**Renewed Furniture Hub**

**Software Requirements Specification**

Version 1.0



**Group Id: S25PROJECTC2242**

**Supervisor Name : Syed Hassan Ali Shah**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 10/05/2024 | 1.0 | This Software Requirements Specification (SRS) outlines the functional and non-functional requirements for an online used furniture marketplace, designed to connect furniture buyers and sellers through an interactive and user-friendly web platform. The system facilitates the buying and selling of furniture items such as sofas, tables, chairs, cabinets, and more from the comfort of users' homes. It supports user account registration, profile management, furniture listing with images and descriptions, and advanced search functionality for filtering items by price, category, and condition. The platform allows buyers to send purchase requests and contact sellers directly. Based on the VU Process Model and utilizing a hybrid approach of Waterfall and Spiral methodologies, this SRS includes detailed use case definitions, system constraints, and workflow plans to ensure the development of a secure, accessible, and efficient online furniture marketplace. | BC230407113 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project)](file:/C:/Users/student/Desktop/ak.as.docx#scope) Page No**: 04**
2. [Functional Requirements Non Functional requirements](file:/C:/Users/student/Desktop/ak.as.docx#FRNFR) Page No: **05,06,07**
3. [Use Case Diagram](file:/C:/Users/student/Desktop/ak.as.docx#UCD) Page No: **08**
4. [Usage Scenarios](file:/C:/Users/student/Desktop/ak.as.docx#UCS)  Page No: **09**
5. [Adopted Methodology](file:/C:/Users/student/Desktop/ak.as.docx#Adopted)  Page No: **24**
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](file:/C:/Users/student/Desktop/ak.as.docx#Gantt)\_\_\_\_\_\_\_\_\_\_Page No: **29**

CS619 Phase#1 SRS Document

* 1. **Scope Of Project**

The primary objective of this project is to design and develop a dynamic and user-centric web application that serves as a comprehensive platform for the buying and selling of used furniture. This system aims to create a virtual e-furniture marketplace that enables users to carry out furniture transactions seamlessly from the comfort of their homes. Furniture items such as sofas, chairs, tables, wardrobes, beds, bookshelves, and cabinets can be easily listed for sale or browsed for purchase through the platform. The solution will serve both buyers and sellers by offering a streamlined, secure, and responsive environment where furniture listings are well-categorized and easily searchable.

The proposed online furniture store will allow customers to view available furniture, filter search results based on criteria such as price range, condition, category, and location, and connect with sellers through a secure communication system. Users will be able to create personal accounts, manage their profiles, and upload detailed furniture listings with images, descriptions, prices, and condition reports. Buyers can send purchase requests to sellers, and upon approval, proceed with the transaction. The system will feature a responsive and user-friendly interface, optimized for accessibility and ease of navigation on both desktop and mobile devices.

Furthermore, the application will support three key user roles: Guests, Registered Users, and Administrators. Guest users can browse and search listings, while registered users gain access to core functionalities such as uploading items, editing listings, placing orders, rating sellers, and filing complaints. The Administrator will oversee system operations, manage users and listings, generate reports, and ensure platform integrity. This project will integrate secure login, profile management, order tracking, and multiple payment options. It will also ensure data protection and facilitate dispute resolution through an in-system complaint mechanism, thereby building a trusted and efficient ecosystem for used furniture trading. This platform aims to enhance convenience, trust, and accessibility in the second-hand furniture market.

It ensures a structured and scalable solution that meets user expectations and administrative control. The system will be built using secure, modern web technologies to ensure long-term usability and performance.

* 1. **Functional Requirements:-**

**1: User Module:-**

**a) User Registration and Sign In:** There will be a proper signup interface for unregistered users to register on the website. A registered user will be able to login to the website by entering the correct credentials in the sign in interface.

**b) View Furniture:** All available Furniture will be available on the website with a proper interface. Any registered user or guest can view information of available furniture and can view the complete description, images, demanded price, furniture owner’s contact & location and other necessary details. The guest can only view and search the available information and cannot make any upload or any purchase request without any registration on the website.

**c) Search Furniture:** Any user registered or unregistered can search for the available furniture on the website. The search can be done using advanced search filters like price range, category, condition, and location etc. If user requirement meets, then system will show results in proper format.

**d) Upload Furniture**: A registered user will need to upload the furniture details if he/she wants to sell furniture. After login to the website, the user will upload all the necessary details i.e. description, price, images, condition etc. of the furniture on the website.

**e) Update details of uploaded Furniture**: If a registered user has uploaded furniture on the website, he/she will be able to edit and update the price or any other details of that furniture. The registered user will also be able remove the furniture from the website.

**f) Update profile:** Upon successful login to the website by a registered user, he/she will be able to update any of his/her profile information and can update his/her account password.

**g) User Review and Feedback:** Any registered user will be able to submit his/her review about the furniture purchased and can give feedback about it to its previous owner.

**h) User Complain:** In case of any fraud or serious issue, any registered user will be able to submit complain about the concerned to the admin. After verification of the matter, the admin can either give warning to the concerned or can block the concerned account on the website.

**i) Give Rating:** Customers can give a rating to the seller concerned after the completion of purchase request according to his/her satisfaction.

**j) Buy Furniture:** A user must log in successfully to the website to place an order for buying furniture. A registered user needs to fill all the order details for buying the furniture of his/her choice. After filling the details, user will be moved to the payment section.

**2: Admin Module: -**

**a) Login:** Using valid login credentials, admin need to login into the system to access the system.

**b) Manage Furniture:** Admin can upload the furniture’s information, view all the added furniture online with their details by the registered users. Admin can block and unblock any furniture for displaying on the website.

**c) Admin Dashboard:** Admin can view the detailed summary of everything such as count of registered users, registered user details, total sold furniture along with buyer and seller details, total available furniture etc.

**d) Manage Users:** All the registered user details will be displayed to the admin. Admin will accept / reject the user registration requests and can block or unblock any user for uploading or buying furniture on the website.

**e) Report Generation:** The admin will be able to generate a complete report of total sales of furniture purchased on daily, weekly, and monthly basis.

**3: Payment Module:**-

Upon successful completion of any order, customers can either pay the amount to the concerned delivery person on the spot or can transfer the amount to the concerned furniture seller’s account number available on the website. The responsibility of the delivery will be on the seller and the responsibility of verifying and ensuring quality of the selected furniture will be on the customer respectively.

* 1. **Non Functional Requirements:-**

**1. Security:**

This includes requirements related to the protection of the system and its data from unauthorized access, as well as the ability to detect and recover from security breaches.

**2. Usability:**

This includes requirements related to the ease of use and understandability of the system for the end-users.

**3. Reliability:**

This includes requirements related to the system’s ability to function correctly and consistently under normal and abnormal conditions.

**4. Maintainability:**

This includes requirements related to the ease of maintaining the system, including testing, debugging, and modifying the system.

**5. Portability:**

This includes requirements related to the ability of the system to be easily transferred to different hardware or software environments.

**6. Compliance:**

This includes requirements related to adherence to laws, regulations, industry standards, or company policies.

**7.Performance:**

The system must respond to user inputs within 2 seconds and handle a minimum of 100 concurrent users.

**8. Accessibility:**

The system must comply with accessibility standards for users with disabilities. The system must comply with accessibility standards for users with disabilities.

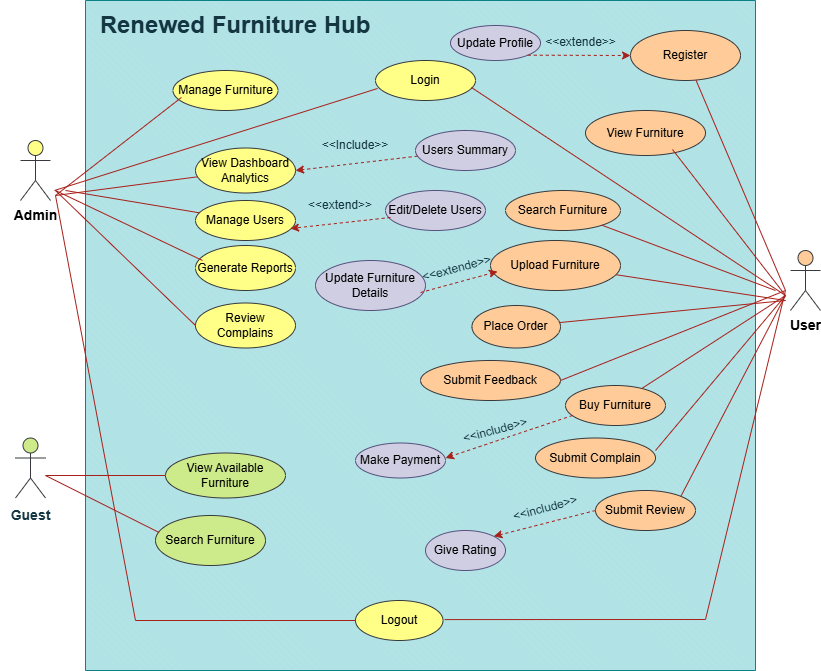
**9. Scalability:**

The system must be able to handle a 20% increase in users and transactions without a significant decrease in performance.

**10. Audit ability:**

The system must provide a clear audit trail of all transactions and updates.

1. **Use Case Diagram(s):-**



1. **Usage Scenario:-**

# Use Case Id: 1(View Available Furniture)

|  |  |
| --- | --- |
| FR-ID | FR-001 |
| Actors | Guest, Registered User, Admin |
| Description | This use case allows a Guest to view furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Guest accesses the system. 2. Navigates to the section to view furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 2(search furniture)

|  |  |
| --- | --- |
| FR-ID | FR-002 |
| Actors | Guest, Registered User, Admin |
| Description | This use case allows a Guest to search furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Guest accesses the system. 2. Navigates to the section to search furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 3(register)

|  |  |
| --- | --- |
| FR-ID | FR-003 |
| Actors | Guest |
| Description | This use case allows a Guest to register. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Guest accesses the system. 2. Navigates to the section to register. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 4(login)

|  |  |
| --- | --- |
| FR-ID | FR-004 |
| Actors | Registered User |
| Description | This use case allows a Guest to login. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Guest accesses the system. 2. Navigates to the section to login. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 5(upload furniture)

|  |  |
| --- | --- |
| FR-ID | FR-008 |
| Actors | Registered User |
| Description | This use case allows a Registered User to upload furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to upload furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 6(edit uploaded furniture)

|  |  |
| --- | --- |
| FR-ID | FR-009 |
| Actors | Registered User |
| Description | This use case allows a Registered User to edit uploaded furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to edit uploaded furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 07(delete uploaded furniture)

|  |  |
| --- | --- |
| FR-ID | FR-010 |
| Actors | Registered User |
| Description | This use case allows a Registered User to delete uploaded furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to delete uploaded furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 08(update profile)

|  |  |
| --- | --- |
| FR-ID | FR-011 |
| Actors | Registered User |
| Description | This use case allows a Registered User to update profile. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to update profile. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 09(change password)

|  |  |
| --- | --- |
| FR-ID | FR-012 |
| Actors | Registered User |
| Description | This use case allows a Registered User to change password. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to change password. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 10(submit review)

|  |  |
| --- | --- |
| FR-ID | FR-013 |
| Actors | Registered User |
| Description | This use case allows a Registered User to submit review. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to submit review. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 11(submit feedback)

|  |  |
| --- | --- |
| FR-ID | FR-014 |
| Actors | Registered User |
| Description | This use case allows a Registered User to submit feedback. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to submit feedback. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 12(submit complaint)

|  |  |
| --- | --- |
| FR-ID | FR-015 |
| Actors | Registered User |
| Description | This use case allows a Registered User to submit complaint. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to submit complaint. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 13(give rating)

|  |  |
| --- | --- |
| FR-ID | FR-016 |
| Actors | Registered User |
| Description | This use case allows a Registered User to give rating. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to give rating. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 13(buy furniture)

|  |  |
| --- | --- |
| FR-ID | FR-017 |
| Actors | Registered User |
| Description | This use case allows a Registered User to buy furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to buy furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 14(make payment)

|  |  |
| --- | --- |
| FR-ID | FR-018 |
| Actors | Registered User |
| Description | This use case allows a Registered User to make payment. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Registered User accesses the system. 2. Navigates to the section to make payment. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 15(view admin dashboard)

|  |  |
| --- | --- |
| FR-ID | FR-020 |
| Actors | Admin |
| Description | This use case allows a Admin to view admin dashboard. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Admin accesses the system. 2. Navigates to the section to view admin dashboard. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 16(manage furniture)

|  |  |
| --- | --- |
| FR-ID | FR-021 |
| Actors | Admin |
| Description | This use case allows a Admin to manage furniture. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Admin accesses the system. 2. Navigates to the section to manage furniture. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 17(manage users.)

|  |  |
| --- | --- |
| FR-ID | FR-022 |
| Actors | Admin |
| Description | This use case allows a Admin to manage users. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Admin accesses the system. 2. Navigates to the section to manage users. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

# Use Case Id: 18(generate reports)

|  |  |
| --- | --- |
| FR-ID | FR-023 |
| Actors | Admin |
| Description | This use case allows a Admin to generate reports. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Admin accesses the system. 2. Navigates to the section to generate reports. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

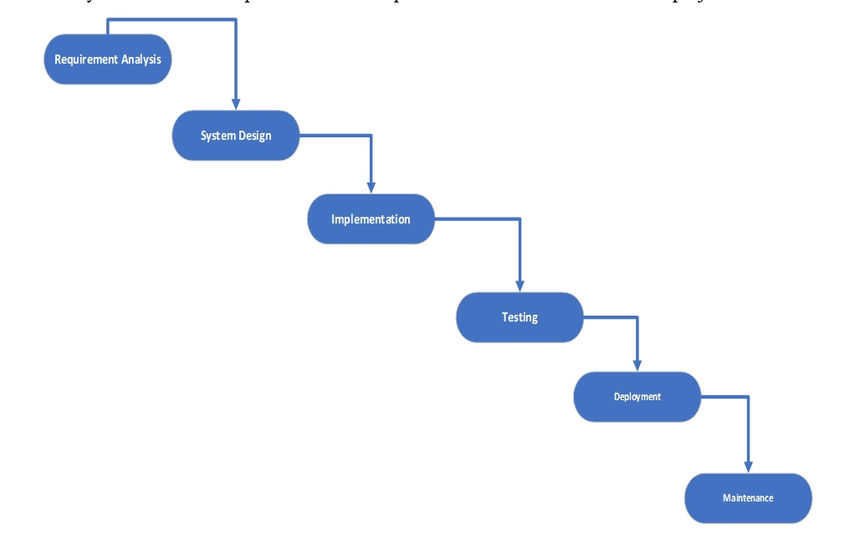
# Use Case Id: 19(review complaints)

|  |  |
| --- | --- |
| FR-ID | FR-024 |
| Actors | Admin |
| Description | This use case allows a Admin to review complaints. It outlines the steps involved in performing this action and the outcomes expected from the system. |
| Alternative Path | - User cancels the action before completion. - System encounters temporary unavailability. - User session expires before action completion. |
| Pre Conditions | User must be on the appropriate page. User must have necessary permissions (if required). System must be available. |
| Action | 1. Admin accesses the system. 2. Navigates to the section to review complaints. 3. Fills necessary information or interacts with the feature. 4. Clicks the confirm/submit button. 5. System processes the request and responds accordingly. |
| Post Conditions | System records the action successfully. Confirmation message is shown to the user. Database updates reflect the change. |
| Author | BC230407113 |
| Exception | System error or invalid input. |
| Modification History | Initial version v 1.1 |

1. **Adopted Methodologies:-**

**1.Waterfall Model:**

The Waterfall model is a linear, sequential software development process where each phase is completed before moving on to the next one, with no overlap or iteration. It follows a rigid structure, requiring fixed and well-defined requirements, and is suitable for projects with complexity and risk.



The six stages above are as follows:

**I. Requirement Analysis and Definition:**

What- The systems services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

**II. System and Software Design:**

How – The system design process partitions the requirements to either hardware of software systems. It establishes and overall system architecture. Software design involves fundamental system abstractions and their relationships

**III.Implementation and Unit Testing: -**

How – During this stage the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specifications.

**IV. Integration and system testing:**

The individual program unit or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

**V.Deployment:-**

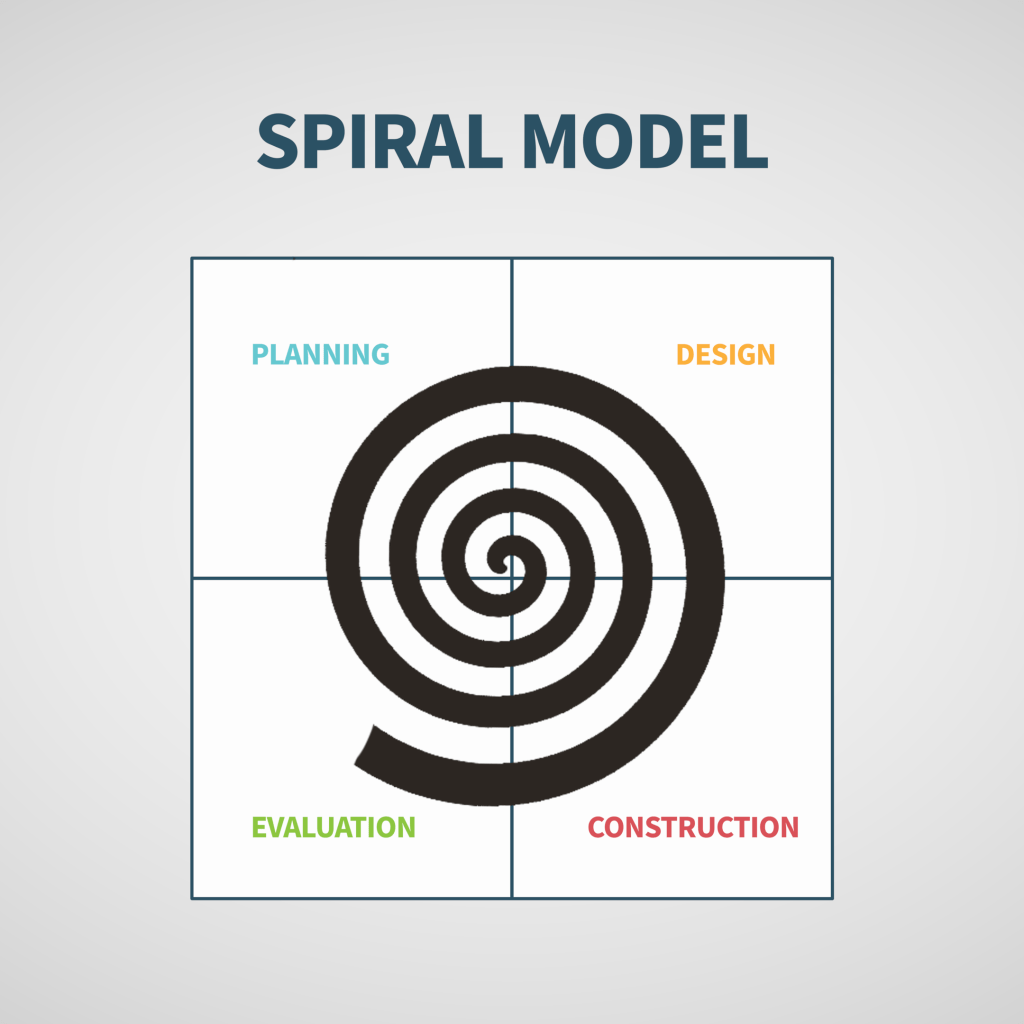
In the Deployment phase, the focus is on ensuring a smooth transition of the software from development to production, and making it available to end-users. This phase involves final testing, packaging, distribution, installation, configuration, user training, and documentation. Once the software is successfully deployed, it is handed over to the maintenance team for ongoing support and maintenance.

**VI. Operation and Maintenance:**

Normally this is the longest phase of the software life cycle. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life-cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered

**2.Spiral Modal:**

The Spiral Model is a [**Software Development Life Cycle (SDLC)**](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/) model that provides a systematic and iterative approach to software development. In its diagrammatic representation, looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a **Phase of the**software development.



**I. Objectives determination and identify alternative solutions:**

In this first quadrant, the main goal is to gather as much information as possible from the customers. This includes understanding their needs, expectations, and any constraints they may have. Once the objectives are clearly defined, the team starts brainstorming different solutions that could meet these objectives. These solutions are then evaluated based on their feasibility, cost, time, and alignment with the customer’s requirements. The best solutions are shortlisted for further analysis in the next quadrant.

**II. Identify and resolve Risks:**

The second quadrant is all about risk management. The team evaluates the shortlisted solutions from the first quadrant and identifies potential risks associated with each one. These risks could be technical, financial, operational, or even related to the market or customer preferences. Once the risks are identified, the team develops strategies to mitigate them. This could involve modifying the solution, developing contingency plans, or even discarding the solution if the risks are too high. The end result of this quadrant is a prototype of the chosen solution, which is then tested and refined in the next quadrant.

Develop the next version of the Product:

**III. The third quadrant is where the actual development happens.**

Based on the prototype from the second quadrant, the team starts building the features of the product. This involves coding, testing, and debugging. The team also verifies that the product meets the objectives defined in the first quadrant and that the risks identified in the second quadrant have been effectively mitigated. By the end of this quadrant, a new version of the product is ready for review.

**IV. Review and plan for the next Phase:**

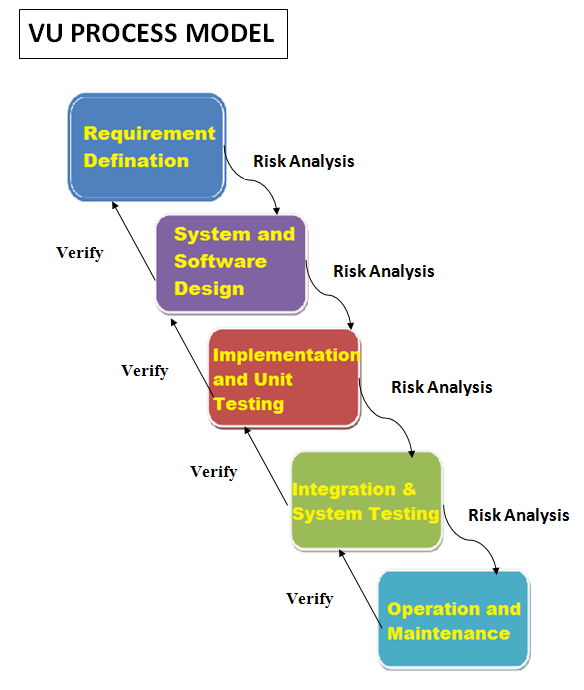
The fourth and final quadrant involves reviewing the product with the customers. They evaluate the product to ensure it meets their needs and provides value. The team collects feedback and uses it to improve the product in the next phase. This quadrant also involves planning for the next phase of the Spiral Model, which could involve scaling the product, adding new features, or even starting a new project.

**Choosen Methodology:-**

I will choose vu process Model for my project which is the combination of waterfall and spiral model.

**3.VU Process Model:**

A structured methodology for process management and improvement, comprising four sequential stages.



**I. Vision (Definition and Goal-Setting)**

Define the desired future state of the process (Vision Statement)

Establish clear goals and objectives (SMART criteria)

Identify key performance indicators (KPIs) to measure success

Define the scope and boundaries of the process

Identify stakeholders and their requirements

**II. Understanding (Current State Assessment)**

Analyze the current state of the process (As-Is process mapping)

Identify strengths, weaknesses, opportunities, and threats (SWOT analysis)

Gather data and feedback from stakeholders (surveys, interviews, etc.)

Identify pain points and areas for improvement

Document the current process (process mapping, procedures, etc.)

**III. Upgrading (Process Improvement and Implementation)**

Design and implement process improvements (To-Be process mapping)

Develop new procedures and standards

Train and support stakeholders

Develop a change management plan

Implement the new process (transition from As-Is to To-Be)

**IV. Validation (Monitoring and Evaluation)**

Monitor and evaluate the improved process

Measure KPIs and adjust as needed

Continuously review and refine the process to ensure sustainability

Celebrate successes and recognize improvements

Reasons For Choosing:-

**I.Structured Approach:**

The VU Process Model emphasizes a structured approach to software development. It ensures that you follow a systematic sequence of steps, from requirements gathering to testing and deployment.

**II. Early Specifications and Design:**

Before writing a single line of code, the VU Process Model focuses on creating detailed specifications and design documentation. This minimizes wastage of effort and time and reduces the risk of schedule slippage or unmet customer expectations1.

**III. Risk Management:**

Like the Spiral Model, the VU Process Model incorporates risk management. It allows you to identify and address potential risks early in the project. By iteratively assessing risks, you can make informed decisions and adjust your approach as needed.

**IV. Adaptability:**

The VU Process Model allows for flexibility. You can adapt it to suit your project’s specific needs. For instance, if certain requirements change during development, you can adjust the design and specifications accordingly.

**V. Clear Documentation:**

The model encourages comprehensive documentation at each stage. This documentation serves as a valuable reference for team members, stakeholders, and future maintenance.

**VI. Gantt Chart Planning:**

The VU Process Model recommends creating a Gantt chart to visualize project activities and milestones. This helps manage project timelines effectively

1. **Work Plan:-**

