

SWP391

**Report 3 – Software Requirement Specification**

– Danang, April 2023 –

Table of Contents

[Table of Tables 4](#_Toc134996273)

[I. Record of Changes 1](#_Toc134996274)

[II. Software Requirement Specification 2](#_Toc134996275)

[1. Product Overview 2](#_Toc134996276)

[*2.* User Requirements 3](#_Toc134996277)

[2.1 Actors 3](#_Toc134996278)

[2.2 Use Cases 3](#_Toc134996279)

[2.3 Feature Common 13](#_Toc134996280)

[3. Functional Requirements 86](#_Toc134996281)

[*3.1* System Functional Overview 86](#_Toc134996282)

[3.2 Authentication 95](#_Toc134996283)

[3.3 Homepage 98](#_Toc134996284)

[3.4 User Profile 100](#_Toc134996285)

[3.5 Course Learning 101](#_Toc134996286)

[3.6 Taking Mock Test 105](#_Toc134996287)

[3.7. Earn Coin 107](#_Toc134996288)

[3.8 My Blog 108](#_Toc134996289)

[3.9 About Us 111](#_Toc134996290)

[3.10 Knowledge Sharer Analyst Data 113](#_Toc134996291)

[3.11 Admin Analyst Data 122](#_Toc134996292)

[4. Non-Functional Requirements 132](#_Toc134996293)

[4.1 External Interfaces 132](#_Toc134996294)

[4.2 Quality Attributes 133](#_Toc134996295)

[*5.* Requirement Appendix 134](#_Toc134996296)

[5.1 Business Rules 134](#_Toc134996297)

[5.2 Common Requirements 134](#_Toc134996298)

[5.3 Application Messages List 136](#_Toc134996299)

Table of Figures

[Figure 1. Overview of Flearning Platform 2](#_bookmark3)

[Figure 2. Flearning Platform Use Case Diagram 4](#_bookmark8)

[Figure 3. Use Case for Authentication 5](#_bookmark9)

[Figure 4. Use Case for Learner 6](#_bookmark10)

[Figure 5. Use Case for Knowledge Sharer 7](#_bookmark11)

[Figure 6. Use Case for Admin 8](#_bookmark12)

[Figure 7. Swimlane diagram for Course creation and learning 9](#_bookmark13)

[Figure 8. Screen Flows for each actor in FLearing Platform 48](#_bookmark18)

[Figure 9. Entity Relationship Diagram 53](#_bookmark22)

[Figure 10. Sign In Screen 54](#_bookmark25)

[Figure 11. Sign Up Screen 55](#_bookmark26)

[Figure 12. Forgot Password Screen 56](#_bookmark27)

[Figure 13. Change Password Screen 57](#_bookmark28)

[Figure 14. Homepage Screen 58](#_bookmark30)

[Figure 15. My Profile Screen 59](#_bookmark32)

[Figure 16. Other User’s Wall Screen 60](#_bookmark33)

[Figure 17. Course Overview Screen 61](#_bookmark35)

[Figure 18. Course Detail Screen 62](#_bookmark36)

[Figure 19. Course Detail Question Screen 63](#_bookmark37)

[Figure 20. Course Detail Video Screen 64](#_bookmark38)

[Figure 21. All Mock Test Screen 65](#_bookmark40)

[Figure 22. Mock Test Detail Screen 66](#_bookmark41)

[Figure 23. Earn Coin Screen 67](#_bookmark43)

[Figure 24. All Blogs Screen 68](#_bookmark45)

[Figure 25. Blog Detail Screen 69](#_bookmark46)

[Figure 26. My Blog Screen 70](#_bookmark47)

[Figure 27. About Us Screen 71](#_bookmark49)

[Figure 28. Knowledge Dashboard Screen 72](#_bookmark51)

[Figure 29. Knowledge Sharer’s Course Screen 73](#_bookmark52)

[Figure 30. Knowledge Sharer’s Video Screen 74](#_bookmark53)

[Figure 31. Knowledge Sharer’s Chapter Screen 75](#_bookmark54)

[Figure 32. Knowledge Sharer’s Question Screen 76](#_bookmark55)

[Figure 33. Knowledge Sharer’s Mock Test Screen 77](#_bookmark56)

[Figure 34. Knowledge Sharer’s Enroller Screen 78](#_bookmark57)

[Figure 35. Knowledge Sharer’s Income Screen 79](#_bookmark58)

[Figure 36. Knowledge Sharer’s Comment Screen 80](#_bookmark59)

[Figure 37. Admin Dashboard Screen 81](#_bookmark61)

[Figure 38. Admin’s User Screen 82](#_bookmark62)

[Figure 39. Admin’s Course Screen 83](#_bookmark63)

[Figure 40. Admin’s Video Screen 84](#_bookmark64)

[Figure 41. Admin’s Question Screen 85](#_bookmark65)

[Figure 42. Admin’s Transaction Screen 86](#_bookmark66)

[Figure 43. Admin’s Comment Screen 87](#_bookmark67)

[Figure 44. Admin’s Blog Screen 88](#_bookmark68)

[Figure 45. Admin’s Mock Test Screen 89](#_bookmark69)

[Figure 46. Admin’s Notification Screen 90](#_bookmark70)

# Table of Tables

[Table 1. All actors in the system 3](#_bookmark6)

[Table 2. Use case Description 11](#_bookmark14)

[Table 3. Screen Descriptions 50](#_bookmark19)

[Table 4. Screen Authorization 52](#_bookmark20)

[Table 5. Non-Screen Functions 52](#_bookmark21)

[Table 6. Entities Description 54](#_bookmark23)

[Table 7. Business Rule 92](#_bookmark76)

[Table 8. Common Requirements 93](#_bookmark78)

[Table 9. Application Message List 96](#_bookmark80)

# Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **A\*, M, D** | **In charge** | **Change Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# Software Requirement Specification

## Product Overview

Nowadays, with the increasing number of students enrolling at FPT University each year, the issue of student accommodation management is becoming crucial. Currently, most accommodation management is done manually using paper-based records and documents, which is time-consuming and prone to errors.

Applying information technology to dormitory management is essential to facilitate easier management, avoid data loss, and enable quick and efficient processing of information with high accuracy. It saves time and effort for managers and enhances the overall experience of students. To address these challenges and provide a more reasonable approach to dormitory management with the application of information technology, we propose developing a Student Dormitory Management System at FPT University.

The Student Dormitory Management System will assist in managing room allocation, accommodation registration, as well as storing and processing information about students and the dormitory situation. It will simplify the management process and save time.

**Figure 1. Overview of FPT Dormitory Management**

## User Requirements

### Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Guest | Guest is a person who accesses the FPT Dormitory Management system without logging. They have access to the landing page and can view announcements, service and contact information about dormitory. |
| 2 | Student | Students are individuals who have an account in the FPT Dormitory Management system. They have the ability to logging to their account using valid credentials, manage their information, view available rooms and their details (e.g., room types, occupancy, amenities, price,…), reserve a room for a specific academic term and receive confirmation of their room reservation and allocation, check-out from the dormitory at the end of each academic term or when vacating the room, submit maintenance requests, billing and payment, receive important updates, deadlines, and relevant information through the system . |
| 3 | Manager | Manager who have full control over the system such as users’ accounts management, room management, room allocation and assignment in the FPT Dormitory Management system... will be called Manager. |

**Table 1. All actors in the system**

**Guest :**

● Sign up (with ID student)

● View landing page

● View service information page

● View contact information page

● View dormitory rules

**User :**

● Logging (with ID student)

● Manage personal profile (Update or change password for account)

● View all rooms and their details (e.g., room types, occupancy, amenities, price,…)

● View roommates information.

● Search by categories (e.g., gender, room types, occupancy, amenities, price,…)

● Reserve a room for a specific academic term (fill form to provide with necessary information and procedures for the check-in process)

● Billing and payment for a specific academic term (e.g., room before each semester, the amount of electricity and water used by month, water and electric bill after each semester)

● Request to change room with another student

● Submit maintenance or recommendations requests for manager

**Manager :**

● Logging (with Manager ID)

● Manage announcements (CRUD an announcement)

● Manage reserve rooms (change the status of the student's room rental request after the room fee has been paid or delete the request reserve room)

● Manage student information (View all student account and their information – update – delete – search)

● Manage room (CRUD a room)

● Manage collect room fee and other fees (Check the payment status for room fee/electricity fee/water fee, Send total bill)

● Statistic (Total number of rented accommodation (all/male/female), total number of unregistered spaces (all/male/female), total student’s room fees paid)

### Use Cases

* + - 1. *Diagram(s)*

###### Use Case Overview

Below is an overview of the use cases for the FPT Dormitory Management System.

A screenshot of a black screen

Description automatically generated with low confidence

**Figure 2. FPT Dormitory Management System Use Case Diagram**

###### Use Case for Authentication

In the FPT Dormitory Management system, all actors except for guests are required to go through authentication upon entry. This process verifies the identity of the user and ensures that only authorised individuals are granted access. By enforcing authentication, the system can protect sensitive information and prevent unauthorised access.

A picture containing screenshot, circle

Description automatically generated

**Figure 3. Use Case for Authentication**

###### Use Case for Student

Below is a subsystem use case for the Student actor.

A screenshot of a cell phone

Description automatically generated with medium confidence

**Figure 4. Use Case for Student**

###### Use Case Manager

A picture containing screenshot, text, black and white, black

Description automatically generatedBelow is a subsystem use case for the Manager actor.

**Figure 5. Use Case for Manager**

**A picture containing text, diagram, screenshot, parallel

Description automatically generated**

#### 2.2.3 Descriptions

Below is a summary of the use cases present in the FLearning system:

|  |  |  |
| --- | --- | --- |
| **ID** | **Use Case** | **Actors** |
| UC-1 | Sign Up | Guest |
| UC-2 | Sign In | User, Manager |
| UC-3 | Sign Out | User, Manager |
| UC-4 | Change Password | User, Manager |
| UC-5 | Manage profile | User, Manager |
| UC-6 | View Landing Page | Guest, User, Manager |
| UC-7 | Manage student information | Manager |
| UC-8 | View all student information | Manager |
| UC-9 | Update student information | Manager |

**Table 2. Use case Description**

### 2.3 Feature Common

#### 2.3.1 Authenticate User

##### 2.3.1.1 Sign Up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID: | **UC-01** | | Use Case Name: | **Sign Up** |
| Created By: | HungPN | | Date Created: | 2023/05/21 |
| Primary Actor: | Guest | | Secondary Actor: | None |
| Description: | This use case describes the process of new visitors creating an account on FDorm website, provide the student ID to become a registered user. | | | |
| Priority: | High | | | |
| Trigger: | The guest clicks on the "Tạo tài khoản" link in the Sign in page or accesses the sign-up page with a link. | | | |
| Preconditions: | PRE-01: Guest has a valid email student ID. | | | |
| Post - Conditions: | POST-1: FDorm system works stably, without error.  POST-2: The guest can use their account to login to FDorm system. | | | |
| Basic Flow: | 1. 2. 3. 4.  5. | The guest accesses the landing page.  Click on “Đăng nhập” button.  System redirects to the sign-in page.  Click on “Tạo tài khoản” link.  Type in all the required fields needed to create an account. | | |
|  | 6. | Click on button “Tạo tài khoản”. | | |
|  | 8. | System redirects to the sign-in page. | | |
| Alternative Flow: | None |  | | |
| Exception Flow: | **5a. The guest fills in invalid username (student ID).**  5a1. The system gives an error message and inform the guest to fill in another username.  **5b. The guest fills in a username (student ID) already exists.**  5b1. The system gives an error message and inform the guest to fill in another username. | | | |
| Business Rules: | BR-02, BR-03, BR-04 | | | |

##### 2.3.1.2 Sign In

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-02** | Use Case Name: | **Sign In** |
| Created By: | HungPN | Date Created: | 2023/05/21 |
| Primary Actor: | User, Manager | Secondary Actor: | None |
| Description: | This use case describes how a user can sign in to their account. The system verifies the credentials and grants access to the user's account. | | |
| Priority: | High | | |
| Trigger: | Users want to login to the FDorm system using a Fdorm account. | | |
| Preconditions: | PRE-01: User has a created and valid account. PRE-02: User account has been authorised. | | |
| Post - Conditions: | POST-1: Fdorm system works stably, without error.  POST-2: User successfully logged into the Fdorm system. | | |
| Basic Flow: | 1. User access to the sign-in page. 2. User to choose login with Fdorm account. 3. User type in account info and press sign in button. 4. System verifies the user's account information and allows the user to access the system. 5. System redirects to Landing page. | | |
| Alternative Flow: | None | | |
| Exception Flow: | **4c. The system fails to verify the user's account information and display message.**  4c1. Users stop the login process.  4c2. Users click on Forgot password. | | |
| Business Rules: | BR-02, BR-16 | | |

##### 2.3.1.3 Sign Out

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-03** | Use Case Name: | **Sign Out** |
| Created By: | HungPN | Date Created: | 2023/05/21 |
| Primary Actor: | User, Manager | Secondary Actor: | None |
| Description: | This use case describes how a user can sign out of their account, which clears any active cookies and logs them out of the system. | | |
| Priority: | High | | |
| Trigger: | The user wants to log out of the Fdorm website and clicks on the log out option. | | |
| Preconditions: | PRE-01: User has logged in into FLearning system. | | |
| Post - Conditions: | POST-1: Fdorm system works stably, without error.  POST-2: System navigates the users back to landing page. | | |
| Basic Flow: | 1. User hovers and clicks the logout option. 2. The user is now no longer determined as any role but Guest. 3. The guest will be navigated back to the Landing page. | | |
| Alternative Flow: | None | | |
| Exception Flow: | None | | |
| Business Rules: | BR-16 | | |

#### 2.3.3 View Common Feature as Manager

##### 2.3.3.1 Manage Student Information

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-07** | Use Case Name: | **Manage Student Information** |
| Created By: | KietNHG | Date Created: | 2023/05/23 |
| Primary Actor: | Manager | Secondary Actor: | None |
| Description: | This use case describes how a manager can manage students’ information. | | |
| Priority: | High | | |
| Trigger: | Manager wants to manage a student’s information. | | |
| Preconditions: | PRE-01: User logged in into the system.  PRE-02: User’s account is a “Manager” account. | | |
| Post - Conditions: | POST-1: FDorm system works stably, without error.  POST-2: System displays menu with options to manage student’s information.  POST-3: User is able to navigate to other pages or features of the application. | | |
| Basic Flow: | 1. Manager chooses the “Manage students” option 2. The system directs the manager to the Manage students page. | | |
| Alternative Flow: | None | | |
| Exception Flow: | None | | |
| Business Rules: | BR-16 | | |

##### 2.3.3.2 View All Student Information

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-08** | Use Case Name: | **View All Student Information** |
| Created By: | KietNHG | Date Created: | 2023/05/23 |
| Primary Actor: | Manager | Secondary Actor: | None |
| Description: | This use case describes how a manager can view all students’ information. | | |
| Priority: | High | | |
| Trigger: | Manager wants to view all students’ information. | | |
| Preconditions: | PRE-01: User logged in into the system.  PRE-02: User’s account is a “Manager” account.  PRE-03: User is in “Manage students” page. | | |
| Post - Conditions: | POST-1: FDorm system works stably, without error.  POST-2: System displays all student’s information.  POST-3: User is able to navigate to other pages or features of the application. | | |
| Basic Flow: | 1. Manager chooses the “Manage students” option 2. The system displays all registered student’s information on the screen. | | |
| Alternative Flow: | None | | |
| Exception Flow: | **2d. The system fails to display students’ information.**  2d1: No students exist in the database. | | |
| Business Rules: | BR-16 | | |

##### 2.3.3.3 Update Student Information

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **UC-09** | Use Case Name: | **Update Student Information** |
| Created By: | KietNHG | Date Created: | 2023/05/23 |
| Primary Actor: | Manager | Secondary Actor: | None |
| Description: | This use case describes how a manager can change or update a student’s information. | | |
| Priority: | High | | |
| Trigger: | Manager wants to update a student’s information. | | |
| Preconditions: | PRE-01: User logged in into the system.  PRE-02: User’s account is a “Manager” account. | | |
| Post - Conditions: | POST-1: FDorm system works stably, without error.  POST-2: Student’s information is updated.  POST-3: User is able to navigate to other pages or features of the application. | | |
| Basic Flow: | 1. Manager clicks the “Update” button on a student’s profile. 2. The system displays the options to update the student’s information. 3. The manager inputs the updated information of the student. 4. The manager clicks “Finish updating”. 5. The updated information of the student is stored. | | |
| Alternative Flow: | 4b. The manager clicks “Cancel”.  5b. The information of the student is unchanged. | | |
| Exception Flow: | **5e. The system fails to update the information.**  5e1. There was an empty field.  5e2. There was an error with the inputted data.  5e3. The manager stopped the updating process. | | |
| Business Rules: | BR-16 | | |