**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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| 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, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- SRS\_V.0.9.docx | - Edit Chapter III | Specific Requirement  - Use Case - Edit Chapter IV | Software Requirement Specification  - Edit UC | | Review | 29-04-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |
| WRMS- SRS\_V.1.0.docx | Release Software Requirement Specification document | | Release | 05-05-2017 | PS,VI, PSU | PS,VI, PSU | PS,VI, PSU |

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

Table of Contents

[1.1 Purpose 7](#_Toc482046632)

[1.2 Project Scope 7](#_Toc482046633)

[1.3 Acronyms and definitions Acronyms 7](#_Toc482046634)

[1.4 Definitions 7](#_Toc482046635)

[Chapter II | Overall Description 9](#_Toc482046636)

[2.1 Product perspective 9](#_Toc482046637)

[2.2 Product Functions 9](#_Toc482046638)

[2.3 User Characteristic 9](#_Toc482046639)

[2.4 Operation Environment 9](#_Toc482046640)

[2.5 Design and Implementation constants 9](#_Toc482046641)

[Chapter III | Software Requirement Specification 9](#_Toc482046642)

[3.1 User requirements 9](#_Toc482046643)

[3.2 System requirements 10](#_Toc482046644)

[3.3 User requirement specification with system requirement specification 13](#_Toc482046645)

[Chapter IV | Requirement Specification 20](#_Toc482046646)

[4.1 Use case diagram 20](#_Toc482046647)

[4.2 Use case description 24](#_Toc482046648)

[UC-01 Register 24](#_Toc482046649)

[UC-02 Edit profile 27](#_Toc482046650)

[UC-03 Login 30](#_Toc482046651)

[UC-04 Logout 32](#_Toc482046652)

[UC-05 Add registration code 33](#_Toc482046653)

[UC-06 View a task 35](#_Toc482046654)

[UC-07 View the statistics of tasks 37](#_Toc482046655)

[UC-08 Create a project 39](#_Toc482046656)

[UC-09 Edit a project 40](#_Toc482046657)

[UC-10 Delete a project 43](#_Toc482046658)

[UC-11 Create a task 44](#_Toc482046659)

[UC-12 Edit a task 46](#_Toc482046660)

[UC-13 Delete a task 48](#_Toc482046661)

[UC-14 Change a status of a task 49](#_Toc482046662)

[UC-15 View a comment 54](#_Toc482046663)

[UC-16 Create a comment 55](#_Toc482046664)

[UC-17 Delete a comment 57](#_Toc482046665)

[UC-18 View activities in a project 59](#_Toc482046666)

[UC-19 View activities of the under guidance students 60](#_Toc482046667)

[UC-20 Generate a weekly report 61](#_Toc482046668)

[UC-21 Print a weekly report 62](#_Toc482046669)

[UC-22 Receive a notification via Web application 63](#_Toc482046670)

[UC-23 Receive an activity notification via Email 65](#_Toc482046671)

**Table of Figures**

[Figure 1 Use case diagram of User management 20](#_Toc482046672)

[Figure 2 Use case diagram of Project management 20](#_Toc482046673)

[Figure 3 Use case diagram of Progress tracking 21](#_Toc482046674)

[Figure 4 Use case diagram of Report export 21](#_Toc482046675)

[Figure 5 Use case diagram of Notification 22](#_Toc482046676)

[Figure 6 Use case diagram of Overall usecase 23](#_Toc482046677)

[Figure 7 AD-01 Registration 26](#_Toc482046678)

[Figure 8 AD-02 Edit profile 29](#_Toc482046679)

[Figure 9 AD-03 Login 31](#_Toc482046680)

[Figure 10 AD-04 Logout 32](#_Toc482046681)

[Figure 11 AD-05 Add Mentor or Supervisor code 34](#_Toc482046682)

[Figure 12 AD-06 View a task 36](#_Toc482046683)

[Figure 14 AD-07 View statistics of tasks (Role as a student) 37](#_Toc482046684)

[Figure 15 AD-08 View statistics of tasks (Role as a Mentor or Supervisor) 38](#_Toc482046685)

[Figure 16 AD-09 Create a project 40](#_Toc482046686)

[Figure 17 AD-10 Edit a project 42](#_Toc482046687)

[Figure 18 AD-11 Delete a project 43](#_Toc482046688)

[Figure 19 AD-12 Create a task 45](#_Toc482046689)

[Figure 20 AD-13 Update a task 47](#_Toc482046690)

[Figure 21 AD-14 Delete a task 48](#_Toc482046691)

[Figure 22 AD-15 Change a task status (List tab) 51](#_Toc482046692)

[Figure 23 AD-16 Change a task status (Card tab) 52](#_Toc482046693)

[Figure 24 AD-17 Change a task status (in task detail page) 53](#_Toc482046694)

[Figure 25 AD-18 View a comment 54](#_Toc482046695)

[Figure 26 AD-19 Create a comment 56](#_Toc482046696)

[Figure 27 AD-20 Delete a comment 58](#_Toc482046697)

[Figure 28 AD-21 View activitiesin a project 59](#_Toc482046698)

[Figure 29 AD-22 View activities of the under guidance students 60](#_Toc482046699)

[Figure 30 AD-23 Generate a weekly report 61](#_Toc482046700)

[Figure 31 AD-24 Print a report 62](#_Toc482046701)

[Figure 32 AD-25 Web notification 64](#_Toc482046702)

[Figure 33 AD-26 Mail notification 65](#_Toc482046703)

**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 to the system by providing the first name, last name, description, company, position, email, password, profile image, signature image, start date, and role of user.
2. Student, Mentor, and Supervisor can edit their own profile.
3. Student, Mentor, and Supervisor can login by providing email and password.
4. Student, Mentor, and Supervisor can logout from the system.
5. Student can add registration code.

**Feature #2: Project management**

1. Student, Mentor, and Supervisor can view tasks of a project.
2. Student, Mentor, and Supervisor can view the statistics of tasks.
3. Student can create a project by providing the name of project, description, and start date.
4. Student can edit a project.
5. Student can delete a project.
6. Student can create a task by providing the name of task, description, and start date.
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. Student, Mentor, and Supervisor can view comments.
2. Student, Mentor, and Supervisor can create a comment by providing a comment message.
3. Student, Mentor, and Supervisor can delete their own comment.
4. Student, Mentor, and Supervisor can view activities of tasks in a specific project.
5. Mentor and Supervisor can view activities of all under guidance students.

**Feature #4: Report export**

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

**Feature #5: Notification**

1. Student, Mentor, and Supervisor can receive an activity notification via Web application.
2. Student, Mentor, and Supervisor can receive an activity notification via Email.

## **System requirements**

1. The 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. The system shall validate first name, last name, description, company, position, email, password, start date, and role of user.
3. The system shall display an error message “The email has already been taken”.
4. The system shall display an error message “The email must be a valid email address”.
5. The system shall display an error message “All fields are required”.
6. The system shall redirect to dashboard.
7. The system shall connect to database.
8. The system saves input data into the database.
9. The system shall redirect to Login page.
10. The 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.
11. The system shall validate first name, last name, description, company, position, email, password, and start date.
12. The system shall provide UI to display an error message “Invalid Email”.
13. The system shall redirect to profile page.
14. The system shall provide UI to receive Email and password.
15. The system shall validate email and password.
16. The system displays an error message “Invalid credential”.
17. The system displays an error message “The password must be at least 6 characters”.
18. The system shall navigate the student to access the student’s dashboard.
19. The system shall navigate the mentor to access the mentor’s dashboard.
20. The system shall navigate the supervisor to access the supervisor’s dashboard.
21. The system shall provide a button to logout.
22. The system shall be logged out.
23. The system shall provide UI to receive a code
24. The system shall validate a code.
25. The system shall display a confirmation message “Albert Einstein (Supervisor)?” with “Confirm” button and “Cancel” button.
26. The system shall link the mentor/supervisor account to the student account.
27. The system shall redirect to profile page.
28. The system shall update a name of mentor or supervisor on student’s profile page.
29. The system shall lock the button.
30. The system provide UI to display list of projects, a button of creating new project, task statistics, and summary of activities frequency.
31. The system shall provide UI to display the task page which contains task statistics, list of tasks, card of tasks tab, and tasks activity tab.
32. The system shall display a card of task on card board.
33. The system shall display an activity of a task.
34. The system shall provide UI to display a dashboard which contains statistics box, activities box, and a list of projects.
35. The system shall redirect to the dashboard of a selected student
36. The system shall provide UI to display a dashboard which contains statistics box, activities box, and a list of projects.
37. The system shall provide “create new project” button in the dashboard.
38. The system shall provide UI to receive project name, project description, start date, “create” button, and “cancel” button.
39. The system shall validate project name, description, and start date.
40. The system redirects to the dashboard.
41. The system shall display an error message “The project name field is required.”.
42. The system displays an error message “The description field is required.”.
43. The system shall provide UI to display student’s dashboard.
44. The system shall provide UI to receive project name, description, start date, “edit” button, and “Cancel” button.
45. The system shall lock the “Edit” button.
46. The system shall provide UI to display a confirmation message “Are you sure to delete this project?” with “Confirm” and “Cancel” button.
47. The system shall delete a project in database.
48. The system shall provide UI to receive tasks name, task description, start date, “create” button, and “cancel” button.
49. The system shall validate task name, description, and start date.
50. The system shall create a new task object in the database.
51. The system shall redirect to Task page.
52. The system shall lock the “create” button.
53. The system shall display an error message “The task name field is required.”
54. The system shall display an error message “The task description field is required.”
55. The system displays an error message “The start date field is required.”
56. The system shall provide UI which receive tasks name, task description, “update” button, and “cancel” button.
57. The system shall provide UI which receive tasks name, task description, “update” button, and “cancel” button.
58. The system shall validate tasks name and task description.
59. The system shall update a task object in the database
60. The system shall provide UI to display a confirmation message “Are you sure to delete this task?” with “Confirm” and “Cancel” button.
61. The system shall delete the selected task from the database.
62. The system shall provide UI to select the start date with “Confirm” button, and “Cancel” button.
63. The system shall validate the start date.
64. The system shall change the task status.
65. The system shall displays a task in card style.
66. The system shall provide UI to display task detail with the status button
67. The system shall lock the “Confirm” button.
68. The system display forbidden sign as a mouse cursor.
69. The system shall lock for changing the status of a task.
70. The system shall remain a card task in the same column.
71. The system shall discard to change the status of task.
72. The system shall request the task from database.
73. The system shall provide UI to display a task detail, comment field, and status buttons.
74. The system shall provide UI to display task detail, status buttons, and comment field with “comment” button
75. The system shall validate a comment field.
76. The system shall save the comment data into the database.
77. The system shall display a comment.
78. The system shall lock a comment button.
79. The system shall provide a bin icon to delete their own comment.
80. The system shall display a confirmation message “Are you sure to delete this comment?” with “Confirm” button and “Cancel” button.
81. The system shall delete the comment in the database.
82. The system shall display a message “Delete comment success”.
83. The system shall redirect to the task detail page.
84. The system shall discard to delete a comment.
85. The system shall provide UI to display task statistics, List tab, Card tab, and activities tab.
86. The system displays all activities of a selected project.
87. The system provides UI to display dashboard with task all task statistics, and all activities of students.
88. The system shall arrange task details in the report template.
89. The system shall provide UI to display a weekly report which is separated by the project.
90. The system shall provide UI to see a preview of a report with “Print” and “Cancel” button.
91. The system sends a weekly report to a printer
92. The system redirects to the report page.
93. The system shall discard to print a report.
94. The system shall detect a new activity.
95. The system shall send a request to Firebase service.
96. The system shall observe data in Firebase.
97. The system shall display an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab.
98. The system shall detect a new activity.
99. The system shall send a request to EmailSender.

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

1. **Visitor can register to the system by providing the first name, last name, description, company, position, email, password, profile image, signature image, start date, and role of user.**
2. The 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. The system shall validate first name, last name, description, company, position, email, password, start date, and role of user.
4. The system shall display an error message “The email has already been taken”.
5. The system shall display an error message “The email must be a valid email address”.
6. The system shall display an error message “All fields are required”.
7. The system shall redirect to dashboard.
8. The system shall connect to database.
9. The system saves input data into the database.
10. The system shall redirect to Login page.

**URS-02: Student, Mentor, and Supervisor can edit their own profile.**

SRS-05: The system provides UI to display an error message “All fields are required”.

SRS-08: The system shall save data into the database.

1. The 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. The system shall validate first name, last name, description, company, position, email, password, and start date.
3. The system shall provide UI to display an error message “Invalid Email”.
4. The system shall redirect to profile page.

**URS-03: Student, Mentor, and Supervisor can login by providing email and password.**

SRS-09: The system shall redirect to Login page.

1. The system shall provide UI to receive Email and password.
2. The system shall validate email and password.
3. The system displays an error message “Invalid credential”.
4. The system displays an error message “The password must be at least 6 characters”.
5. The system shall navigate the student to access the student’s dashboard.
6. The system shall navigate the mentor to access the mentor’s dashboard.
7. The system shall navigate the supervisor to access the supervisor’s dashboard.

**URS-04: Student, Mentor, and Supervisor can logout from the system.**

SRS-09: The system shall redirect to Login page.

1. The system shall provide a button to logout.
2. The system shall be logged out.

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

1. The system shall provide UI to receive a code
2. The system shall validate a code.
3. The system shall display a confirmation message “Albert Einstein (Supervisor)?” with “Confirm” button and “Cancel” button.
4. The system shall link the mentor/supervisor account to the student account.
5. The system shall redirect to profile page.
6. The system shall update a name of mentor or supervisor on student’s profile page.
7. The system shall lock the button.

**URS-06: Student, Mentor, and Supervisor can view tasks of a project.**

1. The system provide UI to display list of projects, a button of creating new project, task statistics, and summary of activities frequency.
2. The system shall provide UI to display the task page which contains task statistics, list of tasks, card of tasks tab, and tasks activity tab.
3. The system shall display a card of task on card board.
4. The system shall display an activity of a task.

**URS-07: Student, Mentor, and Supervisor can view the statistics of tasks.**

1. The system shall provide UI to display a dashboard which contains statistics box, activities box, and a list of projects.
2. The system shall redirect to the dashboard of a selected student
3. The system shall provide UI to display a dashboard which contains statistics box, activities box, and a list of projects.

**URS-08: Student can create a project by providing the name of project, description, and start date.**

1. The system shall provide “create new project” button in the dashboard.
2. The system shall provide UI to receive project name, project description, start date, “create” button, and “cancel” button.
3. The system shall validate project name, description, and start date.
4. The system redirects to the dashboard.
5. The system shall display an error message “The project name field is required.”.
6. The system displays an error message “The description field is required.”.

**URS-9: Student can edit a project.**

SRS-08: The system shall save data into the database.

SRS-39: The system shall validate project name, description, and start date.

SRS-40: The system redirects to the dashboard.

SRS-41: The system shall display an error message “The project name field is required.”.

SRS-42: The system displays an error message “The description field is required.”.

1. The system shall provide UI to display student’s dashboard.
2. The system shall provide UI to receive project name, description, start date, “edit” button, and “Cancel” button.
3. The system shall lock the “Edit” button.

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

SRS-40: The system redirects to the dashboard.

SRS-43: The system shall provide UI to display student’s dashboard.

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

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

1. The system shall provide UI to receive tasks name, task description, start date, “create” button, and “cancel” button.
2. The system shall validate task name, description, and start date.
3. The system shall create a new task object in the database.
4. The system shall redirect to Task page.
5. The system shall lock the “create” button.
6. The system shall display an error message “The task name field is required.”
7. The system shall display an error message “The task description field is required.”
8. The system displays an error message “The start date field is required.”

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

SRS-51: The system shall redirect to Task page.

SRS-53: The system shall display an error message “The task name field is required.”

SRS-54: The system shall display an error message “The task description field is required.”

1. The system shall provide UI which receive tasks name, task description, “update” button, and “cancel” button.
2. The system shall provide UI which receive tasks name, task description, “update” button, and “cancel” button.
3. The system shall validate tasks name and task description.
4. The system shall update a task object in the database

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

SRS-51: The system shall redirect to Task page.

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

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

SRS-51: The system shall redirect to Task page.

1. The system shall provide UI to select the start date with “Confirm” button, and “Cancel” button.
2. The system shall validate the start date.
3. The system shall change the task status.
4. The system shall displays a task in card style.
5. The system shall provide UI to display task detail with the status button
6. The system shall lock the “Confirm” button.
7. The system display forbidden sign as a mouse cursor.
8. The system shall lock for changing the status of a task.
9. The system shall remain a card task in the same column.
10. The system shall discard to change the status of task.

.

**URS-15: Student, Mentor, and Supervisor can view comments.**

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

**URS-16: Student, Mentor, and Supervisor can create a comment.**

1. The system shall provide UI to display task detail, status buttons, and comment field with “comment” button
2. The system shall validate a comment field.
3. The system shall save the comment data into the database.
4. The system shall display a comment.
5. The system shall lock a comment button.

**URS-17: Student, Mentor, and Supervisor can delete their own comment.**

1. The system shall provide a bin icon to delete their own comment.
2. The system shall display a confirmation message “Are you sure to delete this comment?” with “Confirm” button and “Cancel” button.
3. The system shall delete the comment in the database.
4. The system shall display a message “Delete comment success”.
5. The system shall redirect to the task detail page.
6. The system shall discard to delete a comment.

**URS-18: Student, Mentor, and Supervisor can view activities of tasks in a specific project.**

1. The system shall provide UI to display task statistics, List tab, Card tab, and activities tab.
2. The system displays all activities of a selected project.

**URS-19: Mentor and Supervisor can view activities of all under guidance students.**

1. The system provides UI to display dashboard with task all task statistics, and all activities of students.

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

SRS-72: The system shall request the task from database.

1. The system shall arrange task details in the report template.
2. The system shall provide UI to display a weekly report which is separated by the project.

**URS-21: Student can print a weekly report.**

1. The system shall provide UI to see a preview of a report with “Print” and “Cancel” button.
2. The system sends a weekly report to a printer
3. The system redirects to the report page.
4. The system shall discard to print a report.

**URS-22: Student, Mentor, and Supervisor can receive an activity notification via Web application.**

1. The system shall detect a new activity.
2. The system shall send a request to Firebase service.
3. The system shall observe data in Firebase.
4. The system shall display an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab.

**URS-23: Student, Mentor, and Supervisor can receive an activity notification via Email.**

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

# **Chapter IV | Requirement Specification**

## **Use case diagram**

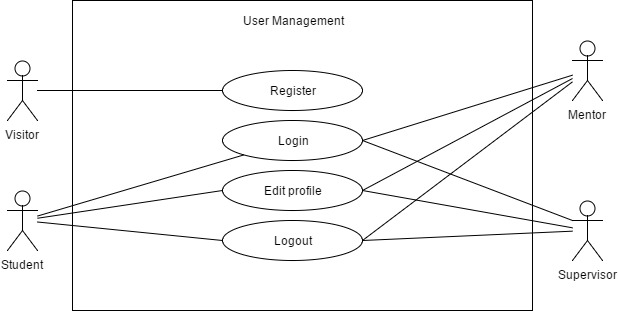


Figure 1 Use case diagram of User management

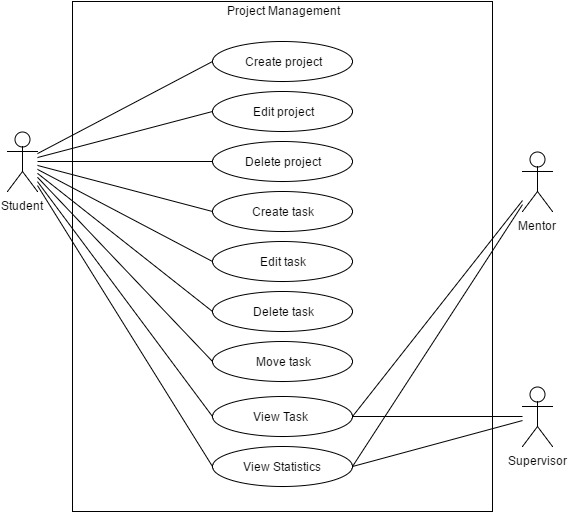


Figure 2 Use case diagram of Project management

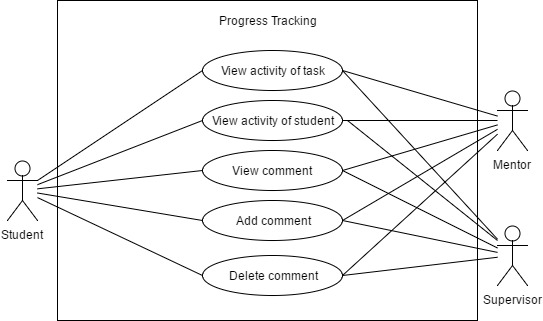


Figure 3 Use case diagram of Progress tracking

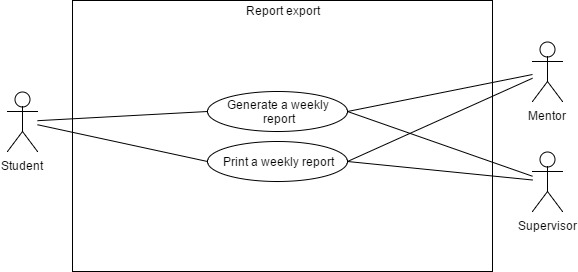


Figure 4 Use case diagram of Report export

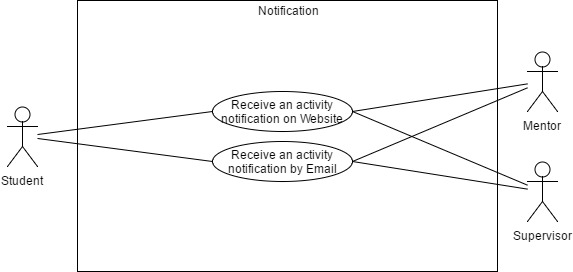


Figure 5 Use case diagram of Notification

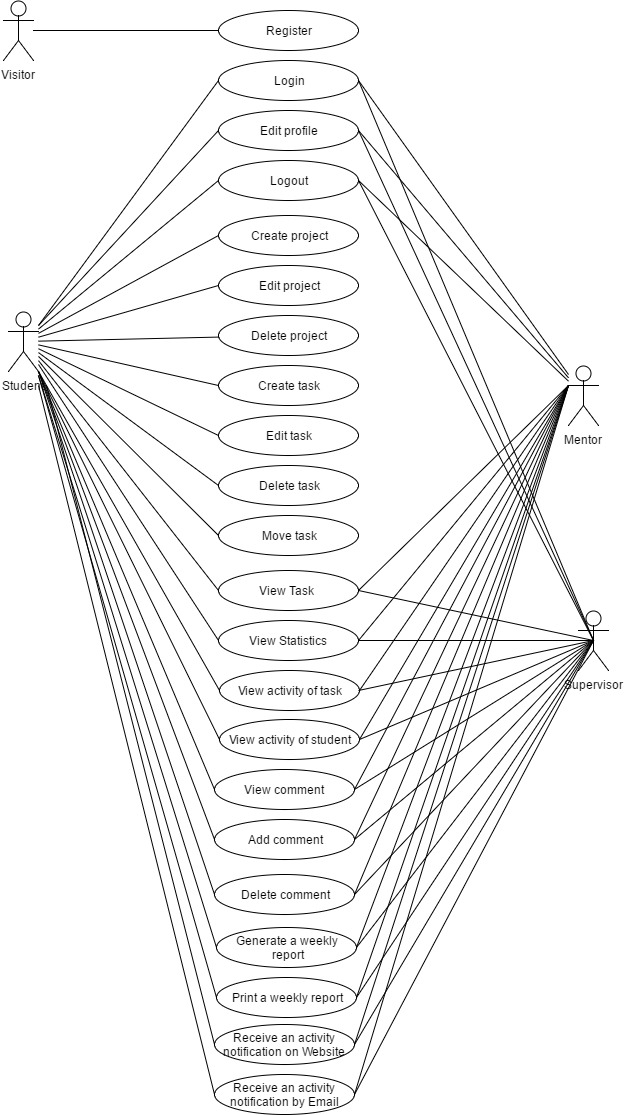


Figure 6 Use case diagram of Overall usecase

## **Use case description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-01 Register | | |
| **Use Case Name:** | Register | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 3/05/2017 |
| **Actors:** | Visitor | | |
| **Description:** | Visitor can register to the WIL report management system. | | |
| **Trigger:** | Visitor enter to WIL report management web application. | | |
| **Preconditions:** | - | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| First name | String | This field is not empty. | Isaac |
| Last name | String | This field is not empty. | Newton |
| Description | String | This field is not empty. | I am a student. |
| Company | String | This field is not empty. | XYZ Company |
| Position | String | This field is not empty. | Student |
| E-mail | String | - This field is not empty.  - The email is in a correct format. | Newton@xmail.com |
| Password | String | This field is not empty. | Newton1 |
| Profile image | String | - This field is not empty.  - The file is uploaded from the browser.  - The file extension is jpg or png. | Newton.jpg |
| Signature image | String | - This field is not empty.  - The file is uploaded from the browser.  - The file extension is jpg or png. | Sign\_newton.jpg |
| Start date | Date time | - This field will be displayed if the role of user is “Student”.  - This field is not empty.  - The date is selected on the providing calendar. | 04/03/2017 |
| Role of user | String | Visitor can select only one role. | Student |
| **Post condition** | * If the Register use case is successful, the information is save in database. * If the Register use case is unsuccessful, the system state remains the same and the system display the error message. | | |
| **Normal Flow:** | 1. The system provides 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. 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. The system validates first name, last name, description, company, position, email, password, start date, and student role. 5. The system saves data into the database. 6. The system redirect to dashboard. | | |
| **Alternative Flows:** | From 4.) If visitor registers with an account with existing email   1. The system displays an error message “The email has already been taken”. 2. The system returns to the step 1 of the normal flow.   From 4.) If visitor inputs incorrect format of email   1. The system displays an error message “The email must be a valid email address”. 2. The system returns to the step 1 of the normal flow.   From 4) If user inputs empty fields:   1. The system displays an error message “All fields are required”. 2. The system returns to the step 1 of the normal flow. | | |
| **Exceptions:** | From 3.) If visitor click cancel button:   1. The system redirects to Login page. | | |
| **Assumption:** | The user understand English. | | |

**Activity diagram of UC-01**

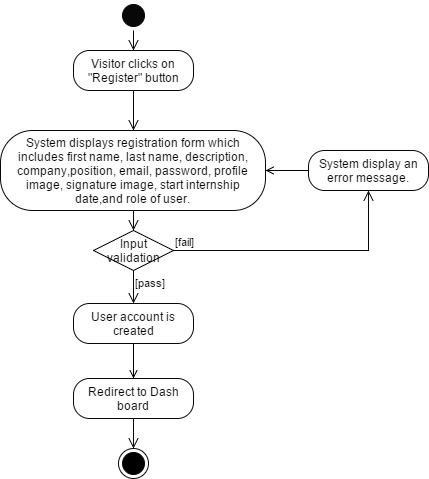


Figure 7 AD-01 Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-02 Edit profile | | |
| **Use Case Name:** | Edit profile | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can edit their own profile. | | |
| **Trigger:** | Student, Mentor, Supervisor clicks on edit profile button. | | |
| **Preconditions:** | Student, Mentor, Supervisor already logged in into the system. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| First name | String | This field is not empty. | Isaac |
| Last name | String | This field is not empty. | Newton |
| Description | String | This field is not empty. | I am a student. |
| Company | String | This field is not empty. | XYZ Company |
| Position | String | This field is not empty. | Student |
| E-mail | String | - This field is not empty.  - The email is in a correct format. | Newton@xmail.com |
| Password | String | This field is not empty. | Newton1 |
| Profile image | String | - This field is not empty.  - The file is uploaded from the browser.  - The file extension is jpg or png. | Newton.jpg |
| Signature image | String | - This field is not empty.  - The file is uploaded from the browser.  - The file extension is jpg or png. | Sign\_newton.jpg |
| Start date | Date time | - This field will be displayed if the role of user is “Student”.  - This field is not empty.  - The date is selected on the providing calendar. | 04/03/2017 |
| Role of user | String | Visitor can select only one role. | Student |
| **Post conditions:** | * If the Edit profile use case is successful, the new information is save into the database. * If the Edit profile use case is unsuccessful, the system state remains the same and the system display the error message. | | |
| **Normal Flow:** | 1. The system provides UI to receive first name, last name, description, company, position, email, password, profile image, signature image, and start date. 2. The user inputs first name, last name, description, company, position, email, password, profile image, signature image, and start date. 3. The user clicks “update” button. 4. The system validates first name, last name, description, company, position, email, password, profile image, signature image, and start date. 5. The system saves data into the database. 6. The system redirects to profile page. | | |
| **Alternative Flows:** | From 4.) If the user inputs incorrect format of email:   1. The system provides UI to display an error message “Invalid Email”. 2. The system returns to the step 1 of the normal flow.   From 4.) If user inputs empty forms:   1. The system provides UI to display an error message “All fields are required”. 2. The system returns to the step 1 of the normal flow. | | |
| **Exceptions:** | From 3.) If the user clicks cancel button:   1. The system redirects to profile page. | | |
| **Assumption:** | The user understands English. | | |

**Activity diagram of UC-02**

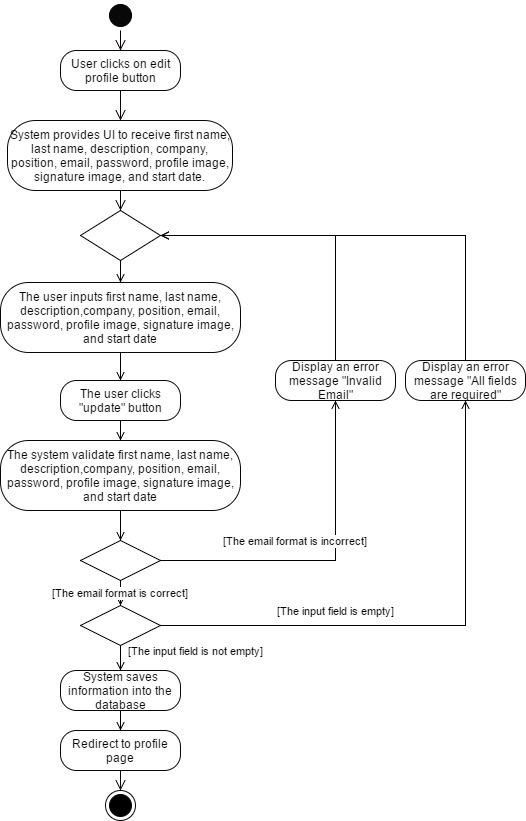


Figure 8 AD-02 Edit profile

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-03 Login | | |
| **Use Case Name:** | Login | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can login into the WIL report management system. | | |
| **Trigger:** | Student, Mentor, Supervisor clicks on login button. | | |
| **Preconditions:** | Student, Mentor, Supervisor already make a registration in WRMS. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Email | String | - This field is not empty.  - The email is in a correct format. | Newton@xmail.com |
| Password | String | This field is not empty. | Newton1 |
| **Post conditions:** | * If the Login use case is successful, the system redirects to the dashboard. * If the Login use case is unsuccessful, the system will display an error message. * The session of a user account is started. | | |
| **Normal Flow:** | 1. The user clicks “Register” button. 2. The system provides UI to receive Email and password. 3. The user inputs email and password. 4. The user clicks “Login” button. 5. The system validates email and password. 6. The system redirects to a dashboard of user. | | |
| **Alternative Flows:** | From 6.)  a.) Student role:   1. The system navigates the student to access the student’s dashboard.   b.) Mentor role:   1. The system navigates the mentor to access the mentor’s dashboard.   c.) Supervisor role:   1. The system navigates the supervisor to access the supervisor’s dashboard   From 5.) If the email or password is incorrect:   1. The system displays an error message “Invalid credential”. 2. The system redirects to Login page.   From 5.) If password is shorter than 6 characters:   1. The system displays an error message “The password must be at least 6 characters”. 2. The system redirects to Login page. | | |
| **Exceptions:** | - | | |
| **Assumption:** | The user understands English. | | |

**Activity diagram of UC-03**

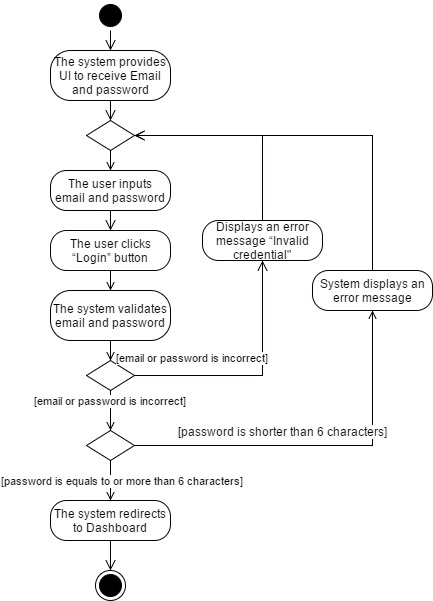


Figure 9 AD-03 Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-04 Logout | | |
| **Use Case Name:** | Logout | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can logout from the WIL report management system. | | |
| **Trigger:** | Student, Mentor, Supervisor clicks on a logout button. | | |
| **Preconditions:** | Student, Mentor, Supervisor already logged in into the system. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | **-** | **-** | **-** |
| **Post conditions:** | The user’s session is expired. | | |
| **Normal Flow:** | 1. The system provides a button to logout. 2. The user clicks logout button. 3. The system is logged out. 4. The system redirects to login page. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumption:** | The user understands English. | | |

**Activity diagram of UC-04**

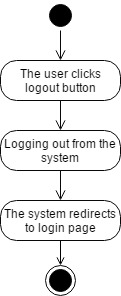


Figure 10 AD-04 Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-05 Add registration code | | |
| **Use Case Name:** | Add registration code | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can add mentor and supervisor code in profile page. | | |
| **Trigger:** | - | | |
| **Preconditions:** | * Student already logged in into the system. * Student has to know the code of Mentor or Supervisor. * Student accesses to profile page. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Mentor/Supervisor Code | String | * This field is not empty. * The code contains 5 characters. | 8f14e |
| **Post conditions:** | The system connects account of student to mentor and supervisor account. | | |
| **Normal Flow:** | 1. The system provides UI to receive a code. 2. The user inputs mentor or supervisor code. 3. The user click “+” button. 4. The system validates a code. 5. The system displays a confirmation message “Albert Einstein (Supervisor)?” with “Confirm” button and “Cancel” button. 6. The user clicks “Confirm”. 7. The system links the mentor/supervisor account to the student account. 8. The system redirects to profile page. 9. The system updates a name of mentor or supervisor on student’s profile page. | | |
| **Alternative Flows:** | From 4) If the code is shorter or longer than 5 characters:   1. The system locks the button. 2. The system returns to the step 1 of the normal flow.   From 4) If the code field is empty:   1. The system locks the button. 2. The system returns to the step 1 of the normal flow. | | |
| **Exceptions:** | From 6.) If student clicks cancel button:   1. The system skips to the step 8 of the normal flow. | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-05**

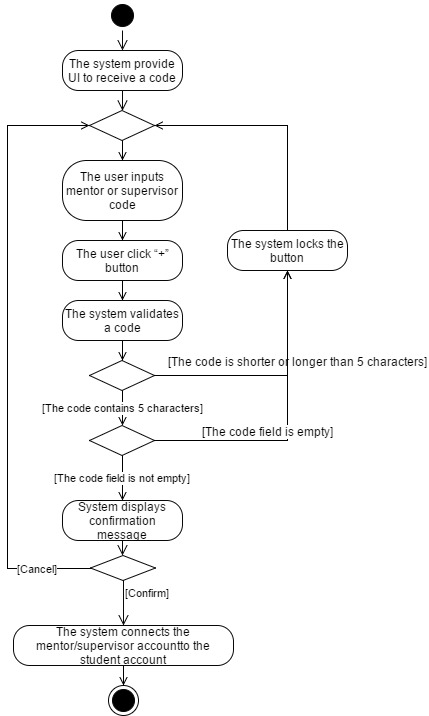


Figure 11 AD-05 Add Mentor or Supervisor code

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-06 View a task | | |
| **Use Case Name:** | View a task | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 27/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can view a task of each project. | | |
| **Trigger:** | Student, Mentor, Supervisor clicks on a project name. | | |
| **Preconditions:** | * Student, Mentor, Supervisor already logged in into the system. * The student is in the dashboard. * Mentor and Supervisor are in the student’s dashboard. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | The system displays the task statistics, list of tasks, card of tasks tab, and tasks activity tab. | | |
| **Normal Flow:** | 1. The system provides UI to display list of projects, a button of creating new project, task statistics, and summary of activities frequency. 2. The user clicks on a project name to see a task page. 3. The system displays the task page which contains task statistics, list of tasks, card of tasks tab, and tasks activity tab. | | |
| **Alternative Flows:** | From 3.)If the user wants to see a card of tasks:   1. The user clicks on “Card” tab. 2. The system displays a card of task on card board.   From 3.)If the user wants to see activities of tasks:   1. The user clicks on “Activities” tab. 2. The system displays an activity of a task. | | |
| **Exceptions:** | - | | |
| **Assumption:** | The user understand English. | | |

**Activity diagram of UC-06**

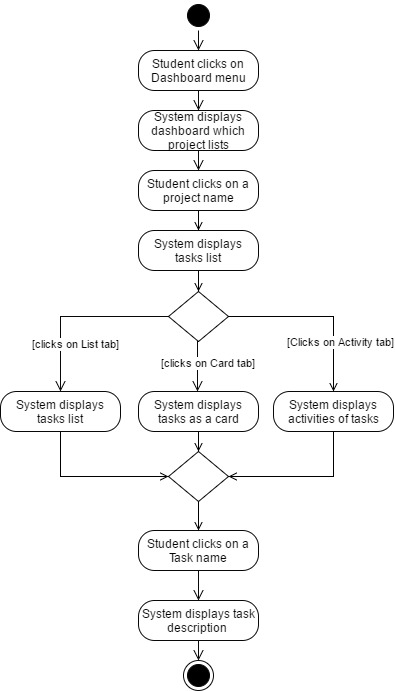


Figure 12 AD-06 View a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-07 View the statistics of tasks | | |
| **Use Case Name:** | View the statistics of tasks | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 26/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | * Student can view the task statistics of all project. * Mentor and Supervisor can view statistics of tasks in a specific student (who is under their guidance). | | |
| **Trigger:** | * Student clicks to see a dashboard page. * Mentor or Supervisor clicks to see a student’s dashboard. | | |
| **Preconditions:** | Student, Mentor, Supervisor already logged in into the system. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | The system displays a dashboard which contains statistics box, activities box, and a list of projects | | |
| **Normal Flow:** | a.) Student role:   1. Student clicks a dashboard menu to see the dashboard. 2. The system provides UI to display a dashboard which contains statistics box, activities box, and a list of projects.   b.) Mentor or supervisor role:   1. The system provides list of student in profile page. 2. Mentor or supervisor clicks on a student name. 3. The system redirects to the dashboard of a selected student. 4. The system provides UI to display a dashboard which contains statistics box, activities box, and a list of projects. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-07**

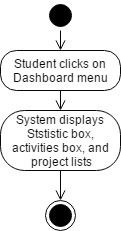


Figure 14 AD-07 View statistics of tasks (Role as a student)

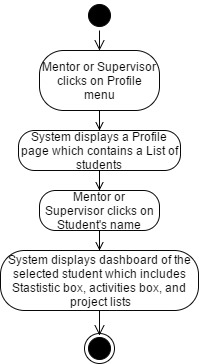


Figure 15 AD-08 View statistics of tasks (Role as a Mentor or Supervisor)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-08 Create a project | | |
| **Use Case Name:** | Create a project | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can create a new project. | | |
| **Trigger:** | Student clicks on “create new project” button. | | |
| **Preconditions:** | * Student already logged in into the system. * Student is in a dashboard. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Project name | String | - This field is not empty.  - The length is not over 225 characters. | Gravity project |
| Description | String | This field is not empty. | This is a Gravity project. |
| Start date | Date time | - This field is not empty.  - The date is selected on the providing calendar. | 13/04/2017 |
| **Post conditions:** | * If the Create a project use case is successful, the system will create a new project object and redirect to the dashboard. * If the Create a project use case is unsuccessful, the system will display an error message. | | |
| **Normal Flow:** | 1. The system provides “create new project” button in the dashboard. 2. Student clicks on “create new project” button. 3. The system provides UI to receive project name, project description, start date, “create” button, and “cancel” button. 4. Student inputs project name, description, and start date. 5. Student clicks on create button. 6. The system validates project name, description, and start date. 7. The system redirects to the dashboard. | | |
| **Alternative Flows:** | From 6.) If the project name field is empty:   1. The system displays an error message “The project name field is required.”. 2. The system returns to the step 3 of the normal flow.   From 6.) If the description field is empty:   1. The system displays an error message “The description field is required.”. 2. The system returns to the step 7 of the normal flow. | | |
| **Exceptions:** | From 5.) If student clicks on “cancel” button:   1. The system skips to the step 7 of the normal flow. | | |
| **Assumptions:** | The user understand English. | | |

**Activity diagram of UC-08**

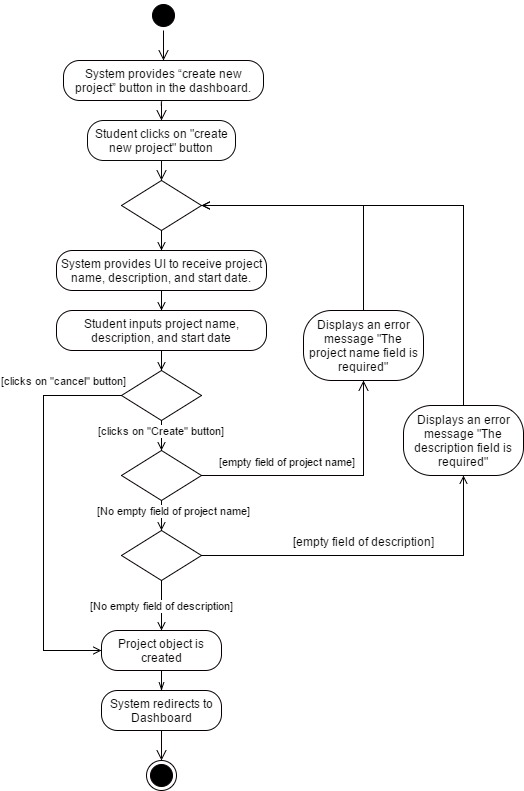


Figure 16 AD-09 Create a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-09 Edit a project | | |
| **Use Case Name:** | Edit a project | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 03/05/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can edit a project. | | |
| **Trigger:** | Student clicks on “edit” button. | | |
| **Preconditions:** | * Student already logged in into the system. * There is at least 1 project in the project list. * Student is in a dashboard. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Project name | String | - This field is not empty.  - The length is not over 225 characters. | Test project |
| Description | String | This field is not empty. | This is a test project. |
| Start date | Date time | - This field is not empty.  - The date is selected on the providing calendar. | 20/04/2017 |
| **Post conditions:** | * If the Edit a project use case is successful, the system will update a project object in database. * If the Edit a project use case is unsuccessful, the system will display an error message. | | |
| **Normal Flow:** | 1. The system provides UI to display student’s dashboard. 2. Student clicks on “edit project” button. 3. The system provides UI to receive project name, description, start date, “edit” button, and “Cancel” button. 4. Student inputs project name, description, and start date. 5. Student clicks on “edit” button. 6. The system validates project name, description, and start date. 7. The system shall save data to database. | | |
| **Alternative Flows:** | From 6.) If the project name field is empty:   1. The system displays an error message “The project name field is required.”. 2. The system returns to the step 3 of the normal flow.   From 6.) If the description field is empty:   1. The system displays an error message “The description field is required.”. 2. The system returns to the step 7 of the normal flow. | | |
| **Exceptions:** | From 5.) If student clicks in “Cancel” button:   1. System redirects to the dashboard. | | |
| **Assumptions:** | The user understands English. | | |

**Activity diagram of UC-09**

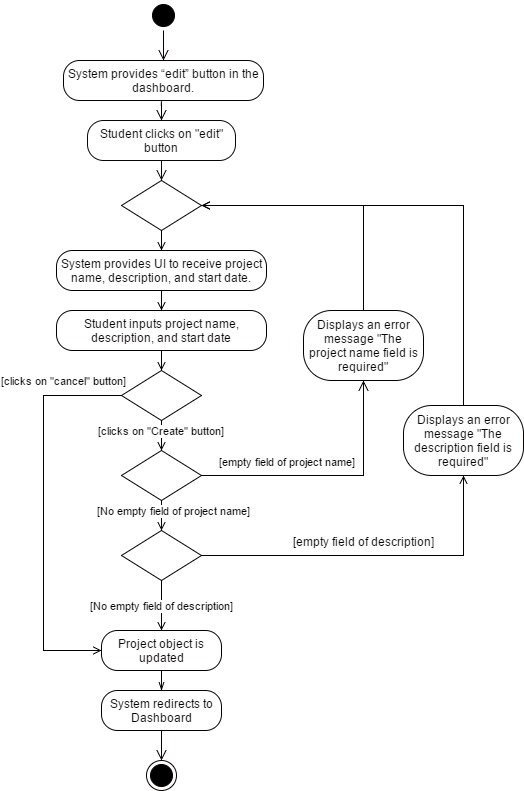


Figure 17 AD-10 Edit a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-10 Delete a project | | |
| **Use Case Name:** | Delete a project | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can delete a project. | | |
| **Trigger:** | Student click on “delete” button. | | |
| **Preconditions:** | * Student already logged in into the system. * There is at least 1 project in the project list. * Student is in a dashboard. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| - | - | - | - |
| **Post conditions:** | A project is removed from the database. | | |
| **Normal Flow:** | 1. The system provides UI to display student’s dashboard. 2. Student clicks on “delete” project button. 3. The system provides UI to display a confirmation message “Are you sure to delete this project?” with “Confirm” and “Cancel” button. 4. Student clicks “Confirm” button. 5. The system deletes a project from in database. 6. The system redirects to the dashboard | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | From 3.) If student click “Cancel” button:   1. The system skip to the step 6 of the normal flow. | | |
| **Assumptions:** | The user understands English. | | |

**Activity diagram of UC-10**

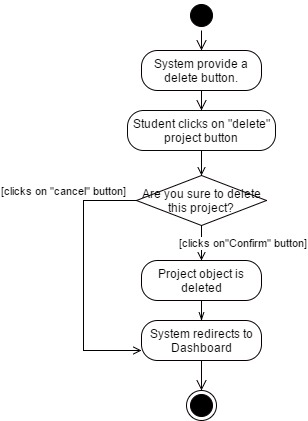


Figure 18 AD-11 Delete a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-11 Create a task | | |
| **Use Case Name:** | Create a task | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 28/02/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can create a task in a project. | | |
| **Trigger:** | Student clicks on “create new task” button. | | |
| **Preconditions:** | * Student already logged in into the system. * Student is in a Task page. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Task name | String | - This field is not empty. | Task1 |
| Description | String | - This field is not empty. | This is a task1 |
| Start date | Date time | - This field is not empty.  - The date is selected on the providing calendar | 04/04/2017 |
| **Post conditions:** | * If the Create a task use case is successful, the system will create a new task object in database. * If the Create a task use case is unsuccessful, the system will display an error message. | | |
| **Normal Flow:** | 1. Student clicks on “Create new task” button. 2. The system provides UI to receive tasks name, task description, start date, “create” button, and “cancel” button. 3. Student inputs task name, description, and start date. 4. Student clicks on “create” button. 5. The system validates task name, description, and start date. 6. The system creates a new task object in the database. 7. The system redirects to Task page. | | |
| **Alternative Flows:** | From 5.) If the task name field is empty:   1. The system displays an error message “The task name field is required.” 2. The system returns to the step 2 of the normal flow.   From 5.) If the description field is empty:   1. The system displays an error message “The task description field is required.” 2. The system returns to the step 2 of the normal flow.   From 5.) If the start date field is empty:   1. The system displays an error message “The start date field is required.” 2. The system returns to the step 2 of the normal flow. | | |
| **Exceptions:** | From 5.) If student clicks on “cancel” button:   1. The system skips to the step 2 of the normal flow. | | |
| **Assumption:** | The user understands English. | | |

**Activity diagram of UC-11**

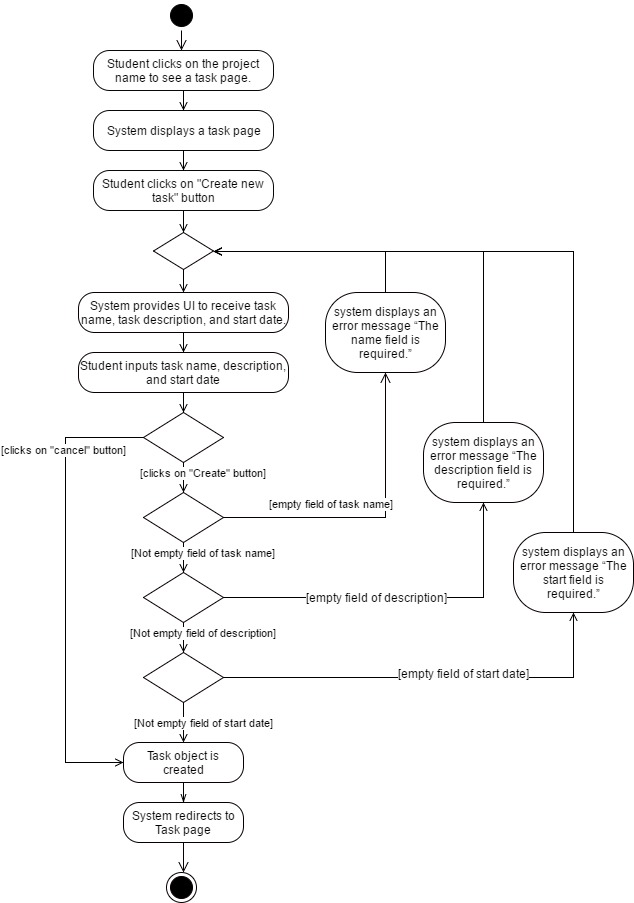


Figure 19 AD-12 Create a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-12 Edit a task | | |
| **Use Case Name:** | 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:** | * Student already logged in into the system. * Student is in a Task page. * There is a task in a project. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Task name | String | This field is not empty. | Edited task |
| Task description | String | This field is not empty | This is the edited task. |
| **Post conditions:** | * If the Edit a task use case is successful, the system will update a task object in database. * If the Edit a task use case is unsuccessful, the system will display an error message. | | |
| **Normal Flow:** | 1. Student clicks on “edit task” button. 2. The system provides UI to receive tasks name, task description, “update” button, and “cancel” button. 3. Student inputs task name and description. 4. Student clicks on “update” button. 5. The system validates tasks name and task description. 6. The system updates a task object in the database. 7. The system redirects to Task page. | | |
| **Alternative Flows:** | From 5.) If the task name field is empty:   1. The system displays an error message “The task name field is required.” 2. The system returns to the step 2 of the normal flow.   From 5.) If the task description field is empty:   1. The system displays an error message “The task description field is required.” 2. The system returns to the step 2 of the normal flow. | | |
| **Exceptions:** | From 5.) If student click on “cancel” button:   1. The system skips to the step 7 of the normal flow. | | |
| **Assumptions:** | The user understands English. | | |

**Activity diagram of UC-12**

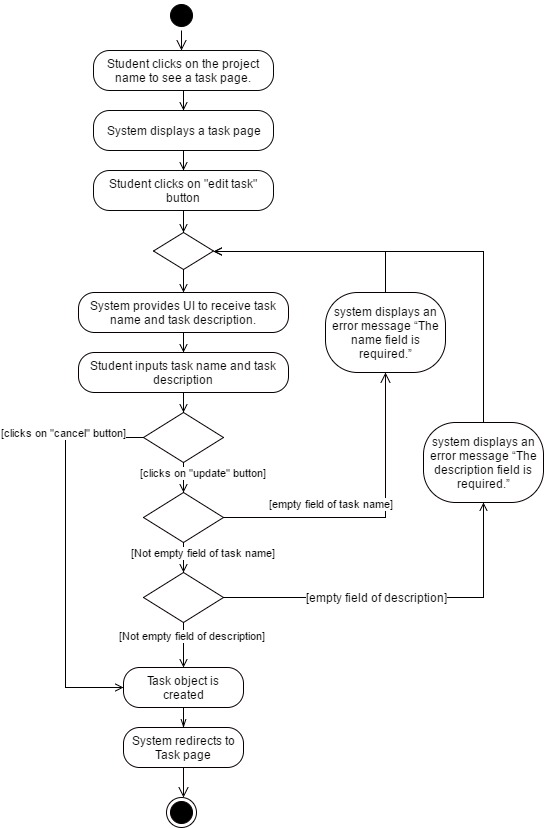


Figure 20 AD-13 Update a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-13 Delete a task | | |
| **Use Case Name:** | 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:** | * Student already logged in into the system. * Student is in a Task page. * There is a task in the selected project. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | Task object is deleted from the database. | | |
| **Normal Flow:** | 1. Student clicks on “delete task” button. 2. The system provides UI to display a confirmation message “Are you sure to delete this task?” with “Confirm” and “Cancel” button. 3. Student click “Confirm” button. 4. The system deletes the selected task from the database. 5. The system redirects to Task page. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | From 5.) If student click “Cancel” button:   1. The system skips to the step 5 of the normal flow. | | |
| **Assumptions:** | The user understands English. | | |

**Activity diagram of UC-13**

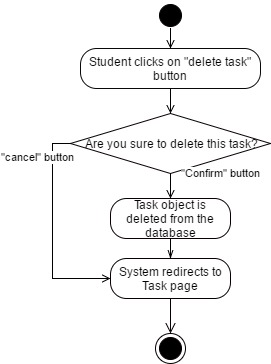


Figure 21 AD-14 Delete a task

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-14 Change a status of a task | | |
| **Use Case Name:** | 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:** | * Student already logged in into the system. * Student is in the Task page. * There is a task in the selected project. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **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. The 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. The system validates the start date. 7. The system changes the task status.   Flow b.) Drag and drop   1. Student clicks on “CARD” tab in the Task page. 2. The system displays a task in card style. 3. Student drags a task card. 4. Student drops a task card. 5. The system provides UI to select the start date with “confirm” button, and “Cancel” button. 6. Student selects start date. 7. Student click “Confirm” button. 8. The system validates the start date. 9. The 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. The system provides UI to display task detail with the status button. 3. Student clicks on the status buttons. 4. The 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. The system validates the start date. 8. The system changes the task status. | | |
| **Alternative Flows:** | From a-5.) If student does not select any start date:   1. System locks the “Confirm” button. 2. The system returns to the step a-3 of the normal flow.   From b-4.) If student does not drop a card in any column:   1. System displays forbidden sign as a mouse cursor. 2. System locks for changing the status of a task. 3. The system returns to the step b-5 of the normal flow.   From b-4.) If student drops a card to the same column:   1. The system remains a card task in the same column. 2. The system returns to the step b-5 of the normal flow.   From b-7.) If student does not select any start date:   1. System locks the “Confirm” button. 2. The system returns to the step b-5 of the normal flow.   From c-5.) If student does not select any start date   1. System locks the “Confirm” button. 2. The system returns to the step c-4 of the normal flow. | | |
| **Exceptions:** | From a-5, b-7, and c-6.) If student selects cancel button   1. Student clicks Cancel button. 2. The system discards to change the status of task. 3. The system redirect to the task page. | | |
| **Assumptions:** | The user understands English. | | |

**Activity diagram of UC-14**

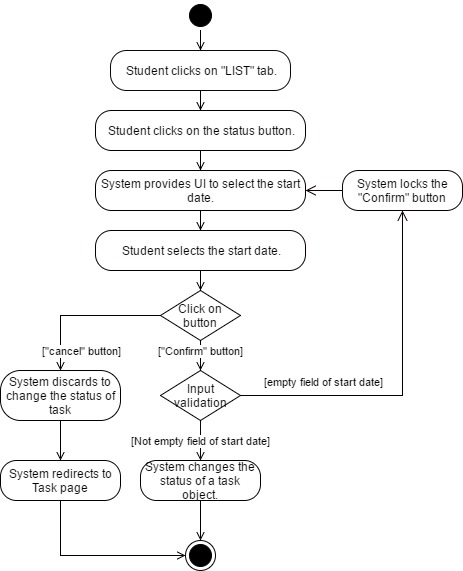


Figure 22 AD-15 Change a task status (List tab)

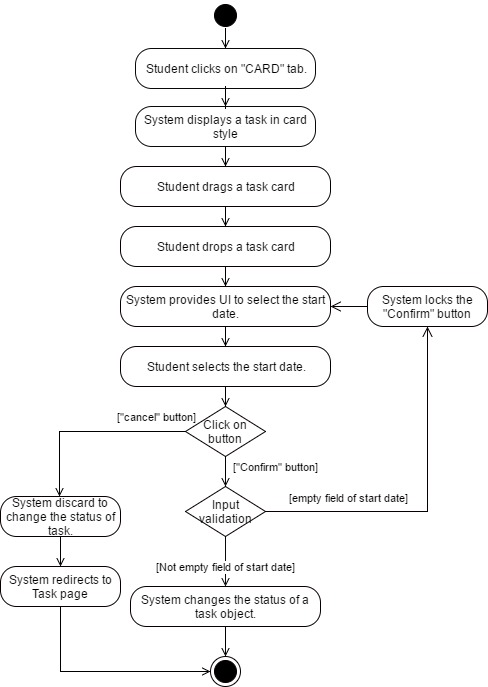


Figure 23 AD-16 Change a task status (Card tab)

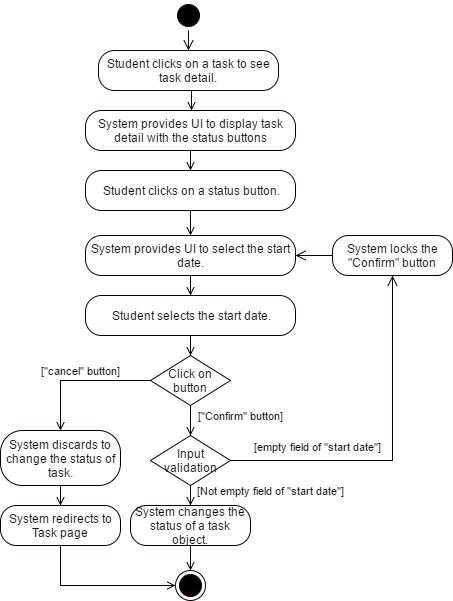


Figure 24 AD-17 Change a task status (in task detail page)

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-15 View a comment | | |
| **Use Case Name:** | View a comment | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can view a comment in the task. | | |
| **Trigger:** | Student, Mentor, Supervisor clicks to see a task detail. | | |
| **Preconditions:** | * Student, Mentor, Supervisor already logged in into the system. * Student, Mentor, Supervisor is in the Task page. * There is a task in the selected project. * There is a comment in the selected task. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | If View comment use case is successful, the system displays a comment of a task. | | |
| **Normal Flow:** | 1. Student, Mentor, Supervisor clicks on a task name to see a task detail. 2. The system requests a task from database. 3. The system provides UI to display a task detail, comment field, and status buttons. 4. Student, Mentor, Supervisor can see a comment. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-15**

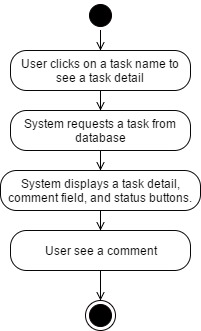


Figure 25 AD-18 View a comment

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-16 Create a comment | | |
| **Use Case Name:** | Create a comment | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student, Mentor, and Supervisor | | |
| **Description:** | Student, Mentor, and Supervisor can create a comment on each task. | | |
| **Trigger:** | Student, Mentor, and Supervisor clicks on “comment” button. | | |
| **Preconditions:** | * Student, Mentor, Supervisor already logged in into the system. * Student, Mentor, Supervisor is in the Task detail page. * There is a task in the selected project. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| Comment | String | This field is not empty. | This is a test comment. |
| **Post conditions:** | If the Crete a comment use case is successful, the system will create a comment object, task activity object, and display a notification popup message. | | |
| **Normal Flow:** | 1. Student, Mentor, Supervisor clicks on a task name to see a task detail. 2. The system provides UI to display task detail, status buttons, and comment field with “comment” button. 3. Student, Mentor, Supervisor input a text in a comment field. 4. Student, Mentor, Supervisor clicks “comment” button. 5. The system validates a comment field. 6. The system saves the comment data into the database. 7. The system displays a comment. | | |
| **Alternative Flows:** | From 3.) If the comment field is empty:   1. The system locks a comment button. 2. The system returns to the step 2 of the normal flow | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-16**

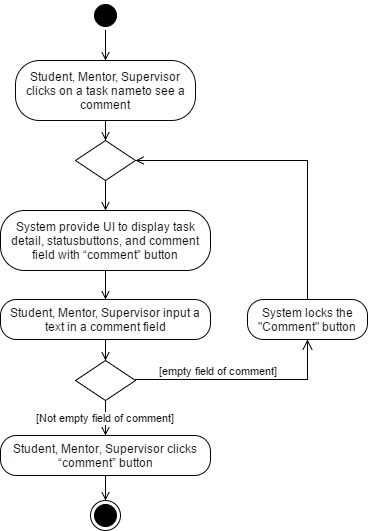


Figure 26 AD-19 Create a comment

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-17 Delete a comment | | |
| **Use Case Name:** | Delete a comment | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 03/03/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student, Mentor, and Supervisor | | |
| **Description:** | Student, Mentor, and Supervisor can delete their own comment by clicking on the bin button. | | |
| **Trigger:** | Student, Mentor, and Supervisor clicks on a bin icon on the top right of the comment box. | | |
| **Preconditions:** | * Student, Mentor, Supervisor already logged in into the system. * Student, Mentor, Supervisor is in the Task detail page. * There is a comment of the owner in the selected task. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | If the Delete a comment use case is successful, the system will delete a comment object from database, create a task activity object, and display a notification popup message. | | |
| **Normal Flow:** | 1. The system provides a bin icon to delete their own comment. 2. Student, Mentor, and Supervisor clicks a bin icon. 3. The system displays a confirmation message “Are you sure to delete this comment” with “Confirm” button and “Cancel” button. 4. Student, Mentor, and Supervisor clicks “Confirm” button. 5. The system deletes the comment in the database. 6. The system displays a message “Delete comment success” 7. The system redirects to the task detail page. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | From 4.) If Student, Mentor, and Supervisor clicks on “cancel” button.   1. The system discards to delete a comment. 2. The system returns to the step 7 of the normal flow. | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-17**

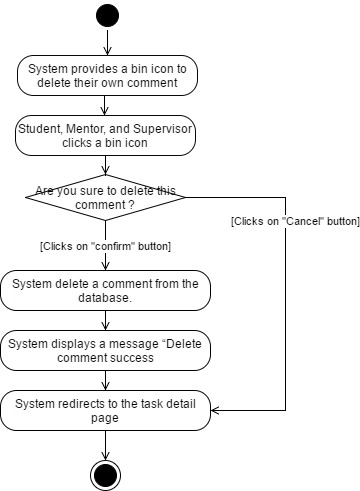


Figure 27 AD-20 Delete a comment

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-18 View activities in a project | | |
| **Use Case Name:** | View activities in a project | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 22/04/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can view all student’s activities of a specific project. The activity can be, for example, creating project, editing project, creating task, editing task, delete task, creating a comment. | | |
| **Trigger:** | Student, Mentor, and Supervisor clicks on “Activities” tab. | | |
| **Preconditions:** | * Student, Mentor, Supervisor already logged in into the system. * Student, Mentor, Supervisor is in the task page. * Mentor and Supervisor are in a student’s dashboard. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | If View activities in a project use case is successful, the system displays all activities of a selected project | | |
| **Normal Flow:** | 1. Student, Mentor, Supervisor clicks on a project name. 2. The system provides UI to display task statistics, List tab, Card tab, and activities tab. 3. Student, Mentor, Supervisor clicks “Activities” tab. 4. The system displays all activities of a selected project. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-18**

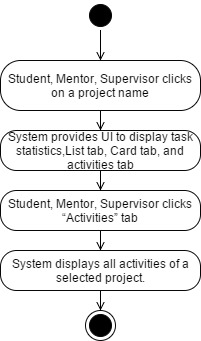


Figure 28 AD-21 View activitiesin a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-19 View activities of the under guidance students | | |
| **Use Case Name:** | View activities of the under guidance students | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/05/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Mentor, Supervisor | | |
| **Description:** | Mentor and Supervisor can view all activities of all under guidance students. The activity can be, for example, creating project, editing project, creating task, editing task, delete task, creating a comment. | | |
| **Trigger:** | Mentor and Supervisor clicks on the dashboard menu. | | |
| **Preconditions:** | * Mentor, Supervisor already logged in into the system. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | If View activities of the under guidance students use case is successful, the system displays dashboard with task statistics, and all activities of students. | | |
| **Normal Flow:** | 1. Mentor and Supervisor clicks on the dashboard menu. 2. The system provides UI to display dashboard with task statistics, and all activities of students. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-19**

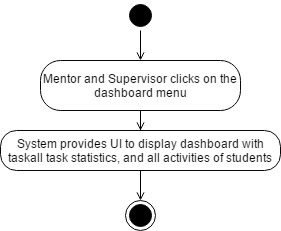


Figure 29 AD-22 View activities of the under guidance students

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-20 Generate a weekly report | | |
| **Use Case Name:** | Generate a weekly report | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student | | |
| **Description:** | Student can generate a weekly report in the web application. | | |
| **Trigger:** | Student clicks on “Report” menu. | | |
| **Preconditions:** | * Student already logged in into the system * There is a task in a database. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | The system display a weekly report which is separated by the project. | | |
| **Normal Flow:** | 1. Student clicks on “Report” menu. 2. The system requests the task from database. 3. The system arranges task details in the report template. 4. The system provides UI to display a weekly report which is separated by the project. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-20**

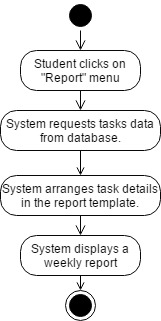


Figure 30 AD-23 Generate a weekly report

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-21 Print a weekly report | | |
| **Use Case Name:** | 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:** | * Student already logged in into the system. * Student already generated a report. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | A weekly report file is sent to a printer. | | |
| **Normal Flow:** | 1. Student clicks on “Print” button. 2. The system provides UI to see a preview of a report with “Print” and “Cancel” button. 3. Student clicks “print” button. 4. The system sends a weekly report to a printer. 5. The system redirects to the report page. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | From 3.) If student clicks on “cancel” button.   1. The system discards to print a report. 2. The system skips to the step 5 of the normal flow. | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-21**

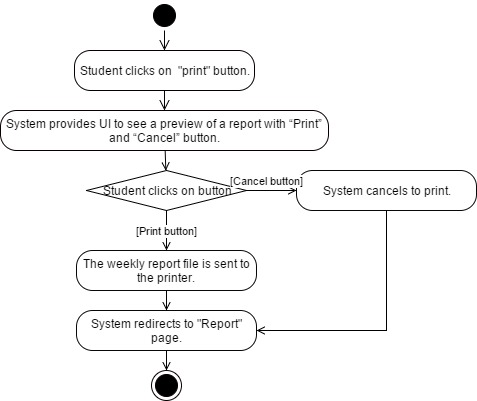


Figure 31 AD-24 Print a report

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-22 Receive a notification via Web application | | |
| **Use Case Name:** | Receive a notification via Web application | | |
| **Created By:** | Phinthip | **Last Updated By:** | Phinthip |
| **Date Created:** | 04/03/2017 | **Last Revision Date:** | 04/05/2017 |
| **Actors:** | Student, Mentor, Supervisor | | |
| **Description:** | Student, Mentor, Supervisor can receive notification via Web application | | |
| **Trigger:** | A new activity is created. The activity can be, for example, creating project, editing project, creating task, editing task, delete task, creating a comment. | | |
| **Preconditions:** | Student, Mentor, Supervisor already logged in into the system. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | - | - | - |
| **Post conditions:** | Notify object is created. | | |
| **Normal Flow:** | 1. The system detects a new activity. 2. The system sends a request to Firebase service. 3. The system observes data in Firebase. 4. The system displays an activity notification pop-up on the website, on “Today” notification tab, and on “All” tab. 5. Student, Mentor, Supervisor receives an activity notification. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-22**

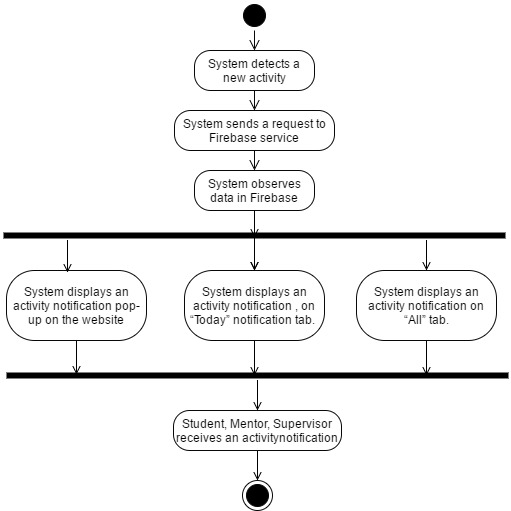


Figure 32 AD-25 Web notification

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-23 Receive an activity notification via Email | | |
| **Use Case Name:** | 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. The activity can be, for example, creating project, editing project, creating task, editing task, delete task, creating a comment. | | |
| **Preconditions:** | Student, Mentor, Supervisor already logged in into the system.  The system have an email of user. | | |
| **Use Case Input Specification** | | | |
| **Input** | **Type** | **Constraint** | **Example** |
| **-** | **-** | **-** | **-** |
| **Post conditions:** | - | | |
| **Normal Flow:** | 1. System detects a new activity. 2. System send a request to EmailSender. 3. User receives an email of activity notification. | | |
| **Alternative Flows:** | - | | |
| **Exceptions:** | - | | |
| **Assumptions:** | The user can understand English. | | |

**Activity diagram of UC-23**

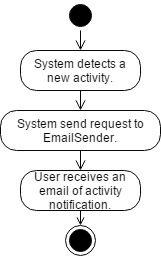


Figure 33 AD-26 Mail notification