**WIL Report Management System**

Software Requirement Specification

­­­

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- SRS\_V.0.1.docx | Created document  - Add Chapter I | Introduction   - Purpose  - Scope  - User Characteristic  - Operation Environment  -Design and Implementation  Constraints  - Dependency Document  - Acronyms  - Document format  - Definition  - Add Chapter II | Overall Description  - Project Perspective  - Project Functions  - Functional Requirement - Add Chapter III | Specific Requirement  - Use Case - Add Chapter IV | Software Requirement Specification | | Draft | 10-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.2.docx | - Edit Chapter I | Introduction   - Purpose  - Scope  - User Characteristic  - Operation Environment  -Design and Implementation  Constraints  - Dependency Document  - Acronyms  - Document format  - Definition  - Edit Chapter II | Overall Description  - Project Perspective  - Project Functions  - Functional Requirement - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification | | Draft | 17-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.3.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Add UC | | Draft | 25-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.4.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Edit UC | | Draft | 28-02-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.5.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Edit UC | | Draft | 07-03-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.6.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Edit UC | | Draft | 20-03-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.7.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Edit UC | | Draft | 13-04-2017 | PS,VI | PS,VI | PS,VI |
| WRMS- SRS\_V.0.8.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Edit UC | | Draft | 22-04-2017 | PS,VI | PS,VI | PS,VI |

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

**Table of Contents**

[1.1 Purpose 7](#_Toc480737420)

[1.2 Project Scope 7](#_Toc480737421)

[1.3 Acronyms and definitions Acronyms 7](#_Toc480737422)

[1.4 Definitions 7](#_Toc480737423)

[Chapter II | Overall Description 9](#_Toc480737424)

[2.1 Product perspective 9](#_Toc480737425)

[2.2 Product Functions 9](#_Toc480737426)

[2.3 User Characteristic 9](#_Toc480737427)

[2.4 Operation Environment 9](#_Toc480737428)

[2.5 Design and Implementation constants 9](#_Toc480737429)

[Chapter III | Software Requirement Specification 10](#_Toc480737430)

[3.1 User requirements 10](#_Toc480737431)

[3.2 System requirements 11](#_Toc480737432)

[3.3 User requirement specification with system requirement specification 13](#_Toc480737433)

[Chapter IV | Requirement Specification 19](#_Toc480737434)

[4.1 Use case diagram 19](#_Toc480737435)

[4.2 Use case description 21](#_Toc480737436)

[UC-01 21](#_Toc480737437)

[UC-02 24](#_Toc480737438)

[UC-03 26](#_Toc480737439)

[UC-04 28](#_Toc480737440)

[UC-05 29](#_Toc480737441)

[UC-06 31](#_Toc480737442)

[UC-07 34](#_Toc480737443)

[UC-08 36](#_Toc480737444)

[UC-09 38](#_Toc480737445)

[UC-10 40](#_Toc480737446)

[UC-11 41](#_Toc480737447)

[UC-12 43](#_Toc480737448)

[UC-13 45](#_Toc480737449)

[UC-14 47](#_Toc480737450)

[UC-15 51](#_Toc480737451)

[UC-16 52](#_Toc480737452)

[UC-17 53](#_Toc480737453)

[UC-18 54](#_Toc480737454)

[UC-19 56](#_Toc480737455)

[UC-20 57](#_Toc480737456)

[UC-21 58](#_Toc480737457)

[UC-22 60](#_Toc480737458)

**Table of Figures**

[Figure 1 Use case diagram of User management 19](#_Toc480737388)

[Figure 2 Use case diagram of Project management 19](#_Toc480737389)

[Figure 3 Use case diagram of Progress tracking 20](#_Toc480737390)

[Figure 4 Use case diagram of Report export 20](#_Toc480737391)

[Figure 5 Use case diagram of Notification 21](#_Toc480737392)

[Figure 6 AD-01 Registration 23](#_Toc480737393)

[Figure 7 AD-02 Edit profile 25](#_Toc480737394)

[Figure 8 AD-03 Login 27](#_Toc480737395)

[Figure 9 AD-04 Logout 28](#_Toc480737396)

[Figure 10 AD-05 Add Mentor or Supervisor code 30](#_Toc480737397)

[Figure 11 AD-06 View a task (Role as a student) 32](#_Toc480737398)

[Figure 12 AD-07 View a task (Role as a Mentor or Supervisor) 33](#_Toc480737399)

[Figure 13 AD-08 View statistics of tasks (Role as a student) 34](#_Toc480737400)

[Figure 14 AD-09 View statistics of tasks (Role as a Mentor or Supervisor) 35](#_Toc480737401)

[Figure 15 AD-10 Create a project 37](#_Toc480737402)

[Figure 16 AD-11 Edit a project 39](#_Toc480737403)

[Figure 17 AD-12 Delete a project 40](#_Toc480737404)

[Figure 18 AD-13 Create a task 42](#_Toc480737405)

[Figure 19 AD-14 Update a task 44](#_Toc480737406)

[Figure 20 AD-15 Delete a task 46](#_Toc480737407)

[Figure 21 AD-16 Change a task status (List tab) 48](#_Toc480737408)

[Figure 22 AD-17 Change a task status (Card tab) 49](#_Toc480737409)

[Figure 23 AD-18 Change a task status (in task detail page) 50](#_Toc480737410)

[Figure 24 AD-18 View a comment 51](#_Toc480737411)

[Figure 25 AD-19 Create a comment 52](#_Toc480737412)

[Figure 26 AD-20 Delete a comment 53](#_Toc480737413)

[Figure 27 AD-21 View task activities (Student role) 54](#_Toc480737414)

[Figure 28 AD-22 View task activities (Mentor and Supervisor role) 55](#_Toc480737415)

[Figure 29 AD-23 Generate a weekly report 56](#_Toc480737416)

[Figure 30 AD-24 Print a report 57](#_Toc480737417)

[Figure 31 AD-25 Web notification 59](#_Toc480737418)

[Figure 32 AD-26 Mail notification 60](#_Toc480737419)

**Chapter I | Introduction**

## **Purpose**

The purpose of this software requirement specification document (SRS) is to detail overview for WIL report management web application. The document describes how an application will interact with other system and human. It is also contain a detail description of functional and non-functional requirement. In other words, it describe what system can do, limitations, type of user, parameters, and other requirements which support the development. The software requirement specification provides developers and user to understand each other in structure details.

## **Project Scope**

WIL report management system (WRMS) is a web application for managing tasks of student, WIL weekly report, and comments of mentor and supervisor. All user shall receive a real-time notification via the web application and Email. This application was designed to use in College of Art, Media, and Technology. WRMS supports desktop and mobile device on chrome, Safari, and opera.

## **Acronyms and definitions Acronyms**

URS User Requirement Specification  
SRS Software Requirement Specification  
WRMS WIL Report Management System  
UC Use Case  
CD Class diagram  
AD Activity diagram  
SD Sequence diagram  
UI User interface  
UML Unified Modeling Language

## **Definitions**

|  |  |
| --- | --- |
| Use case | In UML, a complete task of a system that provides a measurable result of value for an actor. More formally, a use case defines a set of use case instances or scenarios [1] |
| Feature | Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a product in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [2] |
| Requirement | (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. (3) A documented representation of a condition or capability as in definition (1) or (2). [2] |
| Specification | Precise description of an activity or work product which serves as basis or input for further activities or work product. A specification can comprise requirements to a product and how they will be solved. Different parts of a specification (e.g., what is to be done, how it will be done) must not be mixed. [2] |
| UML | Unified Modeling Languages. Standardized notation for modeling design descriptions, architectures or scenarios. Not depending on a specific method. Issued and maintained by the object Management Group (OMG). [2] |
| System | A conceptual entity defined by its boundaries. Examples include companies, divisions, and sets of software applications, components, machines, and devices. [1] |

# **Chapter II | Overall Description**

## **Product perspective**

WIL report management system (WRMS) consist of 5 parts which are User Management system, Tasks management system, Comments management system, Report export system, Notification system. When student generate report, the system shall change all list tasks into sentences and ordered by weekly or monthly. Mentor and Supervisor are able to see the student’ progress and also leave some comments. After leaving a comment, the system will send a real-time notification to all involved user.

## **Product Functions**

WRMS is a web application and mobile application which helps student to create WIL reports from lists of tasks to sentences. It also helps mentor and supervisor to see the progress of student. Furthermore, Mentor and supervisor are able to leave their comments on tasks of student.

## **User Characteristic**

WIL report management web application includes 5 types of user characteristics:

1. Student – the user who is able to login, edit their own profile, logout, view and manage tasks, view task statistics, export reports, and receive notifications.
2. Mentor – the user who is able to login, edit their own profile, logout manage comments, view reports, and receive notifications.
3. Supervisor – the user who is able to login, edit their own profile, logout manage comments, view reports, and receive notifications.
4. User – Student, Mentor, and Supervisor
5. Visitor – the user who is able to register to the system.

## **Operation Environment**

WIL report management system is a web application which requires internet connection. The user have to access to the website though internet browser including Internet Explorer 10 (or above), Safari version 9, and opera. The system support windows 8.1 and 10, Mac OS sierra, android 6.0, and iOS 10.

## **Design and Implementation constants**

2.5.1 The application requires internet connection.  
2.5.2 The prototype is available in English.  
2.5.3 Mentor and Supervisor are not able to manage tasks of student.   
2.5.4 Overload data could affect to the performance of web application.

# **Chapter III | Software Requirement Specification**

## **User requirements**

**Feature #1: User management**

1. Visitor can register into the system.
2. User can edit their own profile.
3. User can login into the system.
4. User can logout from the system.
5. Student can add registration code of supervisor and mentor.

**Feature #2: Project management**

1. User can view tasks of a project.
2. User can view the statistics of tasks.
3. Student can create a project.
4. Student can edit a project.
5. Student can delete a project.
6. Student can create a task.
7. Student can edit a task.
8. Student can delete a task.
9. Student can change a status of task.

**Feature #3: Progress tracking**

1. User can view comments.
2. User can create a comment.
3. User can delete their own comment.
4. User can view activities of tasks.

**Feature #4: Report export**

1. Student can generate a weekly report.
2. Student can print a weekly report

**Feature #5: Notification**

1. User can receive an activity notification via Web application.
2. User can receive an activity notification via Email.

## **System requirements**

1. System shall provide UI for the visitor to receive first name, last name, description, company, position, email, password, profile image, signature image, start date, and role of user.
2. System shall validate first name, last name, description, company, position, email, password, start date, and role of user.
3. System shall provide an error message if the input field(s) are blank.
4. System shall provide an error message if an input email is not matched with constraint.
5. System shall redirect to dashboard.
6. System shall provide registration code for supervisor and mentor.
7. System shall connect to database.
8. System saves input data into the database.
9. System shall provide UI for the student to edit first name, last name, description, company, position, email, password, profile image, signature image, and start date.
10. System shall provide UI for the mentor to edit first name, last name, description, company, position, email, password, profile image, and signature image.
11. System shall provide UI for the supervisor to edit first name, last name, description, company, position, email, password, profile image, and signature image.
12. System validates first name, last name, description, company, position, email, password, and start date.
13. System shall provide UI to display an error message if the student edit their own information which the “email” is not matched with constraint.
14. System shall provide UI to display an error message if the student edit their own information which the “Mentor and supervisor code” does not exist in database.
15. System shall provide UI to display an error message if the mentor edit their own information which the “email” is not matched with constraint.
16. System shall provide UI to display an error message if the supervisor edit their own information which the “email” is not matched with constraint.
17. System shall provide an error message if the input field(s) are blank.
18. System shall redirect to profile page.
19. System shall provide UI to login.
20. System shall receive email and password.
21. System shall verify email and password.
22. System shall display an error message if email or password is wrong.
23. System shall navigate the student to access the student dashboard.
24. System shall navigate the mentor to access the mentor dashboard.
25. System shall navigate the supervisor to access the supervisor dashboard.
26. System shall redirect to Login page.
27. System shall provide a button to logout.
28. System shall logout from the system.
29. System shall validate a code.
30. System shall display a confirmation message.
31. System shall link the mentor/supervisor account to the student account.
32. System shall redirect to profile page.
33. System shall update a name of mentor or supervisor on student’s profile page.
34. System shall lock the button.
35. System shall provide UI to display student’s dashboard.
36. System provide UI to display list of projects, a button of creating new project, task statistics, and summary of activities frequency.
37. System shall provide UI to display projects detail page.
38. System shall provide UI to display lists and cards of tasks.
39. System shall display statistics box of tasks on the top of dashboard page.
40. System shall provide list of student in profile page.
41. System shall link to dashboard of selected student.
42. System shall provide “create new project” button in the dashboard.
43. System shall provide UI which receive project name, project description, start date, “create” button, and “cancel” button.
44. System shall validate empty fields.
45. System redirects to the dashboard.
46. System locks the “create” button.
47. System shall provide UI which receive project name, description, start date, “edit” button, and “Cancel” button.
48. System shall lock the “Edit” button.
49. System shall provide UI to display a confirmation message “Are you sure to delete this project?” with “Confirm” and “Cancel” button.
50. System shall delete a project in database.
51. System shall provide UI to display the Task page.
52. System shall provide UI to receive tasks name, task description, start date, “create” button, and “cancel” button.
53. System shall redirect to Task page.
54. System shall create a new task object in the database.
55. System locks the “create” button.
56. System shall provide UI which receive tasks name, task description, “update” button, and “cancel” button.
57. System provides UI which receive tasks name, task description, “update” button, and “cancel” button.
58. System shall provide UI to display a confirmation message “Are you sure to delete this task?” with “Confirm” and “Cancel” button.
59. System shall delete the task object from database.
60. System shall provide UI to select the start date with “Confirm” button, and “Cancel” button.
61. System shall change the task status.
62. System locks the “Confirm” button.
63. System display forbidden sign as a mouse cursor.
64. System locks for changing the status of a task.
65. System shall request the task data from database.
66. System shall provide UI to display a task detail, comment field, and status buttons.
67. System shall provide a comment field with “comment” button.
68. System shall display the comment.
69. System shall provide UI to delete their own comment.
70. System shall display a confirmation message “Are you sure to delete this comment?” .
71. System shall delete the comment in the database.
72. System redirects to the task detail page.
73. System shall provide UI to display task statistics, List tab, Card tab, and activities tab.
74. System shall display activities of tasks since the project was started.
75. System requests tasks data from database.
76. System arranges task details in the report template.
77. System provide UI to display a weekly report.
78. System shall provide “Print” button.
79. System shall provide UI to see a preview of a report with “Print” and “Cancel” button.
80. System shall redirect to weekly report UI.
81. System shall detect a new activity.
82. System shall push an activity to database.
83. System shall receive notify object from database.
84. System shall display an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab.
85. System shall detect a new activity of student who under their guidance.
86. System shall detect a new activity.
87. System shall send request to EmailSender.

## **User requirement specification with system requirement specification**

1. **Visitor can register to the system.**
2. System shall provide UI for the visitor to receive first name, last name, description, company, position, email, password, profile image, signature image, start date, and role of user.
3. System shall validate first name, last name, description, company, position, email, password, start date, and role of user.
4. System shall provide an error message if the input field(s) are blank.
5. System shall provide an error message if an input email is not matched with constraint.
6. System shall redirect to dashboard.
7. System shall provide registration code for supervisor and mentor.
8. System shall connect to database.
9. System saves input data into the database.

**URS-02: Users can edit their own profile.**

SRS-08: System shall save data into the database.

1. System shall provide UI for the student to edit first name, last name, description, company, position, email, password, profile image, signature image, and start date.
2. System shall provide UI for the mentor to edit first name, last name, description, company, position, email, password, profile image, and signature image.
3. System shall provide UI for the supervisor to edit first name, last name, description, company, position, email, password, profile image, and signature image.
4. System validates first name, last name, description, company, position, email, password, and start date.
5. System shall provide UI to display an error message if the student edit their own information which the “email” is not matched with constraint.
6. System shall provide UI to display an error message if the student edit their own information which the “Mentor and supervisor code” does not exist in database.
7. System shall provide UI to display an error message if the mentor edit their own information which the “email” is not matched with constraint.
8. System shall provide UI to display an error message if the supervisor edit their own information which the “email” is not matched with constraint.
9. System shall provide an error message if the input field(s) are blank.
10. System shall redirect to profile page.

**URS-03: Users can login.**

1. System shall provide UI to login.
2. System shall receive email and password.
3. System shall verify email and password.
4. System shall display an error message if email or password is wrong.
5. System shall navigate the student to access the student dashboard.
6. System shall navigate the mentor to access the mentor dashboard.
7. System shall navigate the supervisor to access the supervisor dashboard.
8. System shall redirect to Login page.

**URS-04: Users can logout.**

SRS-26: System shall redirect to Login page.

1. System shall provide a button to logout.
2. System shall logout from the system.

**URS-05: Student can add registration code.**

SRS-08: System shall save data into the database.

1. System shall validate a code.
2. System shall display a confirmation message.
3. System shall link the mentor/supervisor account to the student account.
4. System shall redirect to profile page.
5. System shall update a name of mentor or supervisor on student’s profile page.
6. System shall lock the button.

**URS-06: User can view tasks of a project.**

SRS-07: System shall connect to database.

1. System shall provide UI to display student’s dashboard.
2. System provide UI to display list of projects, a button of creating new project, task statistics, and summary of activities frequency.
3. System shall provide UI to display projects detail page.
4. System shall provide UI to display lists and cards of tasks.

**URS-07: Students can view the statistics of tasks.**

SRS-07: System shall connect to database.

SRS-35: System shall provide UI to display student’s dashboard.

1. System shall display statistics box of tasks on the top of dashboard page.
2. System shall provide list of student in profile page.
3. System shall link to dashboard of selected student.

**URS-08: Student can create a project.**

SRS-07: System shall connect to database.

SRS-35: System shall provide UI to display student’s dashboard.

SRS-08: System shall save data into the database.

1. System shall provide “create new project” button in the dashboard.
2. System shall provide UI which receive project name, project description, start date, “create” button, and “cancel” button.
3. System shall validate empty fields.
4. System redirects to the dashboard.
5. System locks the “create” button.

**URS-09: Student can edit a project.**

SRS-07: System shall connect to database.

SRS-35: System shall provide UI to display student’s dashboard.

SRS-08: System shall save data into the database.

SRS-44: System shall validate empty fields.

SRS-45: System redirects to the dashboard.

1. System shall provide UI which receive project name, description, start date, “edit” button, and “Cancel” button.
2. System shall lock the “Edit” button.

**URS-10: Students can delete a project**.

SRS-07: System shall connect to database.

SRS-35: System shall provide UI to display student’s dashboard.

1. System shall provide UI to display a confirmation message “Are you sure to delete this project?” with “Confirm” and “Cancel” button.
2. System shall delete a project in database.

SRS-45: System redirects to the dashboard.

**URS-11: Students can create a task.**

SRS-07: System shall connect to database.

SRS-08: System shall save data into the database.

SRS-35: System shall provide UI to display student’s dashboard.

SRS-44: System shall validate empty fields.

1. System shall provide UI to display the Task page.
2. System shall provide UI to receive tasks name, task description, start date, “create” button, and “cancel” button.
3. System shall redirect to Task page.
4. System shall create a new task object in the database.
5. System locks the “create” button.

**URS-12: Student can edit a task.**

SRS-07: System shall connect to database.

SRS-08: System shall save data into the database.

SRS-35: System shall provide UI to display student’s dashboard.

SRS-44: System shall validate empty fields.

SRS-51: System shall provide UI to display the Task page.

SRS-53: System shall redirect to Task page.

1. System shall provide UI which receive tasks name, task description, “update” button, and “cancel” button.
2. System provides UI which receive tasks name, task description, “update” button, and “cancel” button.

**URS-13: Student can delete a task.**

SRS-07: System shall connect to database.

SRS-35: System shall provide UI to display student’s dashboard.

SRS-51: System shall provide UI to display the Task page.

SRS-53: System shall redirect to Task page.

1. System shall provide UI to display a confirmation message “Are you sure to delete this task?” with “Confirm” and “Cancel” button.
2. System shall delete the task object from database.

**URS-14: Students can change a status of task.**

SRS-07: System shall connect to database.

SRS-08: System shall save data into the database.

SRS-35: System shall provide UI to display student’s dashboard.

SRS-51: System shall provide UI to display the Task page.

SRS-53: System shall redirect to Task page.

1. System shall provide UI to select the start date with “Confirm” button, and “Cancel” button.
2. System shall change the task status.
3. System locks the “Confirm” button.
4. System display forbidden sign as a mouse cursor.
5. System locks for changing the status of a task.

.

**URS-15: User can view a comment.**

SRS-07: System shall connect to database.

1. System shall request the task data from database.
2. System shall provide UI to display a task detail, comment field, and status buttons.

**URS-16: User can create a comment.**

SRS-07: System shall connect to database.

SRS-08: System shall save data into the database.

1. System shall provide a comment field with “comment” button.
2. System shall display the comment.

**URS-17: User can delete their own comment.**

SRS-07: System shall connect to database.

SRS-08: System shall save data into the database.

1. System shall provide UI to delete their own comment.
2. System shall display a confirmation message “Are you sure to delete this comment?” .
3. System shall delete the comment in the database.
4. System redirects to the task detail page.

**URS-18: User can view activities of projects.**

SRS-07: System shall connect to database.

1. System shall provide UI to display task statistics, List tab, Card tab, and activities tab.
2. System shall display activities of tasks since the project was started.

**URS-19: Student can generate a weekly report.**

SRS-07: System shall connect to database.

1. System requests tasks data from database.
2. System arranges task details in the report template.
3. System provide UI to display a weekly report.

**URS-20: Student can print a weekly report.**

1. System shall provide “Print” button.
2. System shall provide UI to see a preview of a report with “Print” and “Cancel” button.
3. System shall redirect to weekly report UI.

**URS-21: User can receive an activity notification via Web application.**

1. System shall detect a new activity.
2. System shall push an activity to database.
3. System shall receive notify object from database.
4. System shall display an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab.
5. System shall detect a new activity of student who under their guidance.

**URS-22: User can receive an activity notification via Email.**

1. System shall detect a new activity.
2. System shall send request to EmailSender.

# **Chapter IV | Requirement Specification**

## **Use case diagram**

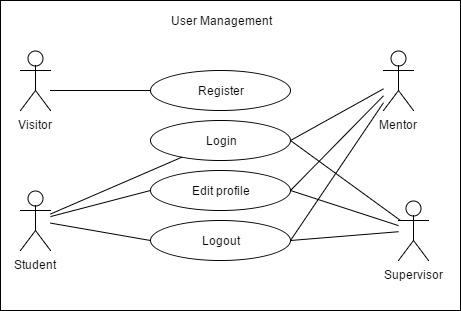


Figure 1 Use case diagram of User management

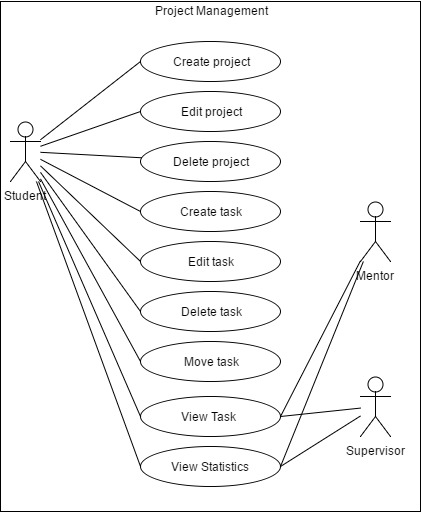


Figure 2 Use case diagram of Project management

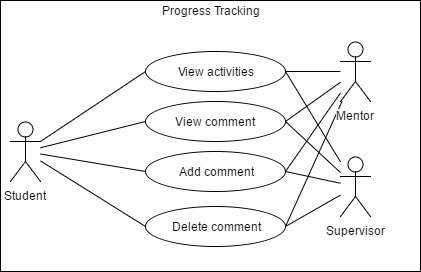


Figure 3 Use case diagram of Progress tracking

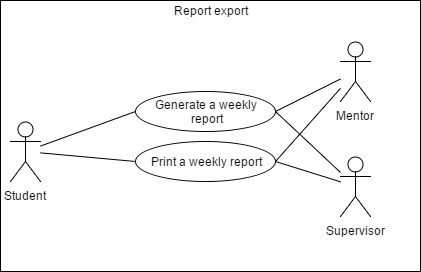


Figure 4 Use case diagram of Report export

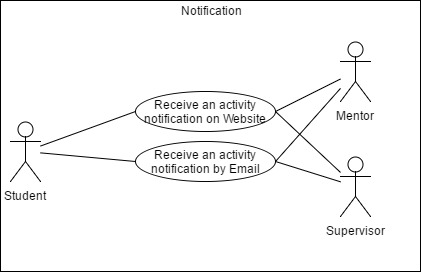


Figure 5 Use case diagram of Notification

## **Use case description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-01 | | |
| **Use Case Name:** | Visitor can register to the system. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 20/04/2017 |
| **Actors:** | Visitor | | |
| **Description:** | Visitor can register to the WIL report management system. | | |
| **Trigger:** | Visitor enter to WIL report management web application. | | |
| **Preconditions:** | - | | |
| **Post conditions:** | Visitor can create a user account. | | |
| **Normal Flow:** | 1. System provides UI for registration. 2. User inputs first name, last name, description, company, position, email, password, profile image, signature image, start date, and role of user. | | |
| **Alternative Flows:** | a.) Student role:   1. System provides UI for the visitor to receive first name, last name, description, company, position, email, password, profile image, signature image, start internship date, and role of user. 2. A visitor inputs first name, last name, description, company, position, email, password profile image, signature image, start date, and student role. 3. A visitor clicks “Create” button. 4. System validates first name, last name, description, company, position, email, password, start date, and student role. 5. System saves data into the database. 6. System redirect to dashboard.   b.) Mentor or supervisor role:   1. System provides UI for the visitor to receive first name, last name, description, company, position, email, password, profile image, signature image, and role of user. 2. A visitor inputs first name, last name, description, company, position, email, password, profile image, signature image, and role. 3. A visitor clicks “Create” button. 4. System validates first name, last name, description, company, position, email, password, and role. 5. System saves input data into the database. 6. System redirect to dashboard. | | |
| **Exceptions:** | From a.4) and b.4) If visitor input incorrect format of email, password or empty forms.   1. System shall provide an error message if the input field(s) are blank. 2. System shall provide an error message if an input email is not matched with constraint.   From a.3) and b.3) If visitor click cancel button:   1. System redirect to Login page. | | |

**Activity diagram of UC-01**

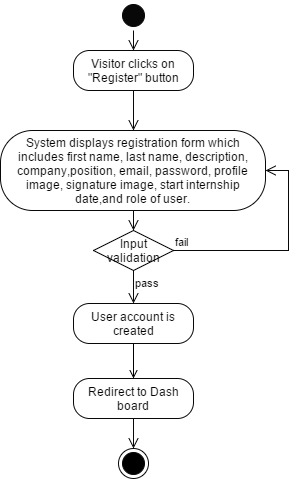


Figure 6 AD-01 Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-02 | | |
| **Use Case Name:** | User can edit their own profile. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 20/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | User can edit their own profile. | | |
| **Trigger:** | User click on edit profile button. | | |
| **Preconditions:** | UC-01, UC-03 System connects to database. | | |
| **Post conditions:** | Profile of user are updated into the database. | | |
| **Normal Flow:** | 1. System checks user id. 2. System provides UI to edit the profile. | | |
| **Alternative Flows:** | From 2)  a.) Student role:   1. System provides UI to receive first name, last name, description, company, position, email, password, profile image, signature image, and start date. 2. Student inputs first name, last name, description, company, position, email, password, profile image, signature image, and start date. 3. Student clicks “update” button. 4. System validates first name, last name, description, company, position, email, password, and start date. 5. System saves data into the database. 6. System redirects to profile page.   b.) Mentor or supervisor role:   1. System provides UI to receive first name, last name, description, company, position, email, password, profile image, and signature image. 2. Student inputs first name, last name, description, company, position, email, password, profile image, and signature image. 3. Student clicks “update” button. 4. System validates first name, last name, description, company, position, email, and password. 5. System saves data into the database. 6. System redirects to profile page. | | |
| **Exceptions:** | From a.4) If student input incorrect format of email:   1. System provides UI to display an error message if the student edit their own information which the “email” is not matched with constraint.   From b.4) If Mentor or supervisor inputs incorrect format of email:   1. System provides UI to display an error message if the mentor or supervisor edit their own information which the “name” is not matched with constraint.   From a.4) and b.4) If user inputs empty forms:   1. System provides UI to display an error message if input fields are empty.   From a.3) and b.3) If a visitor clicks cancel button:   1. System redirects to profile page. | | |

**Activity diagram of UC-02**

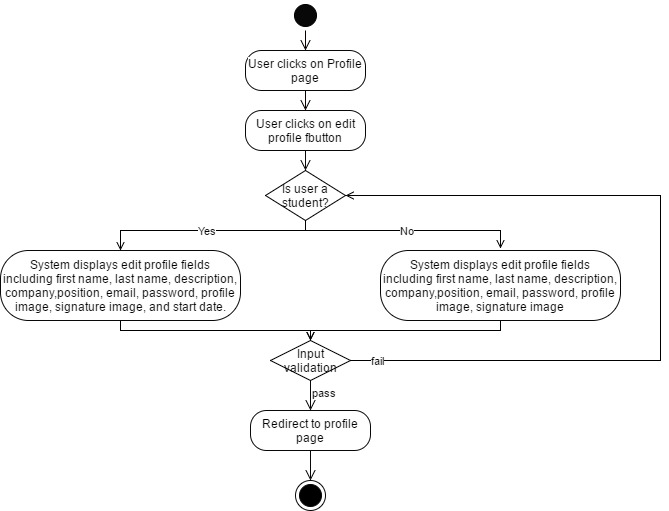


Figure 7 AD-02 Edit profile

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-03 | | |
| **Use Case Name:** | User can login. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 20/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | User can login into the WIL report management system. | | |
| **Trigger:** | User accesses to WIL report management web application.  User clicks on login button. | | |
| **Preconditions:** | User has a user account. | | |
| **Post conditions:** | User are redirected to their role dash board.  The session of a user account is started. | | |
| **Normal Flow:** | 1. System shall provide UI to login 2. User inputs email and password. 3. System shall receive email and password. 4. User click “Login” button. 5. System shall verify email and password. 6. System shall redirect to a dashboard of user. | | |
| **Alternative Flows:** | From 6.)  a.) Student role:   1. System shall navigates the student to access the student dashboard.   b.) Mentor role:   1. System shall navigates the mentor to access the mentor dashboard.   c.) Supervisor role:   1. System shall navigate the supervisor to access the supervisor dashboard | | |
| **Exceptions:** | From 5.) If email and password are invalid:   1. System shall display an error message if email or password is wrong. 2. System shall redirect to Login page. | | |

**Activity diagram of UC-03**

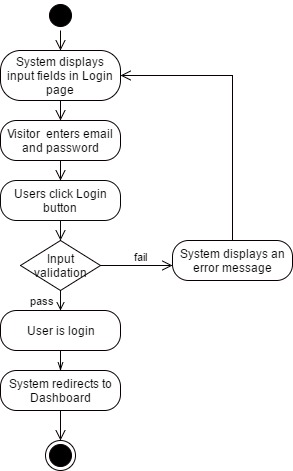


Figure 8 AD-03 Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-04 | | |
| **Use Case Name:** | User can logout. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | User can logout from the WIL report management system. | | |
| **Trigger:** | User click on a logout button. | | |
| **Preconditions:** | UC-03 | | |
| **Post conditions:** | User’s account session is expired. | | |
| **Normal Flow:** | 1. System shall provide a button to logout. 2. User click logout button. 3. System shall logout from the system. 4. System shall redirect to login page. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-04**

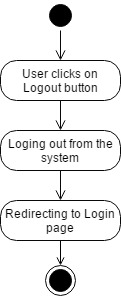


Figure 9 AD-04 Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-05 | | |
| **Use Case Name:** | Student can add registration code. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can add mentor and supervisor code in profile page. | | |
| **Trigger:** | - | | |
| **Preconditions:** | UC-03 Student has to know their mentor and supervisor code.  Student accesses to profile page. | | |
| **Post conditions:** | Account of student connect to mentor and supervisor account. | | |
| **Normal Flow:** | 1. User inputs mentor or supervisor code. 2. User click “+” button. 3. System validates a code. 4. System displays a confirmation message. 5. User clicks “Confirm” 6. System links the mentor/supervisor account to the student account. 7. System redirects to profile page. 8. System shall update a name of mentor or supervisor on student’s profile page. | | |
| **Alternative Flows:** | After 9.) If student would like to add more codes:   1. User click input a code in the same form. 2. Go to normal flow 1.   From 6.) If student slick cancel button:   1. System go back to profile page. | | |
| **Exceptions:** | From 4) If the code is invalid:   1. System locks the button. | | |

**Activity diagram of UC-05**

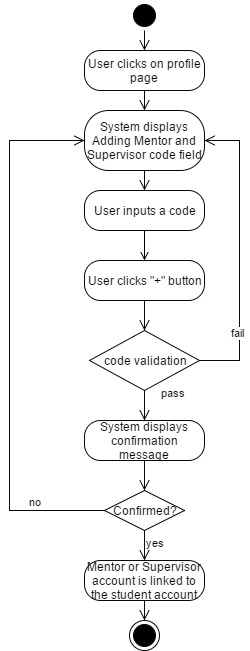


Figure 10 AD-05 Add Mentor or Supervisor code

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-06 | | |
| **Use Case Name:** | User can view tasks of a project. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 27/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can view tasks of each project. | | |
| **Trigger:** | User clicks on a project name. | | |
| **Preconditions:** | UC-03 System connects to database.  Student is in the dashboard.  Mentor and Supervisor are in the student’s dashboard. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | a.) Student role:   1. System provide UI to display list of projects, a button of creating new project, task statistics, and summary of activities frequency. 2. User clicks to a project name. 3. System displays task statistics, list of tasks, card of tasks tab, and tasks activity tab.   b.) Mentor or supervisor role:   1. System provides list of student in profile page. 2. Mentor or supervisor clicks on a student list. 3. System links to dashboard of selected student. 4. User click to a project name. 5. System links to task page. 6. System displays task statistics, list of tasks, card of tasks tab, and tasks activity tab. | | |
| **Alternative Flows:** | From a3 and b6)  If user wants to see a card of tasks:   1. User clicks on “Card” tab. 2. System displays cards of tasks on Kanban board.   From a3 and b6)  If user wants to see activities of tasks:   1. User clicks on “Activities” tab. 2. System displays activities of tasks. | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-06**

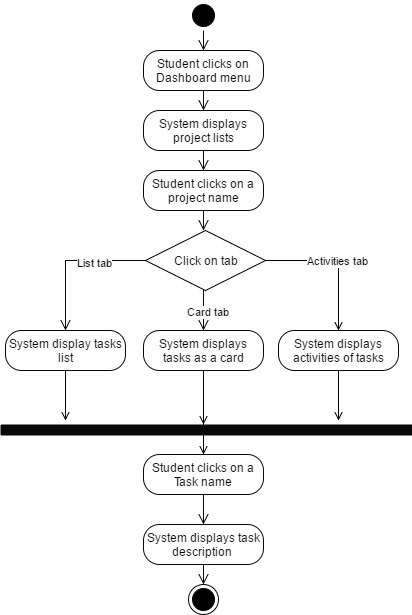


Figure 11 AD-06 View a task (Role as a student)

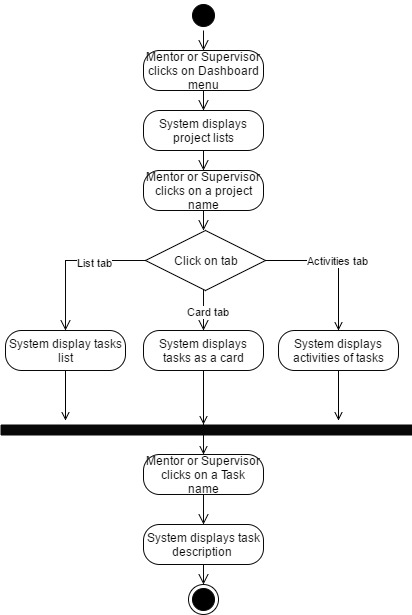


Figure 12 AD-07 View a task (Role as a Mentor or Supervisor)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-07 | | |
| **Use Case Name:** | User can view the statistics of tasks. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student can view statistic of tasks.  Mentor and Supervisor can view statistics of tasks in specific student (who is under their guidance). | | |
| **Trigger:** | User clicks to see a dashboard page. | | |
| **Preconditions:** | UC-03 System connects to database. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | a.) Student role:   1. System provides UI to display a dashboard. 2. System displays statistics box of tasks on the top of dashboard page.   b.) Mentor or supervisor role:   1. System provides list of student in profile page. 2. Mentor or supervisor clicks on a student list. 3. System links to dashboard of selected student. 4. System displays statistics box of tasks on the top of dashboard page. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-07**

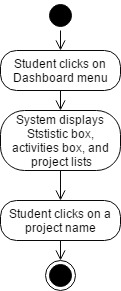


Figure 13 AD-08 View statistics of tasks (Role as a student)

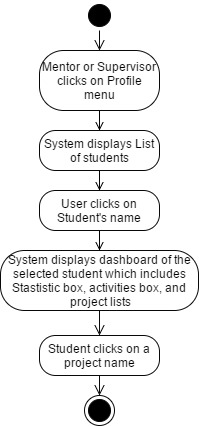


Figure 14 AD-09 View statistics of tasks (Role as a Mentor or Supervisor)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-08 | | |
| **Use Case Name:** | Student can create a project. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can create a new project. | | |
| **Trigger:** | Student click on “create new project” button. | | |
| **Preconditions:** | UC-03 Student is in a dashboard. | | |
| **Post conditions:** | The project object is created.  System redirect project detail page. | | |
| **Normal Flow:** | 1. System provide “create new project” button in the dashboard. 2. Student click on “create new project” button. 3. System shall provide UI which receive project name, project description, start date, “create” button, and “cancel” button. 4. User inputs project name, description, and start date. 5. User clicks on create button. 6. System validates empty fields. 7. System redirects to the dashboard. | | |
| **Alternative Flows:** | From 3.) If s student clicks on “cancel” button:   1. System redirects to the dashboard. | | |
| **Exceptions:** | From 5.) If there are no input data on project name, and description field:   1. System locks the “create” button. | | |

**Activity diagram of UC-08**

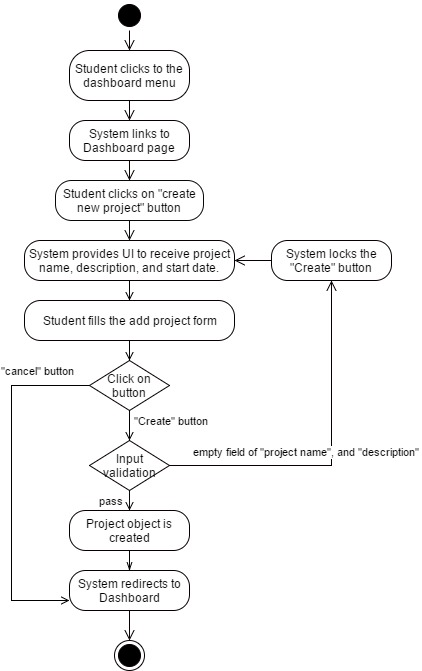


Figure 15 AD-10 Create a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-09 | | |
| **Use Case Name:** | Student can edit a project. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can edit a project. | | |
| **Trigger:** | Student click on “edit” button. | | |
| **Preconditions:** | UC-03  System connect to database.  There is at least 1 project in the project list.  Student is in a dashboard. | | |
| **Post conditions:** | A project object is updated. | | |
| **Normal Flow:** | 1. System provides UI to display student’s dashboard. 2. Student clicks on “edit project” button. 3. System provides UI which receive project name, description, start date, “edit” button, and “Cancel” button. 4. User inputs project name, description, and start date data. 5. User clicks on “edit” button. 6. System validates empty field. 7. System shall save data to database. | | |
| **Alternative Flows:** | From 5.) If student clicks in “Cancel” button:   1. System redirects to the dashboard. | | |
| **Exceptions:** | From 6.) If there is an empty field:   1. System locks the “Edit” button. | | |

**Activity diagram of UC-09**

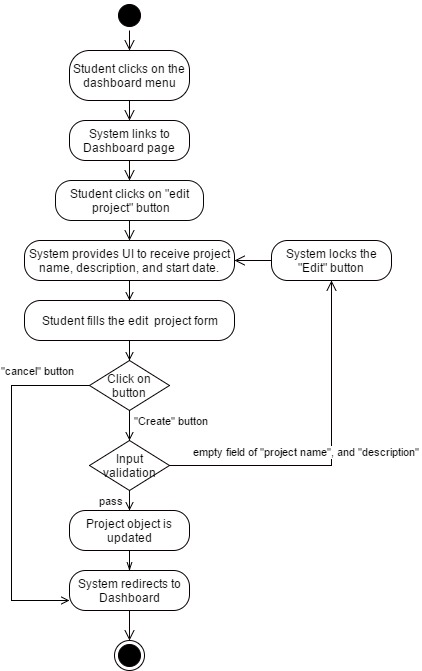


Figure 16 AD-11 Edit a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-10 | | |
| **Use Case Name:** | Student can delete a project. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can delete a project. | | |
| **Trigger:** | Student click on “delete” button. | | |
| **Preconditions:** | UC-03  System connect to database.  There is at least 1 project in the project list.  Student is in a dashboard. | | |
| **Post conditions:** | A project is removed from the database. | | |
| **Normal Flow:** | 1. System provides UI to display student’s dashboard. 2. Student clicks on “delete” project button. 3. System displays a confirmation message “Are you sure to delete this project?” with “Confirm” and “Cancel” button. 4. Student clicks “Confirm” button. 5. System deletes a project from in database. | | |
| **Alternative Flows:** | From 3.) If student click “Cancel” button:   1. System redirect to the dashboard. | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-10**

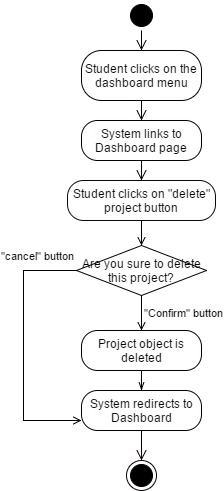


Figure 17 AD-12 Delete a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-11 | | |
| **Use Case Name:** | Student can create a task. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can create a task in a project. | | |
| **Trigger:** | Student click on “create new task” button. | | |
| **Preconditions:** | UC-03 System connects to database.  Student is in a Task page. | | |
| **Post conditions:** | System creates a new task object in database. | | |
| **Normal Flow:** | 1. System displays student’s dashboard. 2. Student click on a project name. 3. System displays the task page. 4. Student clicks on “Create new task” button. 5. System provides UI which receive tasks name, task description, start date, “create” button, and “cancel” button. 6. Student inputs task name, description, and start date data. 7. Student clicks on “create” button. 8. System validates empty field. 9. System creates a new task object in the database. 10. System redirects to Task page. | | |
| **Alternative Flows:** | From 5.) If student click on “cancel” button:   1. System redirects to Task page. | | |
| **Exceptions:** | From 7.) If there is an empty field:   1. System locks the “create” button. | | |

**Activity diagram of UC-11**

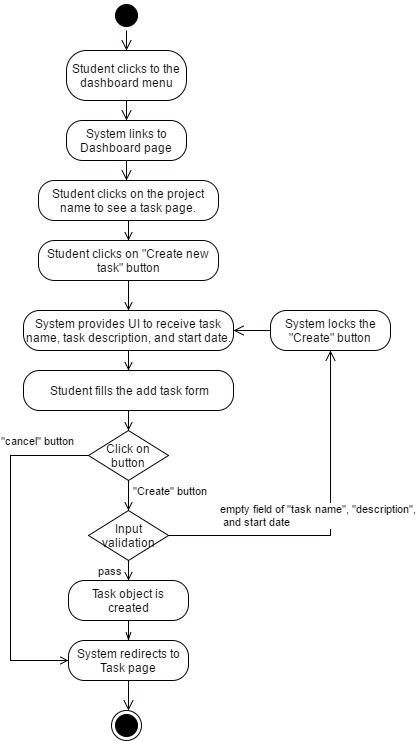


Figure 18 AD-13 Create a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-12 | | |
| **Use Case Name:** | Student can edit a task. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 21/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can edit a task in a project. | | |
| **Trigger:** | Student clicks the edit task button. | | |
| **Preconditions:** | UC-03  System connects to database.  Student is in a Task page.  There is a task in a project. | | |
| **Post conditions:** | The task object is updated. | | |
| **Normal Flow:** | 1. System displays student’s dashboard. 2. Student click on a project name. 3. System displays the task page. 4. Student clicks on “edit task” button. 5. System provides UI which receive tasks name, task description, “update” button, and “cancel” button. 6. Student inputs task name and description data. 7. Student clicks on “update” button. 8. System validates empty field. 9. System updates a task object in the database. 10. System redirects to Task page. | | |
| **Alternative Flows:** | From 5.) If student click on “cancel” button:   1. System redirects to Task page. | | |
| **Exceptions:** | From 7.) If task name field is empty:   1. System locks the “update” button. | | |

**Activity diagram of UC-12**

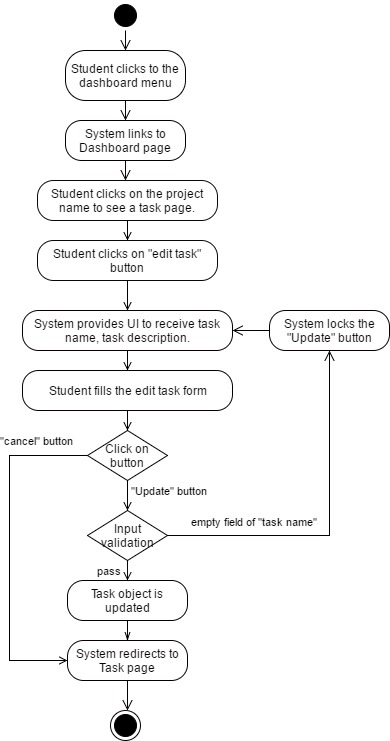


Figure 19 AD-14 Update a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-13 | | |
| **Use Case Name:** | Student can delete a task. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 21/03/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can delete a task in a project. | | |
| **Trigger:** | Student click “delete task” button. | | |
| **Preconditions:** | UC-03  System connects to database.  Student is in Task page.  There is a task in the selected project. | | |
| **Post conditions:** | Task object is deleted from the database. | | |
| **Normal Flow:** | 1. System displays student’s dashboard. 2. Student click on a project name. 3. System displays the task page. 4. Student clicks on “delete task” button. 5. System provides UI to display a confirmation message “Are you sure to delete this task?” with “Confirm” and “Cancel” button. 6. Student click “Confirm” button. 7. The task is deleted from the database. 8. System redirects to Task page. | | |
| **Alternative Flows:** | From 5.) If student click “Cancel” button:   1. System redirect to Task page. | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-13**

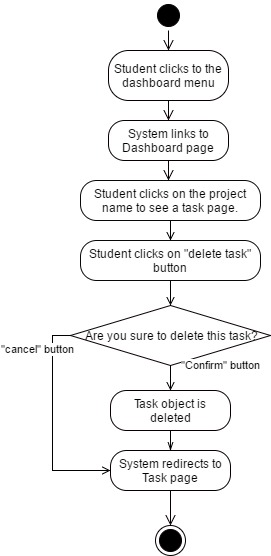


Figure 20 AD-15 Delete a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-14 | | |
| **Use Case Name:** | Student can change a status of a task. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can change status of a task. There are three status which are “To-Do”, “Doing”, and “Done” | | |
| **Trigger:** | Student drag a task card.  Student click the status button. | | |
| **Preconditions:** | UC-03  System connects to database.  Student is in the Task page.  There is a task in the selected project. | | |
| **Post conditions:** | An activity is created. | | |
| **Normal Flow:** | Flow 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. System provides UI to select the start date with “Confirm” button, and “Cancel” button. 4. Student selects start date. 5. Student click “Confirm” button. 6. System changes the task status.   Flow 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. System provides UI to select the start date with “confirm” button, and “Cancel” button. 4. Student selects start date. 5. Student click “Confirm” button. 6. System changes the task status.   Flow c.) Click status button in task detail page   1. Student clicks on a task to see a task detail page. 2. System provides UI to display task detail with the status button. 3. Student clicks on the status button. 4. System provides UI to select the start date with “Confirm” button, and “Cancel” button. 5. Student selects start date. 6. Student click “Confirm” button. 7. System changes the task status. | | |
| **Alternative Flows:** | From a-3 and b-3.) If student selects cancel button   1. Student clicks Cancel button. 2. System redirect to the task page.   From a-3 and b-3.) If student does not select any start date   1. System locks the “Confirm” button.   From b-1.) If student drags a task card and drop it to the same column   1. The task is still in the current column. | | |
| **Exceptions:** | From a.) If student drag a task card and does not drop in any column.   1. System display forbidden sign as a mouse cursor. 2. System locks for changing the status of a task. | | |

**Activity diagram of UC-14**

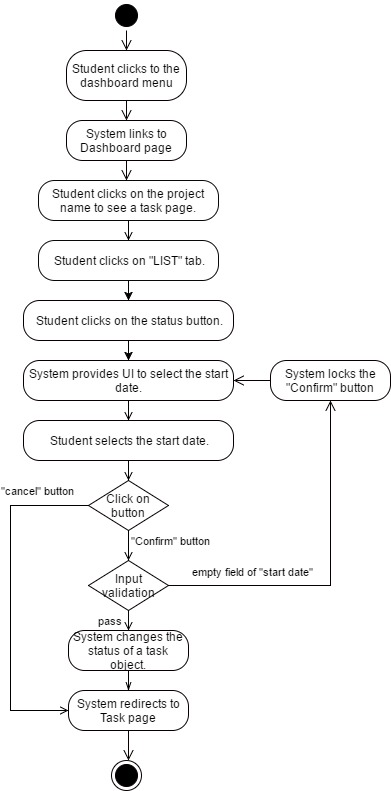


Figure 21 AD-16 Change a task status (List tab)

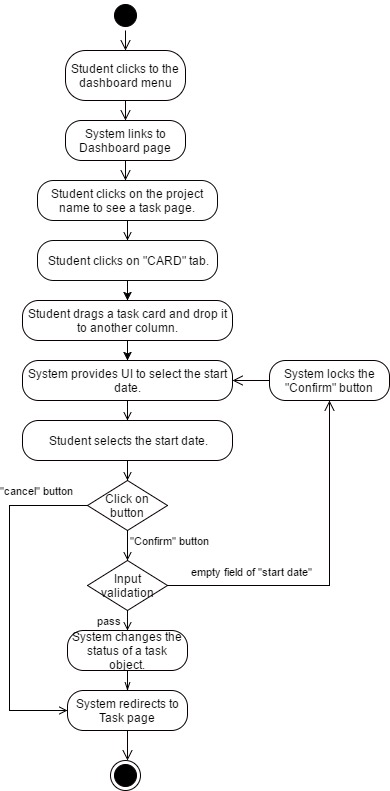


Figure 22 AD-17 Change a task status (Card tab)

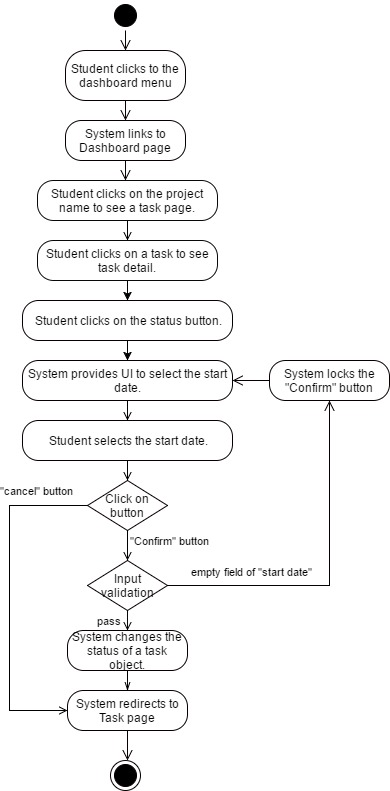


Figure 23 AD-18 Change a task status (in task detail page)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-15 | | |
| **Use Case Name:** | User can view a comment. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can view a comment in the task. | | |
| **Trigger:** | User click to see a task detail. | | |
| **Preconditions:** | UC-03  There is a task in the selected project.  User is in a Task detail page.  There is a comment in the selected task. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | 1. User clicks to see a task detail. 2. System requests the task data from database. 3. System provides UI to display a task detail, comment field, and status buttons. 4. User can see the comment(s). | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-15**

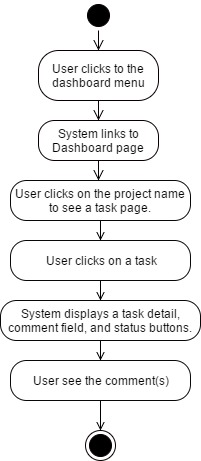


Figure 24 AD-18 View a comment

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-16 | | |
| **Use Case Name:** | User can create a comment. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student, Mentor, and Supervisor | | |
| **Description:** | Student, Mentor, and Supervisor can create a comment(s). | | |
| **Trigger:** | User click on “comment” button. | | |
| **Preconditions:** | UC-03  There is a task in the selected project.  User is in a task detail page. | | |
| **Post conditions:** | The task activity is updated.  Notification message is displays on a screen for 3 seconds. | | |
| **Normal Flow:** | 1. System provides a comment field with “comment” button. 2. User fills the text(s) in a comment form. 3. User clicks “comment” button. 4. System saves the comment data into the database. 5. System displays the comment. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-16**

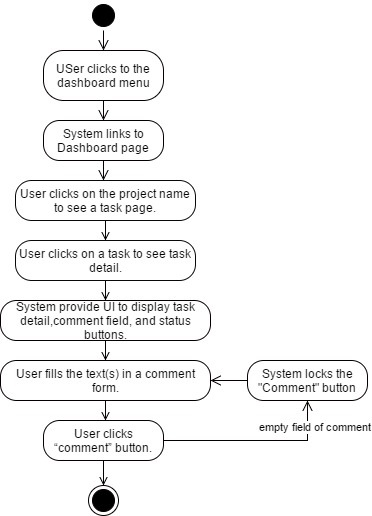


Figure 25 AD-19 Create a comment

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-17 | | |
| **Use Case Name:** | User can delete their own comment. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student, Mentor, and Supervisor | | |
| **Description:** | Student, Mentor, and Supervisor can delete their own comment. | | |
| **Trigger:** | User click on the cross button on the top right of the comment box. | | |
| **Preconditions:** | UC-03  User is in a task detail page.  There is a comment of the owner in the selected task. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | 1. System provides UI to delete their own comment. 2. User clicks “Delete” button. 3. System displays a confirmation message “Are you sure to delete this comment?”. 4. User click “Confirm” button. 5. System deletes the comment in the database. | | |
| **Alternative Flows:** | From 4.) If a user click on “cancel” button.   1. User clicks on Cancel button. 2. System cancels to delete a comment. 3. System redirects to the task detail page. | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-17**

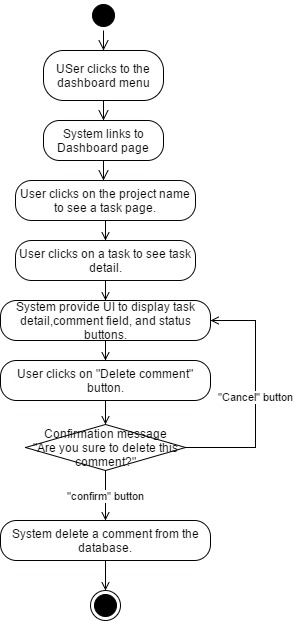


Figure 26 AD-20 Delete a comment

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-18 | | |
| **Use Case Name:** | User can view activities of projects. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student can view activities of projects. Mentor and Supervisor can view activities of student’s task in the selected task. | | |
| **Trigger:** | User click “Activities” tab. | | |
| **Preconditions:** | UC-03  Student is in a dashboard.  Mentor and Supervisor are in a student’s dashboard. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | 1. System provides UI to display task statistics, List tab, Card tab, and activities tab. 2. User click “Activities” tab. 3. System displays activities of tasks since the project was started. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-18**

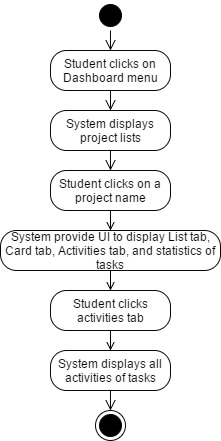


Figure 27 AD-21 View task activities (Student role)

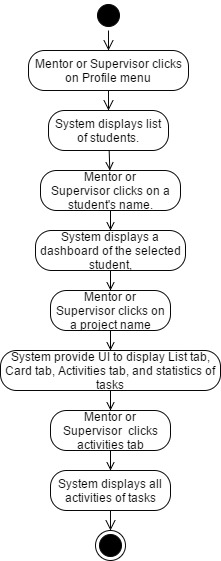


Figure 28 AD-22 View task activities (Mentor and Supervisor role)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-19 | | |
| **Use Case Name:** | Student can generate a weekly report. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can generate a weekly report. | | |
| **Trigger:** | Student clicks on “Report” menu. | | |
| **Preconditions:** | UC-03  There is a task in a system. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | 1. Student clicks on “Report” menu. 2. System requests tasks data from database. 3. System arranges task details in the report template. 4. System provide UI to display a weekly report. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-19**

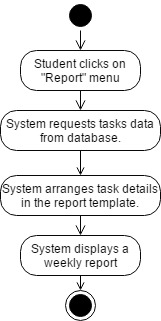


Figure 29 AD-23 Generate a weekly report

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-20 | | |
| **Use Case Name:** | Student can print a weekly report. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can print a weekly report. | | |
| **Trigger:** | User clicks on “print” button in report page. | | |
| **Preconditions:** | UC-03, UC-19 | | |
| **Post conditions:** | A weekly report file is sent to a printer. | | |
| **Normal Flow:** | 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. | | |
| **Alternative Flows:** | From 2.) If student clicks on “cancel” button.   1. System redirects to weekly report UI. | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-20**

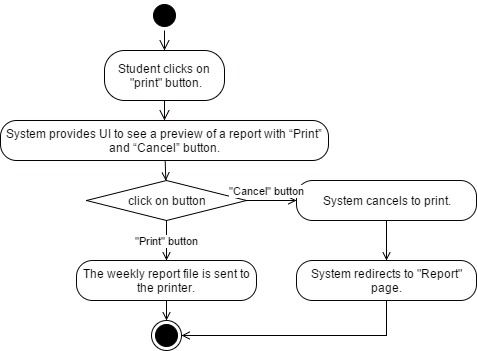


Figure 30 AD-24 Print a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-21 | | |
| **Use Case Name:** | User can receive an activity notification via Web application. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 23/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can receive notification via Web application | | |
| **Trigger:** | 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) | | |
| **Preconditions:** | UC-03 | | |
| **Post conditions:** | Notify object is created. | | |
| **Normal Flow:** | Student role:   1. System detects a new activity. 2. System pushes an activity to database. 3. System receives notify object from database. 4. System displays an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab. 5. User receives an activity notification.   Mentor or supervisor role:   1. System detects a new activity of student who under their guidance. 2. System pushes an activity to database. 3. System receives notify object from database. 4. System displays an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab. 5. User receives an activity notification. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-21**

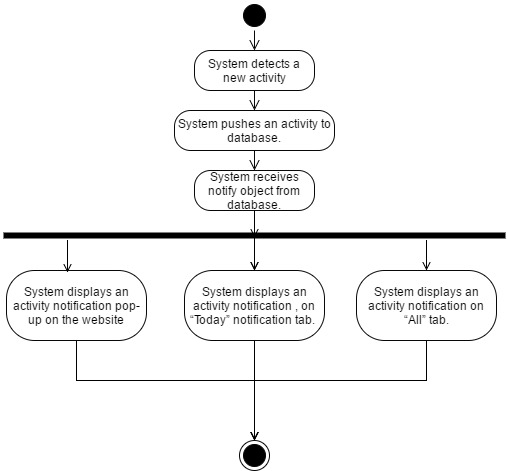


Figure 31 AD-25 Web notification

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-22 | | |
| **Use Case Name:** | User can receive an activity notification via Email. | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 23/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can receive notification via Email. | | |
| **Trigger:** | 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) | | |
| **Preconditions:** | UC-03  System have emails of user. | | |
| **Post conditions:** | - | | |
| **Normal Flow:** | Student role:   1. System detects a new activity. 2. System send request to EmailSender. 3. User receives an email of activity notification.   Mentor or supervisor role:   1. System detects a new activity of student who under their guidance. 2. System sends a request to EmailSender. 3. User receives an email of activity notification. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |

**Activity diagram of UC-22**

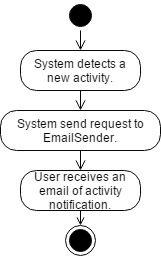


Figure 32 AD-26 Mail notification