "updated\_at": "2017-04-20 12:10:05",  "pivot": {  "student\_id": 8,  "user\_id": 6  }  }  ],  "students": [],  "mentors": [  {  "id": 6,  "first\_name": "Michael",  "last\_name": "Faraday",  "email": "Faraday@xmail.com",  "description": "I am a mentor",  "company": "FaradayCompany",  "position": "Owner",  "avatar": "1492665000.jpg",  "sign": "1492665004.png",  "start": null,  "role": "mentor",  "code": "16790",  "created\_at": "2017-04-20 11:31:35",  "updated\_at": "2017-04-20 12:10:05",  "pivot": {  "student\_id": 8,  "user\_id": 6  }  }  ],  "supervisors": [  {  "id": 7,  "first\_name": "Albert",  "last\_name": "Einstein",  "email": "Einstein@xmail.com",  "description": "I am a Supervisor",  "company": "XYZ University",  "position": "Leuturer",  "avatar": "1492663030.jpg",  "sign": "1492662998.png",  "start": null,  "role": "supervisor",  "code": "8f14e",  "created\_at": "2017-04-20 11:37:38",  "updated\_at": "2017-04-20 14:12:33",  "pivot": {  "student\_id": 8,  "user\_id": 7  }  }  ]  } |

### create(object data): User

**Test description:** Test create(Object data) method by input detail of user as a object 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 detail of student as an object parameter. | first\_name = ‘student’,  last\_name = ‘ja’,  description = ‘i'm student’,  email = ‘student@mail.com’,  company = ‘ZZZ company’,  position = ‘Front-end developer’,  password = ‘123456’,  avatar = ‘’,  role = ‘student’,  start = [  epoc = 1493031284 ] | User  {  "id": 15,  "first\_name": "student",  "last\_name": "ja",  "email": "student@mail.com",  "description": "i'm student",  "company": "ZZZ company",  "position": "Front-end developer",  "avatar": "1494171216.png",  "sign": "1494171223.png",  "start": "2017-04-24",  "role": "student",  "code": null,  "created\_at": "2017-05-07 22:33:46",  "updated\_at": "2017-05-07 22:33:46",  "users": [],  "students": [],  "mentors": [],  "supervisors": []  } |
| 2 | Test by input detail of mentor as an object parameter. | first\_name = ‘mentor,  last\_name = ‘ja’,  description = ‘i'm mentor,  email = ‘mentor@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘mentor,  start = [  epoc = 1493031284 ] | User  {  "id": 16,  "first\_name": "mentor”  "last\_name": "ja",  "email": " mentor @mail.com",  "description": "i'm mentor”,  "company": "ZZZ company",  "position": "PM",  "avatar": "1494171216.png",  "sign": "1494171223.png",  "start": null,  "role": "mentor”  "code": null,  "created\_at": "2017-05-07 22:33:46",  "updated\_at": "2017-05-07 22:33:46",  "users": [],  "students": [],  "mentors": [],  "supervisors": []  } |
| 3 | Test by input detail of student as an object parameter. | first\_name = ‘supervisor’,  last\_name = ‘ja’,  description = ‘i'm supervisor’,  email = ‘supervisor@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘supervisor’,  start = [  epoc = 1493031284 ] | User  {  "id": 17,  "first\_name": " supervisor”  "last\_name": "ja",  "email": " supervisor @mail.com",  "description": "i'm supervisor”,  "company": "CAMT",  "position": "supervisor",  "avatar": "1494171216.png",  "sign": "1494171223.png",  "start": null,  "role": "supervisor”  "code": null,  "created\_at": "2017-05-07 22:33:46",  "updated\_at": "2017-05-07 22:33:46",  "users": [],  "students": [],  "mentors": [],  "supervisors": []  } |
| 4 | Test by input detail of teacher as an object parameter. | first\_name = ‘teacher’,  last\_name = ‘ja’,  description = ‘i'm teacher’,  email = ‘teacher@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘teacher’,  start = [  epoc = 1493031284 ] | null |

### createStudent(object data): User

**Test description:** Test createStudent(Object data) method by input detail of student as a object 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 detail of student as an object parameter. | first\_name = ‘student’,  last\_name = ‘ja’,  description = ‘i'm student’,  email = ‘student@mail.com’,  company = ‘ZZZ company’,  position = ‘Front-end developer’,  password = ‘123456’,  avatar = ‘’,  role = ‘student’,  start = [  epoc = 1493031284 ] | User  {  "id": 15,  "first\_name": "student",  "last\_name": "ja",  "email": "student@mail.com",  "description": "i'm student",  "company": "ZZZ company",  "position": "Front-end developer",  "avatar": "1494171216.png",  "sign": "1494171223.png",  "start": "2017-04-24",  "role": "student",  "code": null,  "created\_at": "2017-05-07 22:33:46",  "updated\_at": "2017-05-07 22:33:46",  "users": [],  "students": [],  "mentors": [],  "supervisors": []  } |
| 2 | Test by empty the field of user detail |  | false |

### createSupervisor(object data): User

**Test description:** Test createSupervisor(Object data) method by input detail of supervisor as a object 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 detail of student as a object parameter. | first\_name = ‘supervisor’,  last\_name = ‘ja’,  description = ‘i'm supervisor’,  email = ‘supervisor@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘supervisor’,  start = [  epoc = 1493031284 ] | User  {  "id": 17,  "first\_name": " supervisor”  "last\_name": "ja",  "email": " supervisor @mail.com",  "description": "i'm supervisor”,  "company": "CAMT",  "position": "supervisor",  "avatar": "1494171216.png",  "sign": "1494171223.png",  "start": null,  "role": "supervisor”  "code": null,  "created\_at": "2017-05-07 22:33:46",  "updated\_at": "2017-05-07 22:33:46",  "users": [],  "students": [],  "mentors": [],  "supervisors": []  } |
| 2 | Test by empty the field of user detail |  | false |

### createMentor(object data): User

**Test description:** Test createMentor(Object data) method by input detail of mentor as a object 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 detail of mentor as a object parameter. | first\_name = ‘mentor,  last\_name = ‘ja’,  description = ‘i'm mentor,  email = ‘mentor@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘mentor,  start = [  epoc = 1493031284 ] | User  {  "id": 16,  "first\_name": "mentor”  "last\_name": "ja",  "email": " mentor @mail.com",  "description": "i'm mentor”,  "company": "ZZZ company",  "position": "PM",  "avatar": "1494171216.png",  "sign": "1494171223.png",  "start": null,  "role": "mentor”  "code": null,  "created\_at": "2017-05-07 22:33:46",  "updated\_at": "2017-05-07 22:33:46",  "users": [],  "students": [],  "mentors": [],  "supervisors": []  } |
| 2 | Test by empty the field of user detail |  | false |

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

**Test description:** Test update(Object data, int id) method by input detail of user and user id as a object 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 detail of user and user id as a object parameter. | Id = 8,  first\_name = ‘Isaac’,  last\_name = ‘Oldton’,  description = ‘i'm a student’,  email = ‘mentor@mail.com’,  company = ‘XYZ University’,  position = ‘Student’,  password = ‘’,  avatar = ‘pic\_newton’,  start = [  epoc = 1493031284 ] | User  {  "id": 8,  "first\_name": "Isaac",  "last\_name": "Oldton",  "email": "Newton@xmail.com",  "description": "I am a student",  "company": "XYZ University",  "position": "Student",  "avatar": "1492663499.png",  "sign": "1492663504.png",  "start": "2017-04-01",  "role": "student",  "code": null,  "created\_at": "2017-04-20 11:43:19",  "updated\_at": "2017-05-07 22:52:03",  } |
| 2 | Test by empty the field of user detail |  | false |

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

**Test description:** Test getReportByUserId(Int id) method by input user id as a object parameter.

**Test data:** refer to **Appendix A** in “User”, “Project”, “Task” and “TaskLog” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input user id and dates with ‘start’ and ‘end’ as a request parameter. | User => id = 8,  Dates => [  Start = ‘2017-04-13’,  End => ‘2017-04-19’  ] | Project[]  [{  "id": 2,  "name": "Gravity project",  "description": "This is a Gravity project.",  "start": "2017-04-13",  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-20 11:57:27",  "updated\_at": "2017-04-21 21:37:18",  "reports": [{  "logs": [ … ]  "start": "03\/27\/2017",  "end": "04\/02\/2017"  }  }] |
| 2 | Test by input user id and empty dates as a request parameter. | User => id = 8,  Dates => [] | Project[]  [] |

### getStatisticByUsers(User[] users): Object

**Test description:** Test getStatisticByUsers(User[] users) method by input users as a object parameter.

**Test data:** refer to **Appendix A** in “User”, “Project” and “Task” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input users as a object parameter. | User => id = 1,  User => id = 2 | {  Total\_projects = 3,  Total\_tasks = 9  } |
| 2 | Test by input empty parameter | User => null | {  Total\_projects = 0,  Total\_tasks = 0  } |

## ProjectService Class

### show(int id): Project

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

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input project id as a request parameter | id = 2 | Project  {  "id": 2,  "name": "Gravity project",  "description": "This is a Gravity project.",  "start": "2017-04-13",  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-20 11:57:27",  "updated\_at": "2017-04-21 21:37:18"  } |
| 2 | Test by input id that is not existed in the database. | Id = 99 | null |
| 3 | Test by input string parameter | Id = ‘string’ | null |

### all(): Project[]

**Test description:** Test all().

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by no input | - | Project[]  [{  "id": 2,  "name": "Gravity project",  "description": "This is a Gravity project.",  "start": "2017-04-13",  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-20 11:57:27",  "updated\_at": "2017-04-21 21:37:18"  },  ….  ] |

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

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

**Test data:** refer to **Appendix A** in “Project” and “TaskLog” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input project id as a request parameter | id = 2 | TaskLog[]  [{  "id": 29,  "start": "2017-05-05",  "task\_type\_id": 1,  "task\_id": 9,  "created\_at": "2017-05-04 14:54:42",  "updated\_at": "2017-05-04 14:54:42",  },  …  ] |
| 2 | Test by input empty parameter | Id = null | null |
| 3 | Test by input id that is not existed in the database. | Id = 99 | null |

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

**Test description:** Test store(Object request, int userId) method by input id of project and detail of object as a request parameter.

**Test data:** refer to **Appendix A** in “Project” and “TaskLog” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input detail of project and user id as a request parameter | userId = 1,  name = ‘Name’,  description = ‘Description’,  start = ‘2014-04-14’ | Project  {  "name": "Name",  "description": "Description",  "start": "2017-04-14",  "user\_id": 1,  "updated\_at": "2017-05-07 23:37:40",  "created\_at": "2017-05-07 23:37:40",  "id": 9  } |
| 2 | Test by input name and user id as a request parameter (description field is empty) | userId = 1,  name = ‘Name’, | false |
| 3 | Test by input description and user id as a request parameter (name field is empty) | userId = 1,  description = ‘Description, | false |

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

**Test description:** Test update(Object request, int userId) method by input id of project and detail of object as a request parameter.

**Test data:** refer to **Appendix A** in “Project” and “TaskLog” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input detail of project and user id as a request parameter | userId = 1,  name = ‘Edit Name’  description = ‘Description’,  start = ‘2014-04-14’ | Project  {  "name": " Edit Name",  "description": "Description",  "start": "2017-04-14",  "user\_id": 1,  "updated\_at": "2017-05-07 23:3840",  "created\_at": "2017-05-07 23:37:40",  "id": 9  } |
| 2 | Test by input name and user id as a request parameter | userId = 1,  name = ‘Name’, | false |
| 3 | Test by input description and user id as a request parameter | userId = 1,  description = ‘Description, | false |

### destroy(int id): Project

**Test description:** Test destroy(int userId) method by input id of project as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input project id as a request parameter | Id = 1 | Project |
| 2 | Test by input project id that is not existed in the database. | Id = 99 | null |
| 3 | Test by input empty parameter | Id = null | null |

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

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

**Test data:** refer to **Appendix A** in “Project” 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 = 8 | Project[]  [{  "id": 2,  "name": "Gravity project",  "description": "This is a Gravity project.",  "start": "2017-04-13",  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-20 11:57:27",  "updated\_at": "2017-04-21 21:37:18",  "tasks": […]  }, {  "id": 6,  "name": "Test project",  "description": "Test project",  "start": null,  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-26 14:26:54",  "updated\_at": "2017-04-26 14:26:54",  "tasks": […]  }, {  "id": 8,  "name": "Test ",  "description": "test",  "start": null,  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-05-04 10:14:25",  "updated\_at": "2017-05-04 10:14:25",  "tasks": []  }] |
| 2 | Test by input user id that is not existed in the database. | Id = 99 | null |
| 3 | Test by input empty parameter | Id = null | null |

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

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

**Test data:** refer to **Appendix A** in “Project” 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 = 8 | Project[]  [{  "id": 2,  "name": "Gravity project",  "description": "This is a Gravity project.",  "start": "2017-04-13",  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-20 11:57:27",  "updated\_at": "2017-04-21 21:37:18",  "tasks": […]  }, {  "id": 6,  "name": "Test project",  "description": "Test project",  "start": null,  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-04-26 14:26:54",  "updated\_at": "2017-04-26 14:26:54",  "tasks": […]  }, {  "id": 8,  "name": "Test ",  "description": "test",  "start": null,  "image\_id": null,  "user\_id": 8,  "created\_at": "2017-05-04 10:14:25",  "updated\_at": "2017-05-04 10:14:25",  "tasks": []  }] |
| 2 | Test by input user id that is not existed in the database. | Id = 99 | null |
| 3 | Test by input empty parameter | Id = null | null |

## TaskService Class

### store(Object request): Mixed

**Test description:** Test store(Object request) method by detail of task as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input detail of task as a request parameter | name = ‘Name’,  description = ‘Description’,  project\_id = 1,  start = ‘2014-04-14’ | Task  {  "id": 12,  "name": "Name",  "description": "<p>Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:45:49",  "type": {  "id": 1,  "name": "ToDo"  }  } |
| 2 | Test by input name, description and project id as a request parameter | name = ‘Name’,  description = ‘Description’,  project\_id = 1 | false |
| 3 | Test by input description and name as a request parameter | name = ‘Name’,  description = ‘Description’, | false |
| 4 | Test by input name as a request parameter | name = ‘Name’, | false |

### update(Object request, int taskId): Mixed

**Test description:** Test update(Object request, int taskId) method by detail of task and task id as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input detail of task as a request parameter | Task\_id = 1,  name = ‘New Name’,  description = ‘New Description’ | Task  {  "id": 12,  "name": " New Name",  "description": "<p> New Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:46:49",  "type": {  "id": 1,  "name": "ToDo"  }  } |
| 2 | Test by input name and task id as a request parameter | Task\_id = 1,  name = ‘New Name’, | false |
| 3 | Test by input task id as a request parameter | Task\_id = 1, | false |

### getTaskLogsByProjectIdAndDates(int projectId, Date start, Date end): Mixed

**Test description:** Test getTaskLogsByUsers(User users) method by input project id. Start and end date as a request parameter.

**Test data:** refer to **Appendix A** in “Task”, “TaskLog” and “Project” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input project id and start date and end date as a request parameter | projectId = 1,  start = 1493031284,  end = 1493031284 | TaskLog[]  [{  "id": 12,  "name": "Name",  "description": "<p>Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:45:49",  "type": {  "id": 1,  "name": "ToDo"  }  },  …  ] |
| 2 | Test by input project id and wrong type date array as a parameter | projectId = 1,  start = ‘string’,  end = ‘string’ | false |
| 3 | Test by input wrong project id and start and end date array as a parameter | projectId = ‘string’,  start = 1493031284,  end = 1493031284 | false |

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

**Test description:** Test getTaskLogsByUsers(User users) method by input list of users as a request parameter.

**Test data:** refer to **Appendix A** in “Task”, “TaskLog” and “User” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input users as a request parameter | [  [Id = 1]  ] | TaskLog[]  [{  "id": 12,  "name": "Name",  "description": "<p>Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:45:49",  "type": {  "id": 1,  "name": "ToDo"  }  },  …  ] |
| 2 | Test by input empty array as a parameter | [] | null |

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

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

**Test data:** refer to **Appendix A** in “Task”, “TaskLog” and “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 | TaskLog[]  [{  "id": 12,  "name": "Name",  "description": "<p>Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:45:49",  "type": {  "id": 1,  "name": "ToDo"  }  },  …  ] |
| 2 | Test by input empty parameter | Id = null | null |
| 3 | Test by input id that is not existed in the database. | Id = 99 | null |

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

**Test description:** Test getTasksByProjectID(int Id) method by input id of project as a request parameter.

**Test data:** refer to **Appendix A** in “Task”, “TaskLog” and “Project” content.

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input project id as a request parameter | Id = 1 | Task[]  [{  "id": 12,  "name": "Name",  "description": "<p>Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:45:49",  "type": {  "id": 1,  "name": "ToDo"  }  },  …  ] |
| 2 | Test by input empty parameter | Id = null | null |
| 3 | Test by input id that is not existed in the database. | Id = 99 | null |

### getTaskByTaskID(int id): Task

**Test description:** Test getTaskByTaskID(int Id) method by input id of task as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input task id as a request parameter | Id = 1 | Task  {  "id": 1,  "name": "Task1",  "description": "<p>Test task 1</p>",  "start": "2017-04-13",  "image\_id": null,  "task\_type\_id": 2,  "project\_id": 2,  "created\_at": "2017-04-20 12:00:13",  "updated\_at": "2017-04-24 17:58:53",  "comments": [ … ]  } |
| 3 | Test by input empty parameter | Id = null | null |
| 4 | Test by input id that is not existed in the database. | Id = 99 | null |

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

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

**Test data:** refer to **Appendix A** in “Task” and “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 | Task[]  [{  "id": 12,  "name": "Name",  "description": "<p>Description</p>",  "start": "2017-04-14",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 1,  "created\_at": "2017-05-07 23:45:49",  "updated\_at": "2017-05-07 23:45:49",  "type": {  "id": 1,  "name": "ToDo"  }  },  …  ] |
| 2 | Test by input empty parameter | Id = null | null |
| 3 | Test by input id that is not existed in the database. | Id = 99 | null |

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

**Test description:** Test changeTo(Object request, int Id) method by input id of task and detail of task as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input task id and type is todo as a request parameter | Id = 1,  Type = ‘ToDo’ | Task  {  "id": 1,  "name": "Task1",  "description": "<p>Test task 1</p>",  "start": "2017-04-13",  "image\_id": null,  "task\_type\_id": 1,  "project\_id": 2,  "created\_at": "2017-04-20 12:00:13",  "updated\_at": "2017-04-24 17:58:53",  "comments": [ … ]  } |
| 2 | Test by input task id and type is doing as a request parameter | Id = 1  Type = ‘Doing’ | Task  {  "id": 1,  "name": "Task1",  "description": "<p>Test task 1</p>",  "start": "2017-04-13",  "image\_id": null,  "task\_type\_id": 2,  "project\_id": 2,  "created\_at": "2017-04-20 12:00:13",  "updated\_at": "2017-04-24 17:59:53",  "comments": [ … ]  } |
| 3 | Test by input task id and type is done as a request parameter | Id = 1,  Type = ‘Done’ | Task  {  "id": 1,  "name": "Task1",  "description": "<p>Test task 1</p>",  "start": "2017-04-13",  "image\_id": null,  "task\_type\_id": 3,  "project\_id": 2,  "created\_at": "2017-04-20 12:00:13",  "updated\_at": "2017-04-24 17:59:59",  "comments": [ … ]  } |

## CommentService Class

### comment(Object request): Mixed

**Test description:** Test comment(Object request) method by input detail of comment as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input task id and text of comment as a request parameter | Text = ‘comment….’,  Task\_id = 1 | Comment  {  "text": "Comment...",  "user\_id": 8,  "task\_id": 1,  "updated\_at": "2017-05-08 00:08:35",  "created\_at": "2017-05-08 00:08:35",  "id": 18  } |
| 2 | Test by input empty data parameter | Text = ‘’,  Task\_id = 1 | false |

### getComment(int id): Comment

**Test description:** Test getComment(int Id) method by input id of comment as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input comment id as a request parameter | Id = 1 | Comment  {  "id": 1,  "text": "<p>Test comment by Supervisor</p>",  "user\_id": 7,  "task\_id": 1,  "created\_at": "2017-04-20 12:13:56",  "updated\_at": "2017-04-20 12:13:56",  } |
| 2 | Test by input id that is not existed in the database. | Id = 99 | null |

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

**Test description:** Test getCommentsByTask(int Id) method by input id of task as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input task id as a request parameter | Id = 1 | Comment[]  [  {  "id": 1,  "text": "<p>Test comment by Supervisor</p>",  "user\_id": 7,  "task\_id": 1,  "created\_at": "2017-04-20 12:13:56",  "updated\_at": "2017-04-20 12:13:56"  },  {  "id": 9,  "text": "<p><i>Test comment 1</i></p>",  "user\_id": 8,  "task\_id": 1,  "created\_at": "2017-04-24 17:58:25",  "updated\_at": "2017-04-24 17:58:25”  }] |
| 2 | Test by input id that is not existed in the database. | Id = 99 | [] |

### destroy(int id): Comment

**Test description:** Test destroy(int Id) method by input id of comment as a request parameter.

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

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

**Test case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Description** | **Input** | **Expected** |
| 1 | Test by input comment id as a request parameter | Id = 1 | Comment |
| 2 | Test by input id that is not existed in the database. | Id = 99 | null |

# **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.  - The information of the user is updated in the database. |
|  | 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.  - The information of the user is updated in the database. |
|  | 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.  - The information of the user is updated in the database. |
|  | 2 | Register with an account with existing email “Einstein@xmail.com” | 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 system displays an error message “The email has already been taken” |
|  | 2 | Register with incorrect email constraint | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein  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 system displays an error message “The email must be a valid email address”. |
|  | 2 | Register with empty first\_name field. (required) | first\_name: ""  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 system displays an error message “The first name field is required”. |
|  | 2 | Register with empty last\_name field. (required) | first\_name: "Albert"  last\_name: ""  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 system displays an error message “The last name field is required”. |
|  | 2 | Register with empty email field. (required) | first\_name: "Albert"  last\_name: "Einstein"  email: “”  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 system displays an error message “The email field is required”. |
|  | 2 | Register with empty password field. (required) | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein@xmail.com  password: ""  description: "I am a Supervisor"  company: "XYZ University"  position: "Lecturer"  avatar: "1492663030.jpg"  sign: "1492662998.png"  start: “null”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The password field is required”. |
|  | 2 | Register with empty start field. (required) | 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: “”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The start field is required”. |
|  | 2 | Register with empty description field. (required) | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein@xmail.com  password: "einstein1"  description: ""  company: "XYZ University"  position: "Lecturer"  avatar: "1492663030.jpg"  sign: "1492662998.png"  start: “null”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The description field is required”. |
|  | 2 | Register with empty company field. (required) | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein@xmail.com  password: "einstein1"  description: "I am a Supervisor"  company: ""  position: "Lecturer"  avatar: "1492663030.jpg"  sign: "1492662998.png"  start: “null”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The company field is required”. |
|  | 2 | Register with empty position field. (required) | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein@xmail.com  password: "einstein1"  description: "I am a Supervisor"  company: "XYZ University"  position: ""  avatar: "1492663030.jpg"  sign: "1492662998.png"  start: “null”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The position field is required”. |
|  | 2 | Register with empty avatar field. (required) | first\_name: "Albert"  last\_name: "Einstein"  email: Einstein@xmail.com  password: "einstein1"  description: "I am a Supervisor"  company: "XYZ University"  position: "Lecturer"  avatar: ""  sign: "1492662998.png"  start: “null”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The avatar field is required”. |
|  | 2 | Register with empty sign field. (required) | 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: ""  start: “null”  code: "8f14e"  role: "supervisor" | - The system displays an error message “The sign field is required”. |
|  | 3 | Click “Cancel” button | Cancel | - The system discards the registration. |

### 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  - The information of the user is updated in the database. |
|  | 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  - The information of the user is updated in the database. |
|  | a.2, b.2 | Edit profile with incorrect email constraint | first\_name: "nIsaac"  last\_name: "nNewton"  email: nNewton  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 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: ""  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 an error message “The first name field is required”. |
|  | a.2, b.2 | Edit profile with empty last name field (required) | first\_name: "nIsaac"  last\_name: ""  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 an error message “The email field is required”. |
|  | a.2 | Edit profile with empty start field (required) | 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: "” | The system displays an error message “The start field is required”. |
|  | a.2, b.2 | Edit profile with empty description field (required) | first\_name: "nIsaac"  last\_name: "nNewton"  email: nNewton@xmail.com  password: "nnewton1"  description: ""  company: "XYZ University"  position: "Student"  avatar: "1592663499.png"  sign: "1592663504.png"  start: "2017-04-02” | The system displays an error message “The description field is required”. |
|  | a.2, b.2 | Edit profile with empty company field (required) | first\_name: "nIsaac"  last\_name: "nNewton"  email: nNewton@xmail.com  password: "nnewton1"  description: "I am a new student"  company: ""  position: "Student"  avatar: "1592663499.png"  sign: "1592663504.png"  start: "2017-04-02” | The system displays an error message “The company field is required”. |
|  | a.2, b.2 | Edit profile with empty company field (required) | first\_name: "nIsaac"  last\_name: "nNewton"  email: nNewton@xmail.com  password: "nnewton1"  description: "I am a new student"  company: "XYZ University"  position: ""  avatar: "1592663499.png"  sign: "1592663504.png"  start: "2017-04-02” | The system displays an error message “The position field is required”. |
|  | a3, b3 | Click “Cancel” button | Cancel | - The system discards to edit the profile. |

### 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 name: "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 id: 8 | 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.  - The information of the project is updated in the database. |
|  | 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. |
|  | 3 | Click “Cancel” button | Cancel | - The system discards to create a project. |

### 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.  - The information of the project is updated in the database. |
|  | 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. |
|  | 3 | Click “Cancel” button | Cancel | - The system discards to edit the project. |

### 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”  - The information of the project is deleted in the database. |
|  | 2 | Click “Cancel” button | Cancel | - The system discards to delete the profile. |

### 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.  - The information of the task is updated in the database.  - 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.” |
|  | 3 | Click “Cancel” button | Cancel | - The system discards to create a task. |

### 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.  - The information of the task is updated in the database.  - 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.” |
|  | 3 | Click “Cancel” button | Cancel | - The system discards to edit the task. |

### 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”  - The information of the task is deleted in the database. |
|  | 2 | Click “Cancel” button | Cancel | - The system discards to delete a task. |

### 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 | Test to change the status on LIST tab. | Start date: "2017-04-01" | - The status of task is changed to “Doing”.  - The information of the task is updated in the database.  - The start date is displayed on a task list. |
|  | b4 | Test to change the status on CARD tab. | Start date: "2017-04-01" | - The status of task is changed to “Doing”.  - The information of the task is updated in the database.  - The start date is displayed on a task list. |
|  | c4 | Test to change the status on task detail page. | Start date: "2017-04-01" | - The status of task is changed to “Doing”.  - The information of the task is updated in the database.  - 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 | Test to 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 | Test to drag a task card and drop it to the same column | - | The task is still in the current column |
|  | a4, b4, c4 | Click “Cancel” button | Cancel | - The system discards to change the status of a task. |

## **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”  - The information of the comment is updated in the database. |
|  | 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”  - The information of the comment is deleted in the database. |
|  | 2 | Click “Cancel” button | Cancel | - The system discards to delete a comment |

### View a activities of tasks in a specific project

**Description:** Test for UC-18. User can view activities of tasks in a specific project.

**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} |

### View activities of all under guidance students

**Description:** Test for UC-19. User can view activities of all under guidance students.

**Precondition:**

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

**Test script**:

1. User clicks “Activities” tab.

**Test Cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | **Test script no.** | **Description** | **Input** | **Expected result** |
|  | 2 | Test to view an activity of all students in a dashboard of Mentor or Supervisor. | - | - System displays list of activities; { Last Thursday at 2:54 PM - Isaac Newton to do Task edited},{ Last Thursday at 2:44 PM - Isaac Newton doing Task3},{04/26/2017 - Isaac Newton to do Task0},{ 04/24/2017 - Isaac Newton doing Task1} |

## **Feature #4**

### Generate a report

**Description:** Test for UC-20. 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-21. 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. |
|  | 3 | Click “Cancel” button | Cancel | - The system discards to print a report. |

## **Feature #5**

### Receive web notification

**Description:** Test for UC-22. 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-23. 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: “You got notification http://se.rajkarj.com” |

# **Appendix A:** database data for testing

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **First name** | **Last name** | **email** | **password** | **role** | **Description** | **Company** | **position** | **Start intership** | **code** | **avatar** | **sign** |
| 1 | Veerapat | In-ongkarn | karjkeng@hotmail.com | karjkeng | Student | Hi!... | Buzzwoo! | Junior Frontend Developer | 09/03/2016 | - | pic\_veerapat | sign\_veerapat |
| 8 | Isaac | Newton | Newton@xmail.com | newton1 | Student | I am a student | XYZ University | Student | 01/04/2017 | - | pic\_newton | sign\_newton |
| 6 | Michael | Faraday | Faraday@xmail.com | faraday1 | Mentor | I am a Mentor | ABCCompany | Owner | - | #16790 | pic\_faraday | sign\_faraday |
| 7 | Albert | Einstein | Einstein@xmail.com | einstein1 | Supervisor | I am a Supervisor | XYZ University | Lecturer | - | #8f14e | pic\_einstein | sign\_einstein |

**User**

**Project**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Start** | **User** | **Task** |
| 1 | 1sr Project | Yeahhh! |  | 1 Veerapat In-ongkarn object |  |
| 2 | Gravity project | This is the gravity project. | 13/04/2017 | 8 Isaac Newton object | Task[1,2,3,4,5,6,9] |
| 3 | Test project | This is a test project. |  | 8 Isaac Newton object | Task[7,8] |
| 4 | Bulb project | This is a bulb project. | 2017-04-23 | 10 Thomas Edison object |  |

**Task**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Start** | **Task\_type** | **Project** | **TaskLog** | **Comment** |
| 1 | Task1 | This is a task 1 | 2017-04-13 | Doing | Project id 2 | TaskLog[7,19] | Comment[1,9] |
| 2 | Task2 | This is a task 2 | 2017-04-16 | Doing | Project id 2 | TaskLog[2,9] |  |
| 3 | Task3 | This is a task 3 | 2017-04-18 | ToDo | Project id 2 | TaskLog[3] |  |
| 4 | Task4 | This is a task 4 | 2017-04-21 | Doing | Project id 2 | TaskLog[4,12] |  |
| 5 | Task5 | This is a task 5 | 2017-04-23 | Done | Project id 2 | TaskLog[5,13,14] |  |
| 6 | Test task |  | 2017-04-29 | ToDo | Project id 2 | TaskLog[11] |  |
| 7 | Test task 1 | This is a test task 1 | 2017-04-23 | Doing | Project id 4 | TaskLog[15,17] | Comment[7] |
| 8 | Test task 2 | This is a test task 2 | 2017-04-24 | ToDo | Project id 4 | TaskLog[16] | Comment[9] |
| 9 | Task 9 | This is a task 9 | 2017-04-23 | ToDo | Project id 2 | TaskLog[18] |  |

**TaskLog**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Start** | **Task\_type** | **Task** |
| 2 | 2017-04-17 | ToDo | Task Object id 2 |
| 3 | 2017-04-18 | ToDo | Task Object id 3 |
| 4 | 2017-04-21 | ToDo | Task Object id 4 |
| 5 | 2017-04-23 | ToDo | Task Object id 5 |
| 7 | 2017-04-13 | ToDo | Task Object id 1 |
| 9 | 2017-04-16 | Doing | Task Object id 2 |
| 11 | 2017-04-29 | ToDo | Task Object id 6 |
| 12 | 2017-04-22 | Doing | Task Object id 4 |
| 13 | 2017-04-28 | Doing | Task Object id 5 |
| 14 | 2017-04-01 | Done | Task Object id 5 |
| 15 | 2017-04-23 | ToDo | Task Object id 7 |
| 16 | 2017-04-24 | ToDo | Task Object id 8 |
| 17 | 2017-04-23 | Doing | Task Object id 7 |
| 18 | 2017-04-23 | ToDo | Task Object id 9 |
| 19 | 2017-04-24 | Doing | Task Object id 1 |

**Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Text** | **User** | **Task** |
| 1 | Test comment by Supervisor | 7 Albert Einstein Object | Task Object id 1 |
| 7 | Test comment at test task 1 | 10 Thomas Edison Object | Task Object id 7 |
| 8 | Test comment on task 6 | 8 Isaac Newton Object | Task Object id 9 |
| 9 | Test comment 1 | 8 Isaac Newton Object | Task Object id 1 |

**Notify**

|  |  |  |
| --- | --- | --- |
| **ID** | **Object\_id** | **Type** |
| 2 | 7 | TASK |
| 4 | 9 | TASK |
| 6 | 1 | COMMENT |
| 9 | 12 | TASK |
| 10 | 13 | TASK |
| 11 | 14 | TASK |
| 15 | 17 | TASK |
| 16 | 7 | COMMENT |
| 18 | 9 | COMMENT |
| 19 | 19 | TASK |