**WIL Report Management System**

Test record

­­­

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- TestRecord\_V.0.1.docx | Created document  - Add Chapter I  - Add Chapter II - Add Chapter III  - Add Chapter IV  - Add Chapter V | | Draft | 27-04-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- TestPlan\_V.0.2.docx | - Edit Chapter II  - Edit Chapter III  - Edit Chapter IV  - Edit Chapter V | | Draft | 28-02-2017 | PS,VI | PS,VI | PS,VI |

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

# **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 | Responsibility**

## **Test Responsibility**

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

# **Chapter III | Unit test**

## UserService Class

### getUserById(int id): User

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user id as a request parameter | id = 1 | Get a user object | Get a user object | Pass |
| Test by input wrong-id parameter | Id = 99 | Get an empty object | Get an empty object | Pass |
| Test by input string parameter | Id = ‘string’ | Get an empty object | Get an empty object | Pass |
| Test by input empty parameter | empty object | Get an empty object | Get an empty object | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user id as a request parameter | id = 1 | Get a list of notifies object | Get a list of notifies object | Pass |
| Test by input wrong-id parameter | Id = 99 | Get an empty object | Get an empty object | Pass |
| Test by input string parameter | Id = ‘string’ | Get a false value | Get a false value | Pass |
| Test by input empty parameter | empty object | Get a false value | Get a false value | Pass |

### getMyStatistic(int userId): Object

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input my user id as a request parameter | id = 8 | total\_projects = 1,  total\_tasks = 7 | total\_projects = 1,  total\_tasks = 7 | Pass |
| Test by input user id as a request parameter | Id = 1 | total\_projects = 1,  total\_tasks = 0 | total\_projects = 1,  total\_tasks = 0 | Pass |
| Test by input wrong-id parameter | Id = 99 | total\_projects = 0,  total\_tasks = 0 | total\_projects = 0,  total\_tasks = 0 | Pass |
| Test by input string parameter | Id = ‘string’ | total\_projects = 0,  total\_tasks = 0 | total\_projects = 0,  total\_tasks = 0 | Pass |
| Test by input empty parameter | empty object | total\_projects = 0,  total\_tasks = 0 | total\_projects = 0,  total\_tasks = 0 | Pass |

### getStatisticByUserID(int id): Object

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input my user id as a request parameter | id = 8 | total\_projects = 1,  total\_tasks = 7 | total\_projects = 1,  total\_tasks = 7 | Pass |
| Test by input user id as a request parameter | Id = 1 | total\_projects = 1,  total\_tasks = 0 | total\_projects = 1,  total\_tasks = 0 | Pass |
| Test by input wrong-id parameter | Id = 99 | total\_projects = 0,  total\_tasks = 0 | total\_projects = 0,  total\_tasks = 0 | Pass |
| Test by input string parameter | Id = ‘string’ | total\_projects = 0,  total\_tasks = 0 | total\_projects = 0,  total\_tasks = 0 | Pass |
| Test by input empty parameter | empty object | total\_projects = 0,  total\_tasks = 0 | total\_projects = 0,  total\_tasks = 0 | Pass |

### getUserByCode(String code): User

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user code as a request parameter | code = 16790 | Get a user object code = 16790 | Get a user object code = 16790 | Pass |
| Text by input wrong-code parameter | code = 11111 | Get an empty object | Get an empty object | Pass |
| Test by input empty parameter | empty object | Get an empty object | Get an empty object | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input student id as a request parameter | id = 8 | Get a mentors of student id = 8. Size = 1 | Get a mentors of student id = 8. Size = 1 | Pass |
| Test by input student id as a request parameter | Id = 1 | Get an empty array. Size = 0 | Get an empty array. Size = 0 | Pass |
| Test by input empty parameter | empty object | Get an empty array. | Get an empty array. | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input student id as a request parameter | id = 8 | Get a mentors of student id = 8. Size = 1 | Get a mentors of student id = 8. Size = 1 | Pass |
| Test by input student id as a request parameter | Id = 1 | Get an empty array. Size = 0 | Get an empty array. Size = 0 | Pass |
| Test by input empty parameter | empty object | Get an empty array. | Get an empty array. | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input id of mentor or supervisor as a parameter and input id of student as a parameter. | id = 12, sid = 8 | Get a user object. | Get a user object. | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input id of mentor or supervisor as a parameter and input id of student as a parameter. | id = 12, sid = 8 | Get a user object. | Get a user object. | Pass |

### create(object data): User

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input detail of student as a object parameter. | first\_name = ‘student’,  last\_name = ‘ja’,  description = ‘i'm student’,  email = ‘student@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘student’,  start = [  epoc = 1493031284 ] | Get a student object. | Get a student object. | Pass |
| 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 ] | Get a mentor object. | Get a mentor object. | Pass |
| 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 ] | Get a supervisor object. | Get a supervisor object. | Pass |
| Test by input detail of teacher as a object parameter. | role = ‘teacher’ | Get a empty object | Get a empty object | Pass |

### createStudent(object data): User

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input detail of student as a object parameter. | first\_name = ‘student’,  last\_name = ‘ja’,  description = ‘i'm student’,  email = ‘student@mail.com’,  company = ‘’,  position = ‘’,  password = ‘123456’,  avatar = ‘’,  role = ‘student’,  start = [  epoc = 1493031284 ] | Get a student object. | Get a student object. | Pass |

### createSupervisor(object data): User

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| 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 ] | Get a supervisor object. | Get a supervisor object. | Pass |

### createMentor(object data): User

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| 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 ] | Get a mentor object. | Get a mentor object. | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| 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 ] | Get a user edited object. | Get a user edited object. | Pass |
| Test by input detail of user and wrong user id as an object parameter. | Id = ‘eight’,  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 ] | Get a false value. | Get a false value. | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| 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’  ] | Get a projects object with reports list. | Get a projects object with reports list. | Pass |
| Test by input user id and empty dates as a request parameter. | User => id = 8,  Dates => [] | Get a projects object with empty reports. | Get a projects object with empty reports. | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input users as a object parameter. | User => id = 1,  User => id = 2 | Total\_projects = 3,  Total\_tasks = 9 | Total\_projects = 3,  Total\_tasks = 9 | Pass |
| Test by input empty parameter | Empty user | Total\_projects = 0,  Total\_tasks = 0 | Total\_projects = 0,  Total\_tasks = 0 | Pass |

## ProjectService Class

### show(int id): Project

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input project id as a request parameter | id = 1 | Get a project object | Get a project object | Pass |
| Test by input wrong-id parameter | Id = 99 | Get an empty object | Get an empty object | Pass |
| Test by input string parameter | Id = ‘string’ | Get an empty object | Get an empty object | Pass |
| Test by input empty parameter | empty object | Get an empty object | Get an empty object | Pass |

### all(): Project[]

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by no input | - | Get a list of projects object | Get a list of projects object | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input project id as a request parameter | id = 8 | Get a list of task logs object | Get a list of task logs object | Pass |
| Test by input empty parameter | empty data | Get an empty array | Get an empty array | Pass |
| Test by input wrong project id parameter | Id = 99 | Get an empty array | Get an empty array | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input detail of project and user id as a request parameter | userId = 1,  name = ‘Name’,  description = ‘Description’,  start = ‘2014-04-14’ | Get a new project object | Get a new project object | Pass |
| Test by input name and user id as a request parameter | userId = 1,  name = ‘Name’, | Get false value | Get false value | Pass |
| Test by input description and user id as a request parameter | userId = 1,  description = ‘Description, | Get false value | Get false value | Pass |
| Test by input user id as a request parameter | userId = 1 | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input detail of project and user id as a request parameter | userId = 1,  name = ‘Name’,  description = ‘Description’,  start = ‘2014-04-14’ | Get a project edited object | Get a project edited object | Pass |
| Test by input name and user id as a request parameter | userId = 1,  name = ‘Name’, | Get false value | Get false value | Pass |
| Test by input description and user id as a request parameter | userId = 1,  description = ‘Description, | Get false value | Get false value | Pass |
| Test by input user id as a request parameter | userId = 1 | Get false value | Get false value | Pass |

### destroy(int id): Project

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input project id as a request parameter | Id = 1 | Get a project deleted object | Get a project deleted object | Pass |
| Test by input project id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user id as a request parameter | Id = 1 | Get a list of projects object | Get a list of projects object | Pass |
| Test by input user id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user id as a request parameter | Id = 1 | Get a list of projects object | Get a list of projects object | Pass |
| Test by input user id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

## TaskService Class

### store(Object request): Task

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input detail of task as a request parameter | name = ‘Name’,  description = ‘Description’,  project\_id = 1,  start = ‘2014-04-14’ | Get a new task object | Get a new task object | Pass |
| Test by input name, description and project id as a request parameter | name = ‘Name’,  description = ‘Description’,  project\_id = 1 | Get false value | Get false value | Pass |
| Test by input description and name as a request parameter | name = ‘Name’,  description = ‘Description’, | Get false value | Get false value | Pass |
| Test by input name as a request parameter | name = ‘Name’, | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input detail of task as a request parameter | Task\_id = 1,  name = ‘New Name’,  description = ‘New Description’ | Get a task edited object | Get a task edited object | Pass |
| Test by input name and task id as a request parameter | Task\_id = 1,  name = ‘New Name’, | Get false value | Get false value | Pass |
| Test by input task id as a request parameter | Task\_id = 1, | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input project id and start date and end date as a request parameter | projectId = 1,  start = 1493031284,  end = 1493031284 | Get a list of task logs object | Get a list of task logs object | Pass |
| Test by input project id and wrong type date array as a parameter | projectId = 1,  start = ‘string’,  end = ‘string’ | Get false value | Get false value | Pass |
| Test by input wrong project id and start and end date array as a parameter | projectId = ‘string’,  start = 1493031284,  end = 1493031284 | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input users as a request parameter | [  [Id = 8]  ] | Get a list of task logs object | Get a list of task logs object | Pass |
| Test by input empty array as a parameter | [] | Get a empty list | Get a empty list | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user id as a request parameter | Id = 1 | Get a list of task logs object | Get a list of task logs object | Pass |
| Test by input user id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input wrong id user parameter | Id = 99 | Get a empty object | Get a empty object | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input project id as a request parameter | Id = 1 | Get a list of tasks object | Get a list of tasks object | Pass |
| Test by input project id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input wrong id project parameter | Id = 99 | Get an empty object | Get an empty object | Pass |

### getTaskByTaskID(int id): Task

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input task id as a request parameter | Id = 1 | Get a task object | Get a task object | Pass |
| Test by input task id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input wrong id task parameter | Id = 99 | Get an empty object | Get an empty object | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input user id as a request parameter | Id = 1 | Get a list of tasks object | Get a list of tasks object | Pass |
| Test by input user id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input wrong id user parameter | Id = 99 | Get an empty object | Get an empty object | Pass |

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

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input task id and type is todo as a request parameter | Id = 1,  Type = ‘ToDo’ | Get a task type ‘ToDo’ object | Get a task type ‘ToDo’ object | Pass |
| Test by input task id and type is doing as a request parameter | Id = 3,  Type = ‘Doing’ | Get a task type ‘Doing’ object | Get a task type ‘Doing’ object | Pass |
| Test by input task id and type is done as a request parameter | Id = 1,  Type = ‘Done’ | Get a task type ‘Done’ object | Get a task type ‘Done’ object | Pass |

## CommentService Class

### comment(Object request): Comment

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input task id and text of comment as a request parameter | Text = ‘comment….’,  Task\_id = 1 | Get a comment object | Get a comment object | Pass |
| Test by input text of comment as a string parameter | Text = ‘comment….’ | Get false value | Get false value | Pass |
| Test by input task id as a string parameter | Task\_id = 1 | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |

### getComment(int id): Comment

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input comment id as a request parameter | Id = 1 | Get a comment object | Get a comment object | Pass |
| Test by input comment id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input wrong id comment parameter | Id = 99 | Get an empty object | Get an empty object | Pass |

### getCommentsByTask(int id): Comment

**Test by:** Virapat In-Ongkarn

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input task id as a request parameter | Id = 1 | Get a list of comments object | Get a list of comments object | Pass |
| Test by input task id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input task id comment parameter | Id = 99 | Get a empty object | Get a empty object | Pass |

### destroy(int id): Comment

**Test by: Virapat In-Ongkarn**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Input** | **Expected** | **Actual result** | **Pass/Fail** |
| Test by input comment id as a request parameter | Id = 1 | Get a comment deleted object | Get a comment deleted object | Pass |
| Test by input comment id as a string parameter | Id = ‘string’ | Get false value | Get false value | Pass |
| Test by input empty parameter | Empty data | Get false value | Get false value | Pass |
| Test by input wrong id comment parameter | Id = 99 | Get an empty object | Get an empty object | Pass |