Website for study

Version <2.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09/11/2022 | 1.0 |  | PTAn, NTBHa, TKDuy |
| 16/11/2022 | 1.0 |  | PTAn, NTBHa, TKDuy |
| 30/11/2022 | 2.0 |  | PTAn, NTBHa, TKDuy |
|  |  |  |  |

Table of Contents

1. Use-case Model 4

1.1 Register 4

1.2 Login 4

1.3 Logout 4

1.4 Contact support 5

1.5 Search 5

1.6 Reminder 5

1.7 Pomodoro 6

1.8 Notion 6

1.9 ToDo List 7

1.10 Full use case 7

2. Use-case Specifications 9

2.1 Use-case: Register 9

2.2 Use-case: Login 9

2.3 Use-case: Logout 9

2.4 Use-case: Contact support 10

2.5 Use-case: Search 10

2.6 Use-case: Reminders 10

2.7 Use-case: Pomodoro timer 11

2.8 Use-case: Create note 11

2.9 Use-case: Delete note 12

2.10 Use-case: Create Todo List 12

2.11 Use-case: Delete Todo List 12

2.12 Use-case: Delete account 13

# Use-case Model

## Register

Diagram

Description automatically generated

*H1. Register*

## Login

Diagram

Description automatically generated

*H2. Login*

## Logout

A picture containing text, sport

Description automatically generated

*H3. Logout*

## Contact support

Diagram, venn diagram

Description automatically generated

*H4. Contact support*

## Search

Diagram, venn diagram

Description automatically generated

*H5. Search*

## Reminder

Diagram

Description automatically generated

*H6. Reminder*

## Pomodoro

Diagram

Description automatically generated

*H7. Pamodoro*

## Notion

Diagram

Description automatically generated

*H8. Notion*

## ToDo List

Diagram

Description automatically generated

*H9. ToDoList*

## Full use case

*H10. Full use case*

Diagram

Description automatically generated

# Use-case Specifications

## Use-case: Register

|  |  |
| --- | --- |
| Use case Name | Register |
| Brief description | This use case describes how the user can register |
| Actors | User, Administrator |
| Basic Flow | 1. At the homepage, the user click on Signup 2. User fill the Signup form 3. User submit 4. System check if information is valid 5. System finish Signup if valid |
| Alternative Flows | **Alternative flow 1: Invalid username**  Request re-entry |
| Pre Conditions | Account name does not exist in the database yet. |
| Post Conditions | The user’s details are stored in the database. |

## Use-case: Login

|  |  |
| --- | --- |
| Use case Name | Login |
| Brief description | This use case describes how the user can login |
| Actors | User, Administrator |
| Basic Flow | 1. At the homepage, the user/admin click on Signin 2. User fill the Signin form 3. User submit 4. System check if information is valid 5. System finish Signin if valid |
| Alternative Flows | **Alternative flow 1: Invalid Sign in (Wrong login credentials)**  Users key in the wrong combination of username and password  Use case ends  Request re-entry |
| Pre Conditions | The user’s account are stored in the database. |
| Post Conditions | The user has access to the system |

## Use-case: Logout

|  |  |
| --- | --- |
| Use case Name | Logout |
| Brief description | This use case describes how the user can logout |
| Actors | User, Administrator |
| Basic Flow | 1. Use case begins when admin/user clicks on “Logout” 2. The system will move the user session and the user is logged out. |
| Alternative Flows | None |
| Pre Conditions | Account is logged in |
| Post Conditions | The user is logged out |

## Use-case: Contact support

|  |  |
| --- | --- |
| Use case Name | Contact support |
| Brief description | This use case describes how the user contact administrator for help |
| Actors | User, Administrator |
| Basic Flow | 1. At the homepage, the user click “Contact us” 2. User fill the form 3. User submit 4. System check and send to admin 5. Admin will be support user |
| Alternative Flows | None |
| Pre Conditions | User have an account. |
| Post Conditions | The user will connect and get help from admin. |

## Use-case: Search

|  |  |
| --- | --- |
| Use case Name | Search |
| Brief description | This use case describes how the user/admin can search note/list in database |
| Actors | User, Administrator |
| Basic Flow | 1. At the homepage, the user click “Search” 2. Input keyword 3. System will check and compare with database 4. Give suggestions |
| Alternative Flows | **Alternative flow 1:** Keyword does not exist  Continue to save normally. |
| Pre Conditions | User have an account. |
| Post Conditions | The user can find note/list in database faster |

## Use-case: Reminders

1. **Create – edit reminders**

|  |  |
| --- | --- |
| Use case Name | Create – edit reminders |
| Brief description | This use case describes how the user create (edit) reminders. |
| Actors | User |
| Basic Flow | 1. At the homepage, the user clicks on the “Reminder” button. 2. System displays the reminder page 3. User set reminder’s name, deadlines,… 4. User clicks the “Done” button, and then clicks the “Add” button.   If the user wants to edit, also choose the “Reminder” button, then click on the reminder wants to edit, edit and click the “Done” button. |
| Alternative Flows | **Alternative flow 1:** The user created a reminder and forgot to click the “Add” button.  **Alternative flow 2:** The user edited a reminder andforgot to click the “Done” button. |
| Pre-conditions | User access to the website |
| Post-conditions | The user successfully adds a new reminder (or edit reminder) in the personal page. |

1. **Delete reminder**

|  |  |
| --- | --- |
| Use case Name | Delete reminder |
| Brief description | This use case describes how the user/admin delete reminders. |
| Actors | User, Administrator |
| Basic Flow | 1. Use case begins when admin/user clicks on “Delete reminder” 2. The system will delete this reminder. |
| Alternative Flows | **None** |
| Pre-conditions | Have the reminder in database |
| Post-conditions | The reminder has been deleted |

## Use-case: Pomodoro timer

|  |  |
| --- | --- |
| Use case Name | Set, Pomodoro timer |
| Brief description | This use case describes how the user and administrator use Pomodoro timer |
| Actors | User, Administrator |
| Basic Flow | 1. At the homepage, the user click to the “Pomodoro” button. 2. System displays the timer page 3. User set Time Duration,... 4. User click Start 5. The timer start to run. |
| Alternative Flows | **Alternative flow 1: User does not star the timer and exit** |
| Pre-conditions | User access to the website |
| Post-conditions | Pomodoro timer runs and go off when the time comes. |

## Use-case: Create note

|  |  |
| --- | --- |
| Use case Name | Create note. |
| Brief description | This use case describes how the user do with notes. |
| Actors | User |
| Basic Flow | 1. At the homepage, the user click to the “Notes” button. 2. System displays the place to write and edit note. 3. User clicks ‘Done’ button to add new notes. 4. Or clicks ‘Edit’ button to edit note. 5. System displays the result. |
| Alternative Flows | **Alternative flow 1: Users did not edit the note although they choose to edit**  Continue to save normally. |
| Pre-conditions | User goes to the study web and have an account. |
| Post-conditions | The user successfully add new notes or edit notes. |

## Use-case: Delete note

|  |  |
| --- | --- |
| Use case Name | Delete note. |
| Brief description | This use case describes how the user and administrator delete note. |
| Actors | User, Administrator |
| Basic Flow | 1. Use case begins when admin/user clicks on “Delete note” 2. The system will delete note from database |
| Alternative Flows | **None** |
| Pre-conditions | Have note in database |
| Post-conditions | The user successfully delete note from database |

## Use-case: Create Todo List

|  |  |
| --- | --- |
| Use case Name | Create TODO List. |
| Brief description | This use case describes how the user and administrator do with TODO lists. |
| Actors | User, Administrator |
| Basic Flow | 1. At the homepage, the user click to the “List” button. 2. System displays a new list. 3. User can rename list and create tag. 4. User clicks ‘Add’ button to create new note or move old note to list. 5. User clicks ‘Done’ button to final. 6. System displays the result. |
| Alternative Flows | **Alternative flow 1: Users did not add note although they choose to add**  Continue to save normally.  **Alternative flow 2: Users did not add note after create new list**  Continue to save normally.  **Alternative flow 3: Users join or move notes to new list**  Move notes from old list to new list |
| Pre-conditions | User goes to the study web and have an account. |
| Post-conditions | The user successfully create (edit) todo list or add/move notes to list or update to the list of notes on account. |

## Use-case: Delete Todo List

|  |  |
| --- | --- |
| Use case Name | Delete TODO List. |
| Brief description | This use case describes how the user and administrator delete TODO lists. |
| Actors | User, Administrator |
| Basic Flow | 1. Use case begins when admin/user clicks on “Delete list” 2. The system will delete list (delete all notes in list). |
| Alternative Flows | **None** |
| Pre-conditions | Have the todo list in database |
| Post-conditions | The user successfully delete todo list (delete all notes in list) . |

## Use-case: Delete account

|  |  |
| --- | --- |
| Use case Name | Delete account |
| Brief description | This use case describes how the user/admin can delete account |
| Actors | User, Administrator |
| Basic Flow | 1. Use case begins when admin/user clicks on “Delete account” 2. The system will move the user session and delete the user’s details in the database. |
| Alternative Flows | **None** |
| Pre Conditions | The user’s account are stored in the database. |
| Post Conditions | The user’s account and details in the database has been deleted. |