

**SOFTWARE REQUIREMENT SPECIFICATION**

**License Marketplace System**

– Hanoi, Jan 2024 –

**Table of Contents**

[I. Record of Changes 3](#_dn2gwrbhr7n3)

[II. Software Requirement Specification 4](#_vbd22grmccnj)

[1. Overall Requirements 4](#_q7havltpywnx)

[1.1 Context Diagram 4](#_w7dtfaqt80ds)

[1.2 Main Business Processes 5](#_p1bdl9w2kxm5)

[1.3 User Requirements 5](#_jcyky67yn9b8)

[1.4 System Functionalities 7](#_nz5zf4i6kb6f)

[1.5 Entity Relationship Diagram 7](#_i931jlmxycq)

[2. Use Case Specifications 8](#_1y8fyihinmee)

[2.1 <<Feature Name1>> 8](#_uz8s0iu40n43)

[2.2 Xyz Feature 10](#_xn06jmhqxh2y)

[3. Functional Requirements 11](#_tifntvf5s2mw)

[3.1 Feature Name1 11](#_oduebmq3xunt)

[3.2 User Authentication 11](#_toi4xaxct1dy)

[3.3 System Administration 12](#_bkmr78vku96h)

[4. Non-Functional Requirements 13](#_soe00xcdlxpg)

[3.1 External Interfaces 13](#_pzb43pv2bh02)

[3.2 Quality Attributes 13](#_c6d8eqt0l5yr)

[5. Requirement Appendix 14](#_b3c89zi1d5ds)

[5.1 Business Rules 14](#_yho193gwq5ab)

[5.2 System Messages 14](#_2gr8zrl3rpod)

[5.3 Other Requirements… 15](#_3163hous5ore)

# I. Record of Changes

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

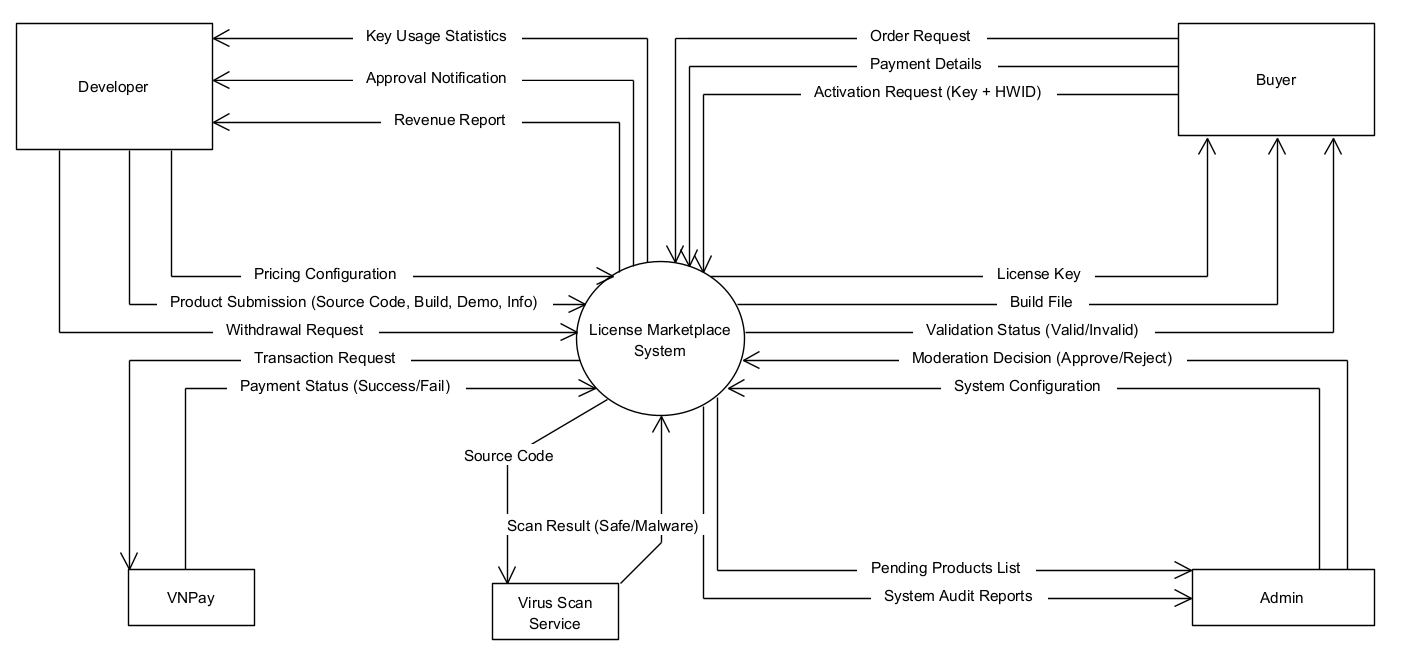
\*A - Added M - Modified D - Deleted

# II. Software Requirement Specification

## 1. Overall Requirements

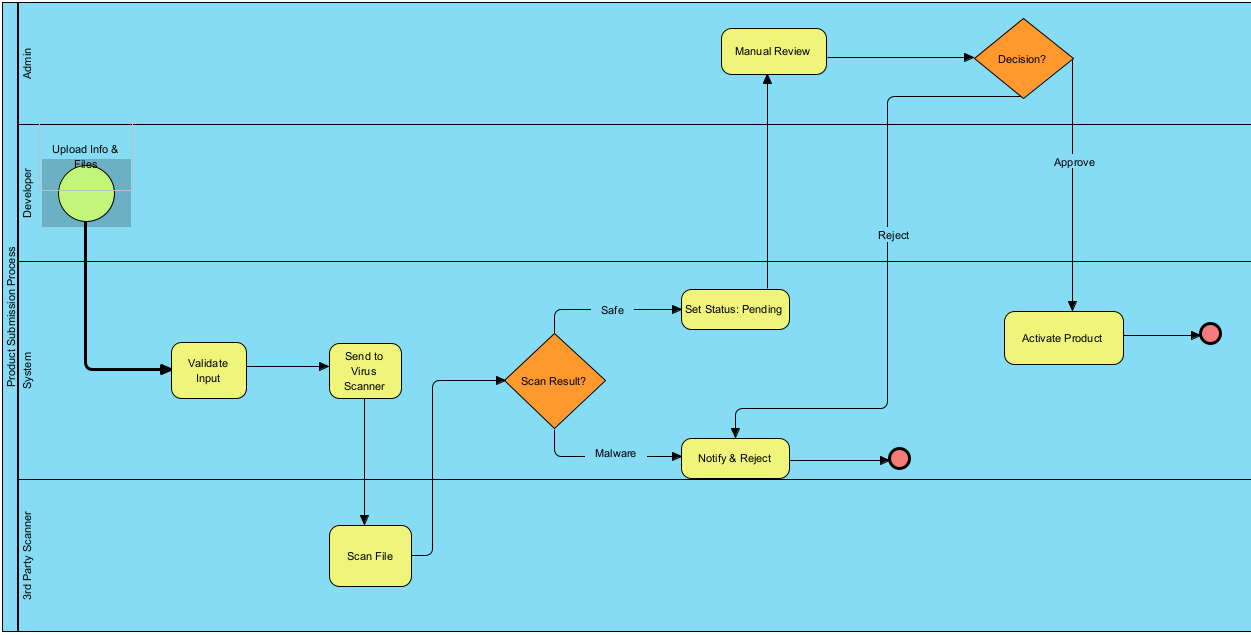
### 1.1 Context Diagram

**License Marketplace System** is a new web-based platform that streamlines the distribution, sale, and copyright protection of software products, replacing the fragmented and insecure manual processes currently used by independent developers. The context diagram below illustrates the external entities and system interfaces for **Release 1.0**, focusing on secure code submission, automated virus scanning via **3rd Party Security Services**, and real-time license key validation. The system is expected to evolve over several releases, ultimately connecting to recurring billing services for subscription-based software and providing advanced analytics APIs for developers to track user engagement.

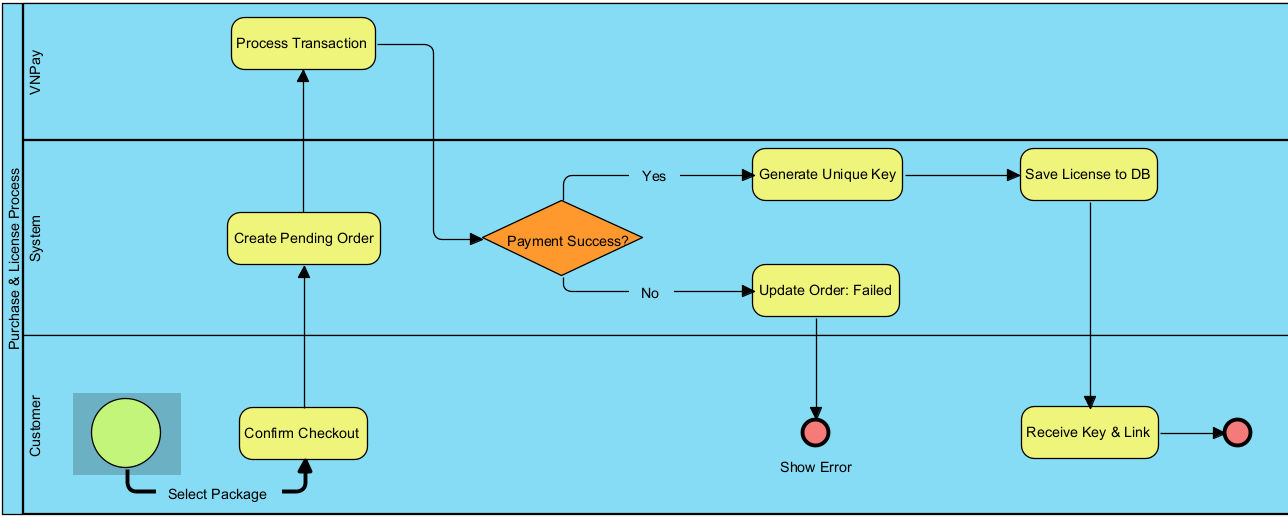


### 1.2 Main Business Processes

#### 1.2.1 Product Submission Processing



#### 1.2.2 Purchase & License Generation Process



### 1.3 User Requirements

#### 1.3.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Buyer (End User) | An individual who accesses the system to browse, purchase, and use software licenses. |
| 2 | Developer | An individual or entity who creates software and uses the platform to distribute their products. |
| 3 | Admin | A super-user responsible for managing the platform's operation, content moderation, and system configuration. |
| 4 | Payment Gateway | An external system that processes online financial transactions: VNPay |
| 5 | 3rd Party Security Service | An external system used for automated malware and virus scanning: |
| 6 | Client Software | The software installed on the Buyer's machine that interacts with the system for license validation. |

#### 1.3.2 Use Cases (UC)

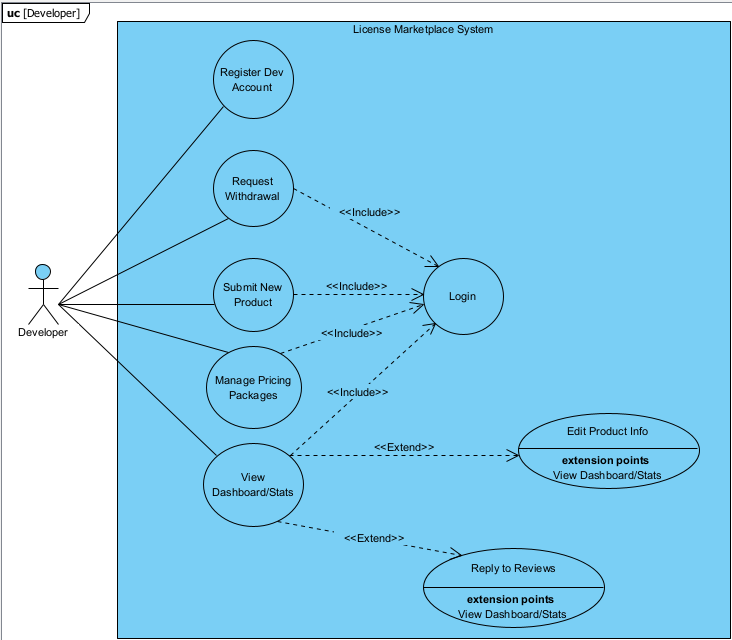
|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID** | **Use Case Name** | **Primary Actor** | **Description (Outcome of Value)** |
| **UC-01** | Register Account | Buyer, Developer | The actor creates a new account on the system to access restricted features (buying or selling). |
| **UC-02** | Log In | All Actors | The actor authenticates their identity to access their specific dashboard and permissions. |
| **UC-03** | Search Products | Buyer (Guest) | The actor searches for software by name, category, or filter to find a product of interest. |
| **UC-04** | View Product Details | Buyer (Guest) | The actor views detailed information (video, description, pricing) of a specific product to make a purchase decision. |
| **UC-05** | Purchase License | Buyer | The actor selects a license package (e.g., monthly, lifetime) and completes the payment process to obtain the right to use the software. |
| **UC-06** | Download Product | Buyer | The actor retrieves the installation file (build) of the purchased software from the system. |
| **UC-07** | Submit New Product | Developer | The actor uploads product details, source code, and build files for administrator review. |
| **UC-08** | Manage Pricing Packages | Developer | The actor defines or updates the cost and duration (e.g., 1 month, 1 year) for their software licenses. |
| **UC-09** | Request Withdrawal | Developer | The actor requests a payout of their accumulated earnings from the system wallet to their bank account. |
| **UC-10** | Approve Product | Administrator | The actor reviews a submitted product (and its virus scan results) and publishes it to the marketplace. |
| **UC-11** | Reject Product | Administrator | The actor declines a submitted product due to policy violations or malware detection, providing a reason to the Developer. |
| **UC-12** | Process Withdrawal | Administrator | The actor reviews a financial withdrawal request and transfers funds to the Developer. |
| **UC-13** | Ban User | Administrator | The actor locks a specific user account to prevent access due to suspicious activity or policy violations. |
| **UC-14** | Activate License | Client App (User) | The Client Application (acting on behalf of the user) validates the input key and hardware ID with the server to unlock the software. |

#### 1.3.2 Use Case Diagrams

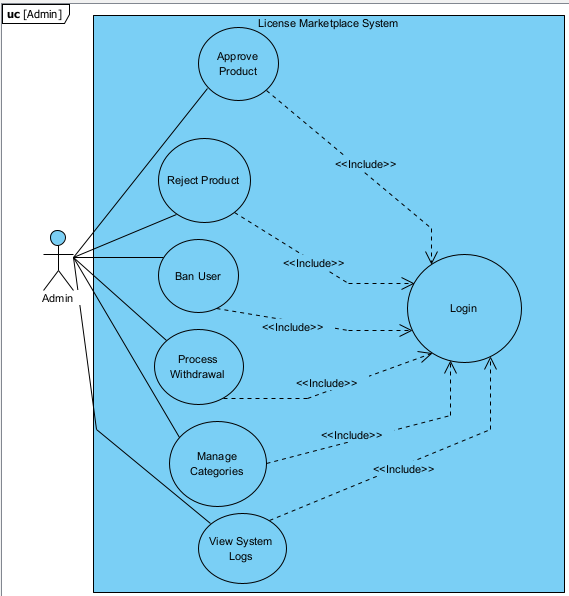
##### 1.3.2.1 UCs for Buyer

**

##### 1.3.2.2 UCs for Developer

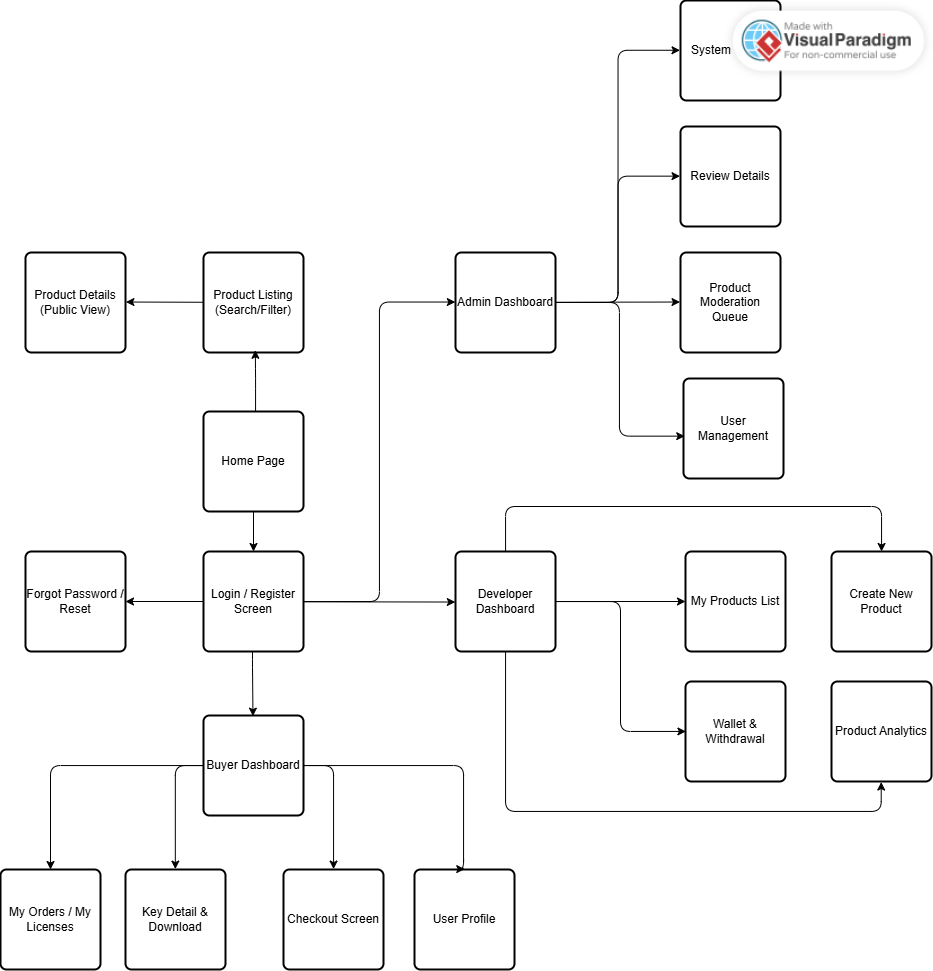
**

##### 1.3.2.3 UCs for Admin



### 1.4 System Functionalities

#### 1.4.1 Screens Flow



#### 1.4.2 Screen Authorization

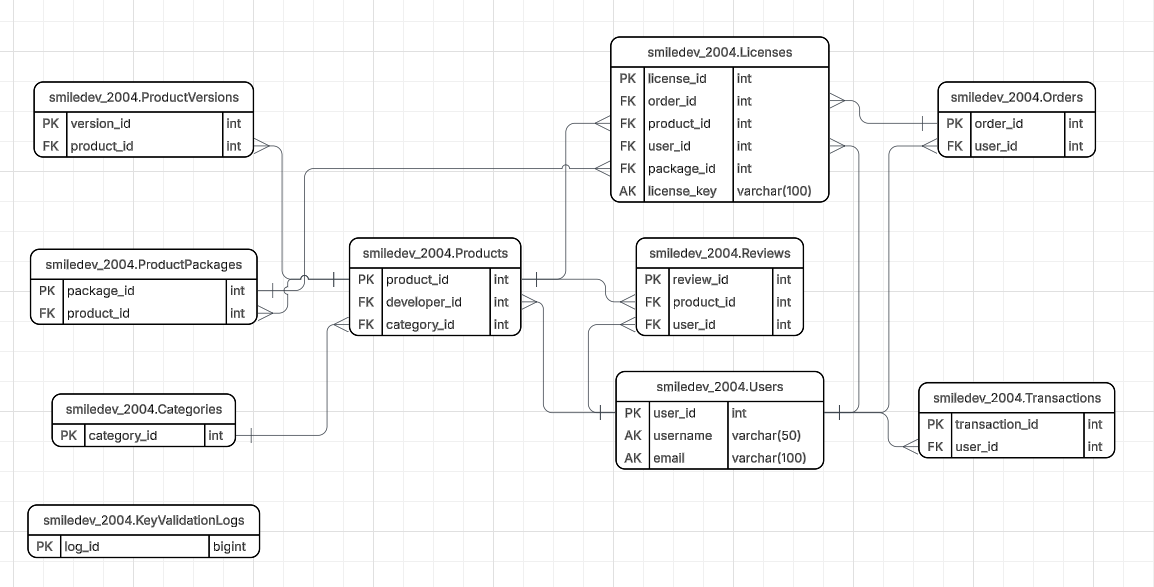
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen / Feature** | **Guest** | **Buyer** | **Developer** | **Administrator** |
| **Public Screens** | | | | |
| Home Page | X | X | X | X |
| Login / Register | X |  |  |  |
| Product Listing (Search/Filter) | X | X | X | X |
| Product Details (Public View) | X | X | X | X |
| Forgot Password | X |  |  |  |
| **Buyer Screens** | | | | |
| Buyer Dashboard |  | X |  |  |
| My Licenses / My Orders |  | X |  |  |
| Key Detail & Download |  | X |  |  |
| Checkout / Payment |  | X |  |  |
| User Profile (Buyer View) |  | X |  |  |
| **Developer Screens** |  |  |  |  |
| Developer Dashboard |  |  | X |  |
| Create New Product |  |  | X |  |
| Manage Products List |  |  | X |  |
| Wallet & Withdrawal |  |  | X |  |
| Product Analytics |  |  | X |  |
| **Admin Screens** | | | | |
| Admin Dashboard |  |  |  | X |
| Product Moderation Queue |  |  |  | X |
| Review Product Details |  |  |  | X |
| User Management (Ban/Unban) |  |  |  | X |
| System Configuration |  |  |  | X |
| Transaction History (All) |  |  |  | X |

#### 1.4.3 Non-UI Functions

*[Provide the descriptions for the non-screen system functions, i.e batch/cron job, service, API, etc.]*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | <<Feature Name>> | <<Function Name1>> | <<Function Name1 Description>> |
| 2 | … |  |  |

### 1.5 Entity Relationship Diagram



**Entities Description**

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | Users | Stores account information for all system roles (Admin, Developer, Buyer). It manages authentication credentials, profile details, and the digital wallet balance. |
| 2 | Products | Contains core information about the software listed by Developers (e.g., name, description, demo video). It serves as the central entity linking Developers to Buyers. |
| 3 | ProductVersions | Manages the versioning history of software (e.g., v1.0, v1.1). It stores file paths for both Source Code (for Admin security checks) and Build Files (for User downloads). |
| 4 | ProductPackages | Defines the pricing models and duration limits for each product (e.g., "1-Month Subscription", "Lifetime License"). |
| 5 | Categories | Manages software classifications (e.g., System Tools, Game Assets, UI Kits) to help users categorize and filter products effectively. |
| 6 | Orders | Records purchase transactions made by Buyers. It tracks the total amount, payment method, and the current payment status (Pending/Completed). |
| 7 | Licenses | (Core Entity) Stores the unique license keys generated upon successful orders. It manages critical DRM data including expiration dates, activation status, and the bound Hardware ID. |
| 8 | Transactions | Logs the history of all financial movements within the system, including deposits, purchase deductions, developer revenue credits, and withdrawals. |
| 9 | Reviews | Stores feedback and ratings (1-5 stars) provided by Buyers for purchased products, helping to establish product quality and trust. |
| 10 | ActivityLogs | An audit trail that records critical actions performed by Administrators and Developers (e.g., Approving a product, Changing configuration) for security and accountability. |
| 11 | KeyValidationLogs | Logs the history of API requests from client software when validating licenses. Used for monitoring usage frequency and detecting potential piracy attempts. |

## 2. Use Case Specifications

*[Provide specifications for the use cases (UCs) those are covered in the system. The UCs are grouped by the system features and even sub features.* ***You just need to provide UC specifications for complex UCs involving in the main workflows (business processes)****. Other UCs (i.e CRUD or data-viewing UCs) are simple, and you just need to refer the descriptions in the Functional Requirement (part 3) below)]*

### 2.1 <<Feature Name1>>

#### 2.1.1 UC Name1

|  |  |  |  |
| --- | --- | --- | --- |
| Primary Actors |  | Secondary Actors |  |
| Description |  | | |
| Preconditions |  | | |
| Postconditions |  | | |
| Normal Sequence/Flow |  | | |
| Alternative Sequences/Flows |  | | |

***Primary and Secondary Actors***

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Name the primary actor that will be initiating this UC and any other secondary actors who will participate in completing execution of the UC.

***Description***

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case. The description can be in the form of a user story (As a **<type of user>**, I want **<some goal>** so that **<some reasons>**)

***Preconditions***

List any activities that must take place, or any conditions that must be true, before the use case can be started.

***Postconditions***

Describe the state of the system at the successful conclusion of the use case execution.

***Normal Flow***

Provide a description of the user actions and corresponding system responses that will take place during execution of the use case under normal, expected conditions.

***Alternative Flows***

Describe below two information if any:

* Other successful usage scenarios that can take place within this use case. State the alternative flow, and describe any differences in the sequence of steps that take place.
* Any anticipated error conditions that could occur during execution of the use case and how the system is to respond to those conditions.

#### 2.1.2 Login System

|  |  |  |  |
| --- | --- | --- | --- |
| Primary Actors | Customer | Secondary Actors | None |
| Description | As a user, I want to be able to log into the system so that I can use the system’s authenticated features and access my personalized account. | | |
| Preconditions | User account has been created & authorized | | |
| Postconditions | * User logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Sequence/Flow | ***Login System***   1. User clicks Login button from the page header or accesses an authenticated feature (from a link or type the page URL directly into the address bar) 2. System show the User Login screen 3. User types in the login details (email, password) 4. User clicks the Login button 5. System validates the login details (BR-01, BR-02) 6. System allows user to access 7. System tracks user’s success login to the Activity Log 8. System directs user to the Home Page (or the previous calling page if any) | | |
| Alternative Sequences/Flows | ***Step 2.1\_Google Login***   1. User clicks Google Login button to login system using Google account 2. System redirects the user to the Google’s Login screen 3. User types in the Google account details and chooses to login 4. Google validates user’s login information successfully and redirect him/her back to the system 5. Return to step 5 of normal flow.   ***Step 4\_System can’t authenticate the user***  User can’t be authenticated & get relevant error message in one of below cases   1. He/she leaves the Email and/or Password field blank (MSG10) 2. Input Email or Password are incorrect (MSG09) 3. Input Email/Password are correct but email has not been verified (MSG11) 4. The user account is blocked / inactive (MSG12)   If user inputs wrong logging-in details 5 times continuously, system will lock his/her account in 30 minutes (with relevant warning message - MSG13) | | |

#### 2.1.3 UC Name2

…

### 2.2 Xyz Feature

…

## 3. Functional Requirements

*[Provide descriptions about the system’s functions/screens. The functions/screens are grouped by the system features, and even sub-features if needed. For the screens, you need to provide the screen layouts (mock-up screens) and relevant specifications if needed]*

### 3.1 Feature Name1

#### 3.1.1 SubFeature Name1.1

##### 3.1.1.1 Screen/Function Name1

*[Content #1: UI layout (Mockup screen prototype)]*

*[Content #2: brief descriptions of the screen/function, mapped to the relevant use cases]*

*[Content #3: provide further descriptions for the screen’s components/fields using table format below]*

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| Field Name1 | Field description: data type min/max length or value, initial data, etc. |
| Field Name2 | … |
| ***Field Group-Name1*** | |
| Field Name3 | … |
| Field Name4 | … |
| ***Field Group-Name2*** | |
| … | … |

##### 3.1.1.2 Screen/Function Name2

…

#### 3.1.2 SubFeature Name1.2

…

### 3.2 User Authentication

#### 3.2.1 User Register

…

#### 3.2.2 User Login

…

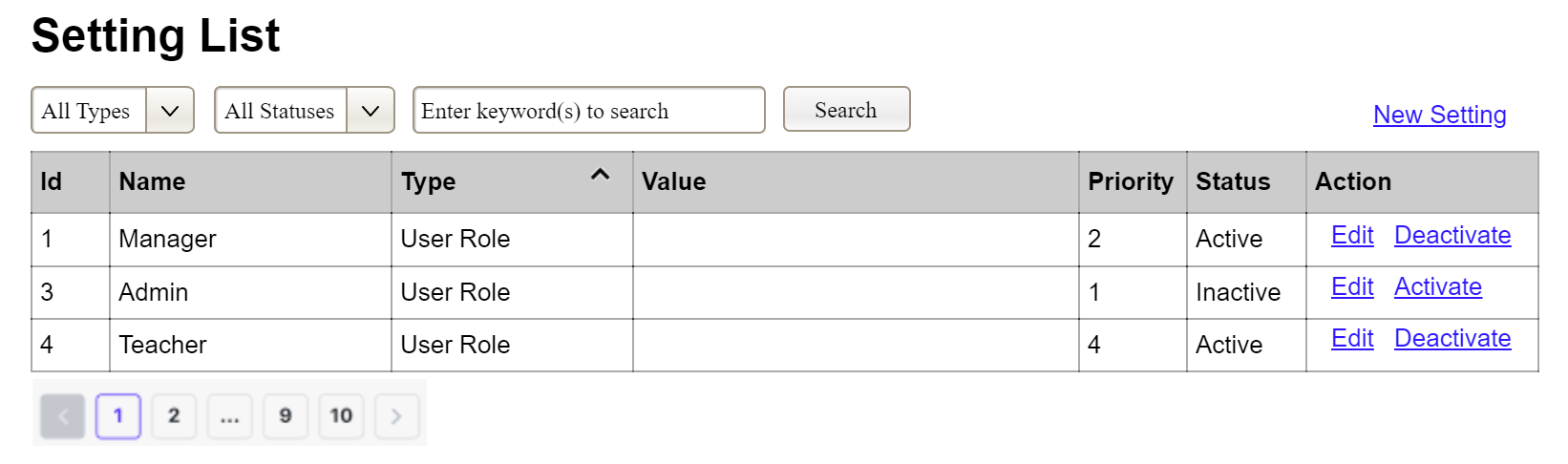
#### 3.2.3 Password Reset

…

### 3.3 System Administration

#### 3.3.1 Master Data

##### 3.3.1.1 Setting List



(1)

(2)

(3)

This screen allows the Administrator to:

* View Setting List: view list of current master data.
* Filter Setting List: filter master data by data types, statuses
* Search Settings: enter keyword(s) to search master data by their names or values
* Sort Setting List: sort master data list (ascending, descending) by clicking column headers

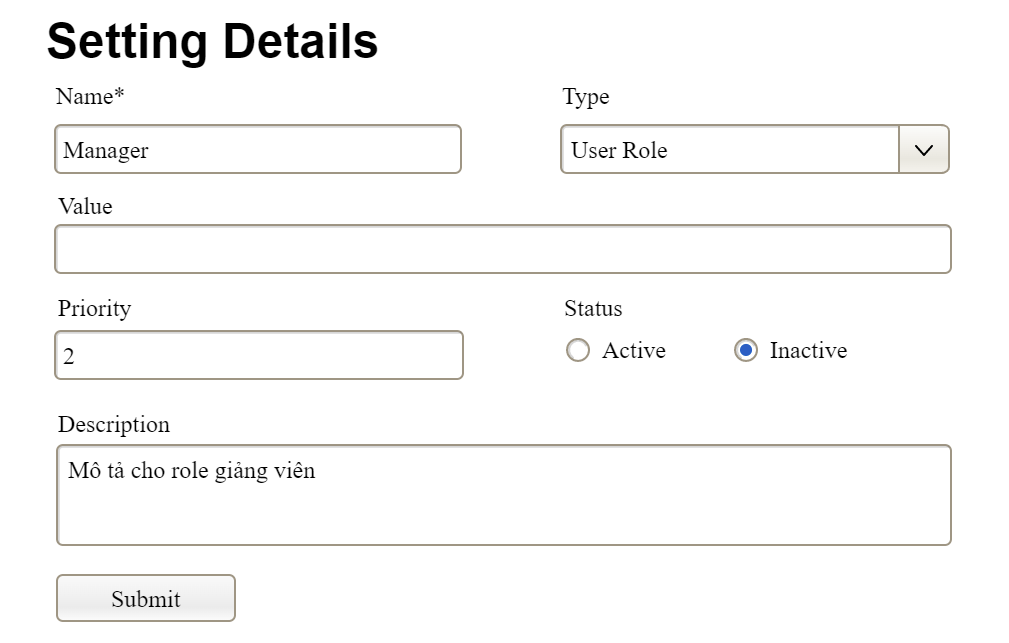
On the screen, s/he can also

* Activate/Deactivate Setting: change status of a specific inactive/active master data
* Choose to go to the Setting Details screens for adding new or updating an existing master data by clicking the New Setting or Edit link.

**Field Description**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| (1) | Initial values: all the active setting names with null or blank type  Hover the mouse to show the field name: “Setting Type” |
| (2) | Initial values: All Statuses, Active, Inactive (default value “All Status”)  Hover the mouse to show the field name: “Setting Status” |
| (3) | The change-status action is Activate or Deactivate depending on the current status of the relevant setting (Inactive or Active, respectively). |

##### 3.3.1.2 Setting Details



This screen allows the Administrator to:

* Add New Setting: add new master data.
* Update Setting Details: update details of a specific master data

**Field Description**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| Name | Data type: non-digit string, max length of 20 characters |
| Type | Initial data values: all active setting names (with null or blank type) |
| Value | Data type: any string, max length of 100 characters |
| Priority | Data type: a positive integer |
| Description | Data type: any string, max length of 200 characters |

#### 3.3.2 User Management

##### 3.3.2.1 User List

…

##### 3.3.2.2 User Details

…

## 4. Non-Functional Requirements

### 3.1 External Interfaces

*[This section provides information to ensure that the system will communicate properly with users and with external hardware or software/system elements.]*

### 3.2 Quality Attributes

*[List all the required system characteristics (quality attributes) specification. Some of the possible attributes are provided with the guide/descriptions are mentioned here]*

#### 3.2.1 Usability

*[This section includes all those requirements that affect usability. For example, specify the required training time for a normal user and a power user to become productive at particular operations specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]*

#### 3.2.2 Performance

*[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.*

*Response time for a transaction (average, maximum)*

*Throughput, for example, transactions per second*

*Capacity, for example, the number of customers or transactions the system can accommodate*

*Resource utilization, such as memory, disk, communications, and so forth.]*

#### 3.2.3 …

## 5. Requirement Appendix

*[Provide business rules, common requirements, or other extra requirements information here]*

### 5.1 Business Rules

*[Provide common business rules that you must follow. The information can be provided in the table format as the sample below]*

|  |  |
| --- | --- |
| **ID** | **Rule Definition** |
| BR-01 | Delivery time windows are 15 minutes, beginning on each quarter hour. |
| BR-02 | Deliveries must be completed between 10:00 A.M. and 2:00 P.M. local time, inclusive. |
| BR-03 | All meals in a single order must be delivered to the same location. |
| BR-04 | All meals in a single order must be paid for by using the same payment method. |
| BR-11 | If an order is to be delivered, the patron must pay by payroll deduction. |
| BR-12 | Order price is calculated as the sum of each food item price times the quantity of that food item ordered, plus applicable sales tax, plus a delivery charge if a meal is delivered outside the free delivery zone. |
| BR-24 | Only cafeteria employees who are designated as Menu Managers by the Cafeteria Manager can create, modify, or delete cafeteria menus. |
| BR-33 | Network transmissions that involve financial information or personally identifiable information require 256-bit encryption. |
| BR-86 | Only regular employees can register for payroll deduction for any company purchase. |
| BR-88 | An employee can register for payroll deduction payment of cafeteria meals if no more than 40 percent of his gross pay is currently being deducted for other reasons. |

### 5.2 System Messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In line | There is not any search result | *No search results.* |
| 2 | MSG02 | In red, under the text box | Input-required fields are empty | *The \* field is required.* |
| 3 | MSG03 | Toast message | Updating asset(s) information successfully | *Update asset(s) successfully.* |
| 4 | MSG04 | Toast message | Adding new asset successfully | *Add asset successfully.* |
| 5 | MSG05 | Toast message | Confirming email of asset hand-over is sent successfully | *A confirmation email has been sent to {email\_address}.* |
| 6 | MSG06 | Toast message | Resetting asset information successfully | *Return asset(s) successfully.* |
| 7 | MSG07 | Toast message | Deleting asset information successfully | *Delete asset(s) successfully.* |
| 8 | MSG08 | In red, under the text box | Input value length > max length | *Exceed max length of {max\_length}.* |
| 9 | MSG09 | In line | Username or password is not correct when clicking sign-in | *Incorrrect user name or password. Please check again.* |
| 10 | .. |  |  |  |

### 5.3 Other Requirements…