

Requirement & Design Specification

**Global Assess Management System (GAMS)**

**Version: 1.0**

– Hanoi, August 2022 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | A\* M, D | In charge | Change Description |
| V1.0 | 15/2 | A | KienNTHE11 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[Record of Changes 2](#_Toc167292851)

[I. Overview 4](#_Toc167292852)

[1. User Requirements 4](#_Toc167292853)

[1.1 Actors 4](#_Toc167292854)

[1.2 Use Cases 4](#_Toc167292855)

[2. System Functionalities 5](#_Toc167292856)

[2.1 Screens Flow 5](#_Toc167292857)

[2.2 Screen Authorization 5](#_Toc167292858)

[2.3 Non-UI Functions 6](#_Toc167292859)

[3. High Level Design 6](#_Toc167292860)

[3.1 Database Design 6](#_Toc167292861)

[3.2 Code Packages 7](#_Toc167292862)

[II. Functional Requirements 8](#_Toc167292863)

[1. <<Feature Name1>> 8](#_Toc167292864)

[1.1 <<Screen/Function Name1>> 8](#_Toc167292865)

[1.2 User Login 8](#_Toc167292866)

[1.3 … 8](#_Toc167292867)

[2. System Administration 8](#_Toc167292868)

[2.1 System Settings 8](#_Toc167292869)

[2.2 … 9](#_Toc167292870)

[3. … 9](#_Toc167292871)

[III. Detailed Designs 9](#_Toc167292872)

[1. <Feature/Function Name1> 9](#_Toc167292873)

[1.1 Class Diagram 9](#_Toc167292874)

[1.2 Sequence Diagram(s) 10](#_Toc167292875)

[1.3 Database Queries 10](#_Toc167292876)

[2. <Feature/Function Name2> 10](#_Toc167292877)

# **I. Overview**

## **1. User Requirements**

### 1.1 Actors

*[An actor is someone/something that interacts with the system.*

* *The only external entities that interact with the system*
* *﻿Actors are outside the system and not part of it*
* *﻿A user is an individual, whereas an actor represents the role played by all users of the same type*
* *There are other types of actors in addition to or in place of human actors: external systems, I/O devices, or timers*

*Following are some questions you might ask to help user representatives identify actors*

* *Who (or what) is notified when something occurs within the system?*
* *Who (or what) provides information or services to the system?*
* *Who (or what) helps the system respond to and complete a task?*

*This part gives the description of system actors, you can follow the table form as below]*

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Administrator | Actor description here.. |
| 2 | Menu Manager | .. |
| 3 | … |  |

### 1.2 Use Cases

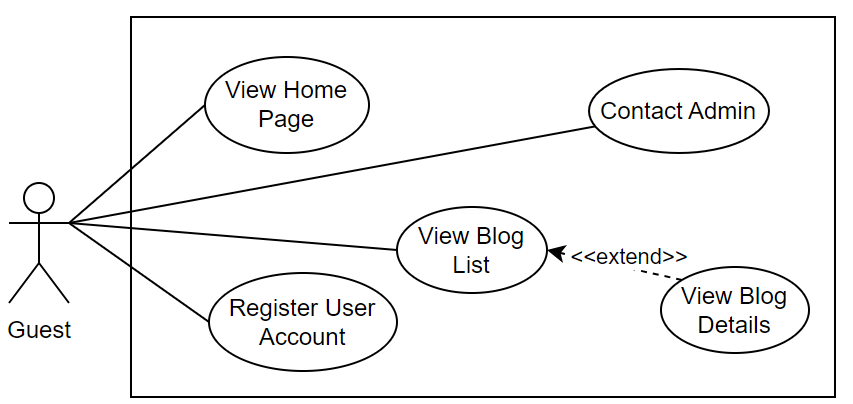
*[A use case (UC) describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value. The names of use cases are always written in the form of a verb followed by an object. Select strong, descriptive names to make it evident from the name that the use case will deliver something valuable for some user.*

*Following are some questions you might ask to help user representatives identify use cases*

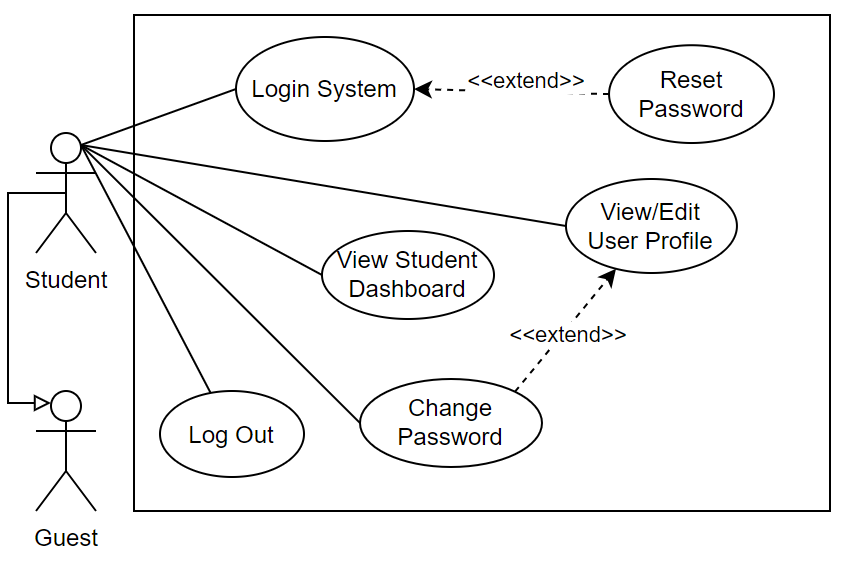
* *What will the actor use the system for?*
* *Will the actor create, store, change, remove, or read data in the system?*
* *Will the actor need to inform the system about external events or changes?*
* *Will the actor need to be informed about certain occurrences in the system?*

*In this section, you need to provide the UC diagram(s) to show the actor-UCs and UC-UC relationships like the sample below. You can have multiple UC diagrams for the system, each diagram is for one actor or one workflow]*

#### 1.2.1 UCs for Guest

**

#### 1.2.2 UCs for Student

**

#### 1.2.3 …

## **2. System Functionalities**

*[Provide functionality overview of software system: screen flow, screen descriptions, system user roles, screen authorization, non-screen functions, ERD]*

### 2.1 Screens Flow (Trong Screen Flow, mỗi "screen" là một màn hình chức năng)

*[This part shows the system screens and the relationship among screens. You can draw the Screens Flow for the system in the form of diagram as below.]*



### 2.2 Screen Authorization

*[Provide the system roles authorization to the system features (down to screens, and event to the screen activities if applicable) in the table form as below – replace Role-Name1, Role-Name2,… with your specific system user role names]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Role-Name1** | **Role-Name2** | **Role-Name3** | **…** |
| <<Screen Name1>> | X |  | X | X |
| <<Screen Activity>> |  |  | X | X |
| <<Screen Name2>> | X |  | X |  |
| Query All Data | X |  |  |  |
| Query Own Data |  |  | X |  |
| Add New Data |  |  | X | X |
| Update All Data |  |  |  | X |
| Update Own Data |  |  |  | X |
| Delete Data |  |  |  |  |
| … |  |  |  |  |

### 2.3 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 | … |  |  |

## **3. High Level Design**

### 3.1 Database Design

*[Provide the tables relationship like example below]*

#### 3.1.1 Database Schema



#### 3.1.2 Table Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *<Table name>* | *<Description of the table>*  *- Primary keys: <<list of primary key fields>>*  *- Foreign keys: <<list of foreign key fields>>*  *- Unique keys:<<list of unique field or unique field set>>* |
| *02* | *<Table name2>* | *…* |

### 3.2 Code Packages

*[Provide the package diagram for the system (or sub systems) and package description similar to sample diagram and using description table format below]*

#### 3.2.1 Package Diagram



#### 3.2.2 Package Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *member\_authority* | *<Description of the package: purpose, contents,..>* |
| *02* | *registration* | *<Description of the package: purpose, contents,..>* |
| *03* | *…* |  |

# II. 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]*

## **1. <<Feature Name1>>**

### 1.1 <<Screen/Function Name1>>

*[Screen/Function description]*

*[Screen layout(s)]*

*[Screen specifications: field initializations, the showing/hiding of some fields, etc.]*

### 1.2 User Login

This screen allows user to be authenticated to the system screens/functionalities.

|  |  |
| --- | --- |
| ***S1\_User Login screen*** | ***S2\_Select account to login (with Google)*** |
|  |  |

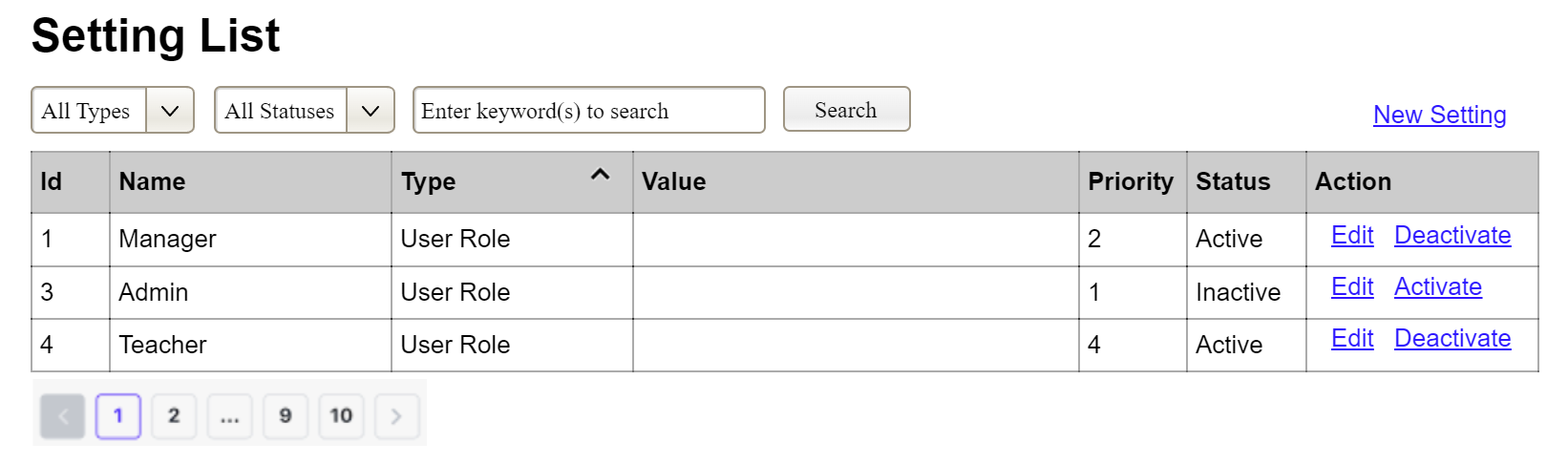
### 1.3 …

## **2. System Administration**

### 2.1 System Settings

#### 2.1.1 Setting List

This is for the administrator to view the list of current system settings. On the screen, s/he can also activate or deactivate (change status) of a specific setting.



(2)

(3)

(1)

(1) Setting Type:

* Initialized with all the active setting types filled in,
* Allow user to filter the setting list by a specific setting type
* Default value is “All Types”, allowing user to see the settings at all types

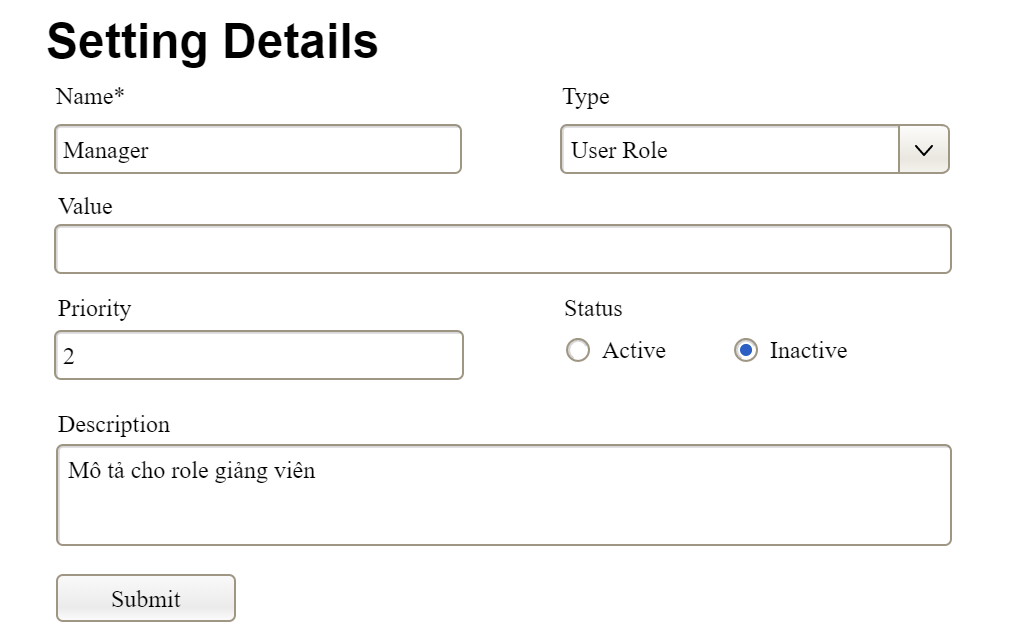
(2) Setting Status:

* Initialized with two values Active and Inactive filled in
* Allow user to filter the setting list by a specific status (Active or Inactive)
* Default value is “All Types”, allowing user to see the settings at all statuses

(3) The change-status action is Activate or Deactivate depending on the current status of the relevant setting (Inactive or Active, respectively). The user needs to confirm the status changing via a modal confirmation message before getting that done.

#### 2.1.2 Setting Details

This is for the administrator to add new or view/update an existing system setting



### 2.2 …

## 3. …

# III. Detailed Designs

## 1. <Feature/Function Name1>

*[Provide the detailed design for the function <Feature/Function Name1>. It includes class diagram and sequence diagram(s)]*

### **1.1 Class Diagram**

*[This part presents the class diagram for the relevant feature]*



### **1.2 Sequence Diagram(s)**

*[Provide the sequence diagram(s) for the feature/function, see the sample below. Sequence diagrams with similar message flow can be omitted, but you need to mention the appropriate reference to the similar sequence diagram]*



### **1.3 Database Queries**

*[Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]*

## 2. <Feature/Function Name2>

…