

OFFICE BUILDING VISITOR MANAGEMENT SYSTEM FUNCTIONAL REQUIREMENTS DOCUMENT

BY

OWEN MIANO KABUGI

Contents

[1. User Management Process 1](#_Toc155363201)

[1.1 Introduction 1](#_Toc155363202)

[1.2 User Types 1](#_Toc155363203)

[1.3 User Data Dictionary 1](#_Toc155363204)

[1.4 Processes 2](#_Toc155363205)

[1.4.1 Add user 2](#_Toc155363206)

[1.4.2 Edit user details 3](#_Toc155363207)

[1.4.3 Change User Password 4](#_Toc155363208)

[1.4.4 Deactivate user’s account 5](#_Toc155363209)

[1.4.5 Reset User Account Password 6](#_Toc155363210)

[2 Building Registration Process 8](#_Toc155363211)

[2.1 Introduction 8](#_Toc155363212)

[2.2 Building Data Dictionary 8](#_Toc155363213)

[2.3 Processes 8](#_Toc155363214)

[2.3.1 Adding Building 8](#_Toc155363215)

[2.3.2 Edit a building 9](#_Toc155363216)

[2.3.3 Remove a building 10](#_Toc155363217)

[3. Building Management Personnel Registration Process 12](#_Toc155363218)

[3.1 Introduction 12](#_Toc155363219)

[3.2 Building Users Data Dictionary 12](#_Toc155363220)

[3.3 Processes 12](#_Toc155363221)

[3.3.1 Adding Building management personnel 12](#_Toc155363222)

[4. Floor Registration Process 13](#_Toc155363223)

[4.1 Introduction 13](#_Toc155363224)

[**Participating Actors** 13](#_Toc155363225)

[4.2 Floor Data Dictionary 13](#_Toc155363226)

[4.3 Processes 14](#_Toc155363227)

[4.3.1 Adding floor 14](#_Toc155363228)

[4.3.2 Edit a floor 15](#_Toc155363229)

[4.3.3 Remove a floor 16](#_Toc155363230)

[5. Office Registration Process 17](#_Toc155363231)

[5.1 Introduction 17](#_Toc155363232)

[5.2 Office Data Dictionary 17](#_Toc155363233)

[5.3 Processes 17](#_Toc155363234)

[5.3.1 Adding office 17](#_Toc155363235)

[5.3.2 Editing office 19](#_Toc155363236)

[5.3.3 Removing office 20](#_Toc155363237)

[6. Physical Card Registration Process 21](#_Toc155363238)

[6.1 Introduction 21](#_Toc155363239)

[6.2 Card Data Dictionary 21](#_Toc155363240)

[6.3 Processes 21](#_Toc155363241)

[6.3.1 Adding physical card 21](#_Toc155363242)

[6.3.2 Editing physical card 22](#_Toc155363243)

[6.3.3 Removing physical card 24](#_Toc155363244)

[7. Host registration 25](#_Toc155363245)

[7.1 Introduction 25](#_Toc155363246)

[7.2 Host Data Dictionary 25](#_Toc155363247)

[7.3 Processes 25](#_Toc155363248)

[7.3.1 Adding Host 25](#_Toc155363249)

[7.3.2 Edit host 26](#_Toc155363250)

[7.3.3 Remove Host 27](#_Toc155363251)

[8. Host Users registration 29](#_Toc155363252)

[8.1 Introduction 29](#_Toc155363253)

[8.2 Host Users Data Dictionary 29](#_Toc155363254)

[8.3 Processes 29](#_Toc155363255)

[8.3.1 Adding host user 29](#_Toc155363256)

[9. Visitor management process 31](#_Toc155363257)

[9.1 Introduction 31](#_Toc155363258)

[9.2 Visit Data Dictionary 31](#_Toc155363259)

[9.3 Processes 32](#_Toc155363260)

[9.3.1 Visitor pre-registration process 32](#_Toc155363261)

[9.3.2 Visitor check-in process 33](#_Toc155363262)

[9.3.3 Host check-out Process. 35](#_Toc155363263)

[9.3.4 Building check-out Process 36](#_Toc155363264)

# 1. User Management Process

## 1.1 Introduction

A user is an entity in the Office Building Visitor Management system.

This involves setting up user roles, using secure ways for users to log in, and specifying what each user can do. A well-organized user system makes sure the system is secure, works well, and follows the rules, creating a smooth and controlled experience for users.

## 1.2 User Types

There are three kinds of users, each with different abilities:

* **Super Admins**: Possessing comprehensive privileges.
* **Building Management Personnel**: Has defined privileges in the system. Responsible for the check-in and check-out process for visitors from the building.
* **Host Users**: Host users has defined privileges in the system. Access to features for managing pre-registration of visitors. Authority to check-in and check-out visitors within their office.
* **Visitors**: Visitors have no privileges in the system

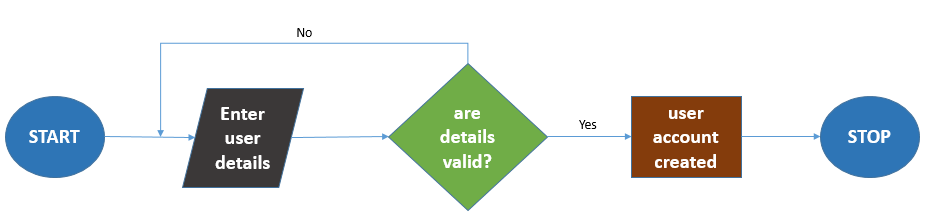
## 1.3 User Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| name | STRING | YES | Name of the user. |
| id\_number | INT(8) | YES | Specifies the type of identification document i.e.  1. NATIONAL\_ID |
| phone | STRING | YES | Valid phone number of user |
| email | STRING | YES | Valid email address of the user |
| password | STRING | YES | Valid strong password of the user |
| role | DROPDOWN | YES | Indicating the role to which the user belongs |
| status | STRING | YES | Represents the account status active  inactive |

## 1.4 Processes

### 1.4.1 Add user

The following section describes the process of adding user details within the system.



**Participating actors**

**System Admin**: Responsible for initiating and overseeing the user addition process.

**New User**: The individual for whom the user account is being created.

**Pre-conditions**

Before the system admin adds a user, it is assumed that:

* The system is operational and accessible.
* The system admin is logged in and has the necessary authorization to add users.

**Flow of Events**

1. The system presents the admin with an interface to add a new user.
2. The system should enable the admin fill in the required details for the new user, adhering to the user data dictionary.
3. The system conducts real-time validations, ensuring that all mandatory fields are filled and the entered data complies with predefined formats.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the user addition process, it immediately identifies the issues.
2. The system generates error messages within the user interface, guiding the admin on the necessary corrections.
3. The admin, operating within the system, reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the user addition form.

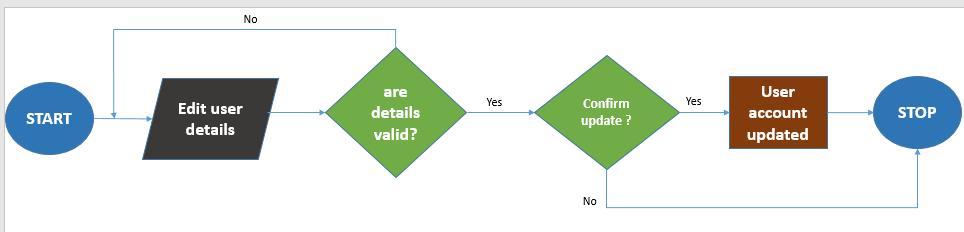
**Output**

Upon successful input of the required details by the administrator, the expected output will be the creation of a user account, signifying the successful addition of the new user to the system.

The system will additionally trigger a notification to be sent to the newly created user, providing relevant information and welcoming them to the system.

### 1.4.2 Edit user details

The following section describes the process of updating user details within the system.



**Participating Actors:**

**User:** The individual initiating the process to update their account

**Preconditions:**

Before the user updates their account, it is assumed that:

* The system is operational and accessible.
* The user is logged in and has the necessary authorization to update their user details.

**Flow of Events**

1. The system allows the user to modify their necessary account details, adhering to the user data dictionary.
2. Real-time validations are conducted to ensure accuracy and compliance with predefined formats for the updated information.
3. The system should enable the user to confirm the decision to update their details.

**Alternate Flow of Events**

In the specified update process, the user's password is excluded from modification.

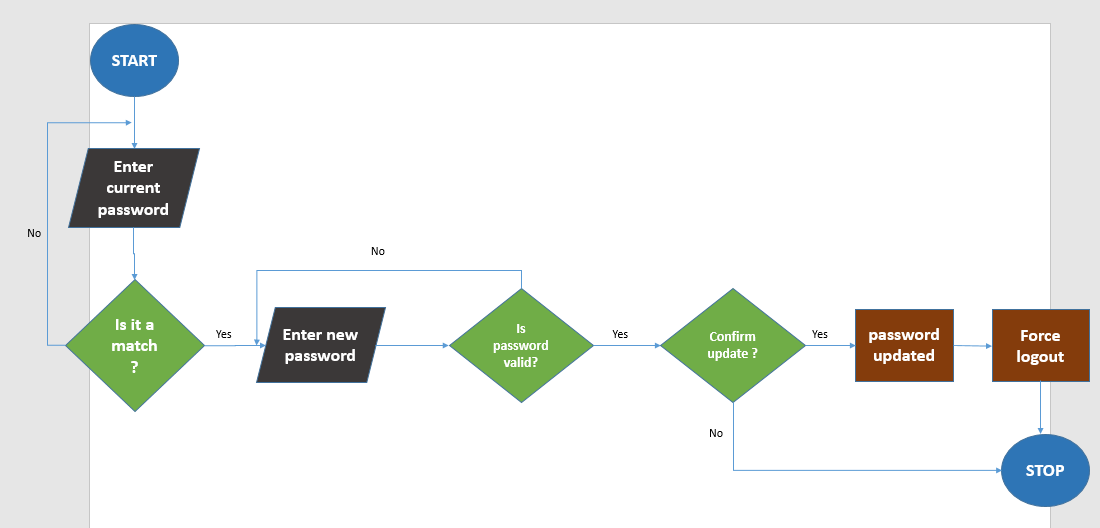
1. If the system detects incorrect or incomplete data during the update process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the user on necessary corrections.
3. The user, operating within the system, reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the user to resubmit the updated their details.

**Output**

Upon successful confirmation by the user, the expected output is the updated user account, reflecting the changes made in the system. Additionally, the system may trigger a notification to inform the user about the modifications made to their account.

### 1.4.3 Change User Password

The following section outlines the steps involved in changing one's password within the system.



**Participating Actors:**

**User**: The individual initiating the process to change their password.

**Preconditions:**

Before the user changes their password, it is assumed that:

* The system is operational and accessible.
* The user is logged in and has the necessary authorization to change their password.

**Flow of Events**

1. Within the interface, the system provides the necessary fields for the user to input their current password and set a new password.
2. The system permits the user to input the necessary information, ensuring compliance with the system's predefined password criteria
3. Real-time validations are conducted by the system to ensure the accuracy and compliance of the entered data.
4. The system should enable the user to confirm the decision to update their details.

**Alternate Flow of Events**

* If the system detects incorrect or incomplete data during the password change process, it immediately identifies the issues.
* Error messages are generated within the user interface, guiding the user on necessary corrections.
* The user reviews the error messages and rectifies the information as guided by the system.
* Once the corrections are made, the system re-initiates the validation process before allowing the user to resubmit the password change request.

**Output:**

Upon successful confirmation by the user, the expected output is the updated authentication credentials, reflecting the changes made in the system. The system may also trigger a notification to inform the user about the successful password change of their account.

### 1.4.4 Deactivate user’s account

This process focuses on the deactivation of a user account, ensuring that the system admin can securely and effectively manage user access within the system.

**Participating Actors**

**System Admin:** Initiates and oversees the process of deactivating a user account.

**User (Previously Created):** The individual for whom the user account is being deactivated.

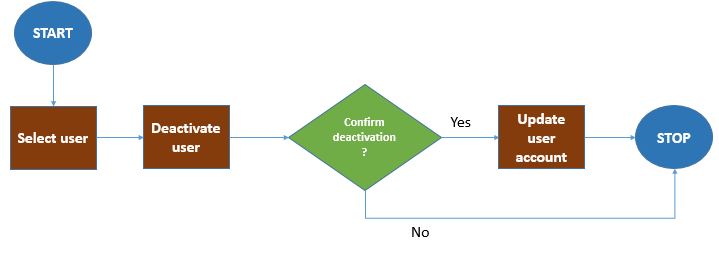
**Preconditions:**

Before the system admin deactivates a user account, it is assumed that:

The system is operational and accessible.

The system admin is logged in and has the necessary authorization to deactivate user accounts.

**Flow of Events**



1. Within the interface, the system enables the admin to select the specific user account that needs to be deactivated.
2. The system should enable the admin to confirm the decision to deactivate the user’s account.
3. The system processes the deactivation request, updating the user account status accordingly.

**Output:**

Upon successful confirmation by the administrator, the expected output is the updated user account status, indicating that the account has been deactivated in the system. Additionally, the system may trigger a notification to inform the user about the deactivation of their account.

This process focuses on the deactivation of a user account, ensuring that the system admin can securely and effectively manage user access within the system.

### 1.4.5 Reset User Account Password

This section outlines the process for resetting a user's account password.

**Participating Actors**

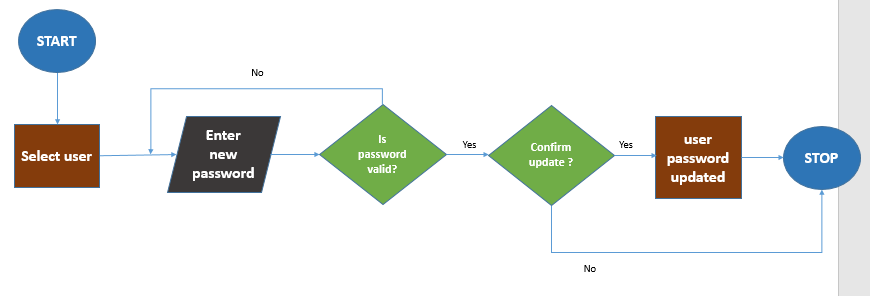
* **System Admin:** Initiates the process of resetting a user's account password**.**
* **User:** The individual for whom the account password is being reset**.**

**Preconditions:**

Before the system admin resets a user's account password, it is assumed that:

* The system is operational and accessible.
* The system admin is logged in and has the necessary authorization to reset user account passwords.

**Flow of Events**



1. Within the interface, the system enables the admin to select the specific user account for which the password needs to be reset.
2. The system generates a temporary password or provides a secure mechanism for the admin to set a new password on behalf of the user.
3. The system admin inputs the necessary information, adhering to the system's predefined password criteria.
4. Real-time validations are conducted by the system to ensure the accuracy and compliance of the entered data.
5. The system should enable the admin to confirm the decision to update the user’s password.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the password reset process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The system admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the password reset and building addition request.

**Output:**

Upon successful confirmation by the administrator, the expected output is the updated authentication credentials, reflecting the changes made in the system. The system may also trigger a notification to inform the user about the password reset for their account.

# 2 Building Registration Process

## 2.1 Introduction

The purpose of the building registration process is to record and validate essential details about a new building within the system.

## 2.2 Building Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| reg\_no | STRING | YES | is a unique identifier assigned to a building for official records, documentation, or regulatory purposes |
| name | STRING | YES | Name of the building. |
| location | STRING | YES | Location of the building. |
| developers | STRING | NO | Individuals or entities responsible for building the structure. |
| built\_date | date | NO | Date when the building was constructed. |

## 2.3 Processes

### 2.3.1 Adding Building

The following section describes the process of adding building details within the system.

**Participating Actors**

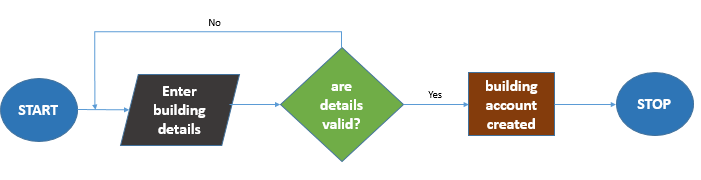
* **Building owner**: provides the building details
* **System Admin**: Initiates the process of adding a building.

**Preconditions**

Before the system admin adds a building, it is assumed that:

* The system is operational and accessible.
* The system admin is logged in and has the necessary authorization to add buildings.

**Flow of Events**



1. Within the interface, the system enables the admin to input the necessary information about the building, such as building name, location, and other relevant details.
2. Real-time validations are conducted by the system to ensure the accuracy and completeness of the entered building information.
3. Upon successful validation, the system updates its records to include the newly added building.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the building addition process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The system admin, along with the building owner, reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the building addition request.

**Output**

The expected output is the establishment of a new building account in the system.

### 2.3.2 Edit a building

The following section describes the process of updating building details within the system.

**Participating Actors:**

**System Admin:** Initiates the process of editing building details.

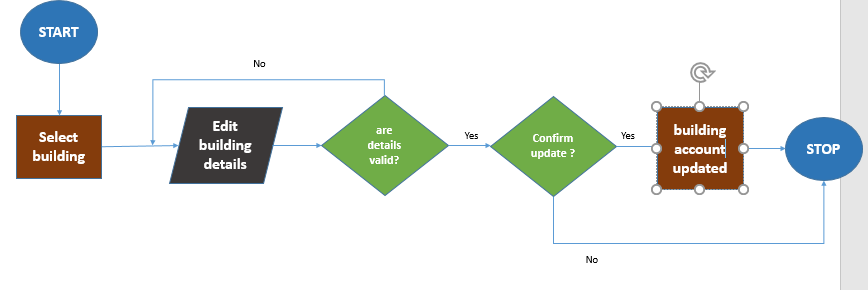
**Building Owner:** Individual responsible for the building.

**Preconditions**

Before the system admin edits building details, it is assumed that**:**

* The system is operational and accessible.
* The system admin is logged in and has the necessary authorization to edit building details.

**Flow of Events**



1. Within the interface, the system displays a list of existing buildings, and the admin selects the specific building to edit.
2. The system should enable the admin modify the necessary details of the selected building, such as name, location, or other relevant information.
3. Real-time validations are conducted by the system to ensure the accuracy and completeness of the edited building information.
4. The system should enable the admin to confirm the decision to update the building’s details.

**Alternate Flow of Events:**

1. If the system detects incorrect or incomplete data during the building editing process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The system admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the building editing request.

**Output:**

Upon successful confirmation by the administrator, the expected output is the updated records reflecting the changes made in the system.

### 2.3.3 Remove a building

The following section describes the process of removing a building account from the system.

**Participating Actors:**

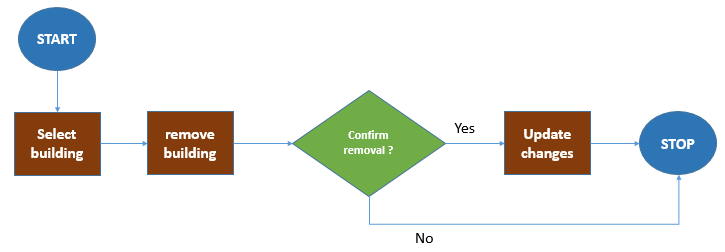
**System Admin:** Initiates the process of removing a building account**.**

**Preconditions:**

Before the system admin removes a building account, it is assumed that**:**

* The system is operational and accessible.
* The system admin is logged in and has the necessary authorization to remove building accounts.

**Flow of Events**



1. Within the interface, the system presents a list of existing buildings, and the admin selects the specific building account to be removed.
2. The system should enable the admin to confirm the decision to remove the building account and provide any necessary justifications.

**Output**

Upon successful confirmation by the administrator, the expected output is the updated records reflecting the removal of the building from the system.

# 3. Building Management Personnel Registration Process

## 3.1 Introduction

The purpose of the building management registration process is to enable the registration of individuals or entities responsible for managing and overseeing building-related activities within the system.

## 3.2 Building Users Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| building\_id | DROPDOWN | YES | Foreign key referencing the building table. Representing the building to which the personnel belongs to |
| user\_id | DROPDOWN | YES | Foreign key referencing the user table. Establishes a relationship with the user\_id field in the user table. |

## 3.3 Processes

### 3.3.1 Adding Building management personnel

The following section describes the process of adding building management personnel details within the system.

**Participating Actors**

**System Admin**: Initiates the process of adding a building management personnel.

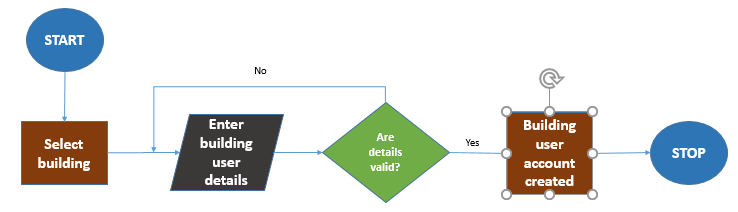
**Building Management Personnel**: Individuals or entities seeking registration for building management roles.

**Preconditions**:

Before the system admin adds building management personnel, it is assumed that:

* The system is operational and accessible.
* The system admin is logged in and has the necessary authorization to add personnel.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific building to which the personnel will be associated.
2. The admin then inputs the necessary details for the new personnel, adhering to the user data dictionary.
3. The system conducts real-time validations, ensuring all mandatory fields are filled, and the entered data complies with predefined formats.

**Alternate Flow of Events:**

1. If the system detects incorrect or incomplete data during the personnel addition process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the personnel addition form.

**Output**

Upon successful input of the required details by the administrator, the expected output is the creation of a building management personnel account associated with the selected building. The system may also trigger notifications to inform the new personnel about their account creation and building association.

# 4. Floor Registration Process

## 4.1 Introduction

The purpose of the floor registration process is to systematically record and manage information related to individual floors within a building**.**

**Participating Actors**

**Authorized Building Management Personnel:** Individuals responsible for managing and overseeing floor-related activities.

## 4.2 Floor Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| building\_id | DROPDOWN | YES | Foreign key referencing the building table. Representing the building to which the floor belongs to |
| floor | STRING | YES | 1st Floor,2nd floor |

## 4.3 Processes

### 4.3.1 Adding floor

The following section describes the process of adding floor details within the system.

**Participating Actors**:

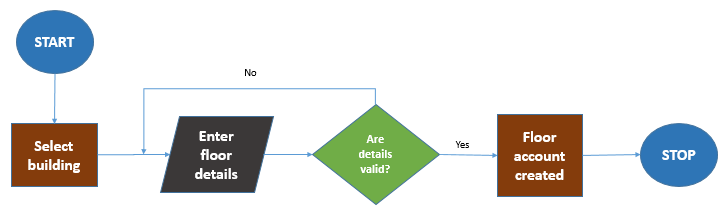
**System Admin**: Initiates the process of registering a floor.

**Preconditions**:

Before the system admin registers a floor, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to register floors.

**Flow of Events**



1. Within the interface, the system should enable the admin to first select the specific building to which the floor will be associated.
2. The system should the admin to input floor details, adhering to the floor data dictionary.

**Alternate Flow of Events:**

1. If the system detects incorrect or incomplete data during the floor registration process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the floor registration form.

**Output**

Upon successful validation, associated with the selected building, the expected output is the creation of a floor record.

### 4.3.2 Edit a floor

The following section describes the process of updating floor details within the system.

**Participating Actors**

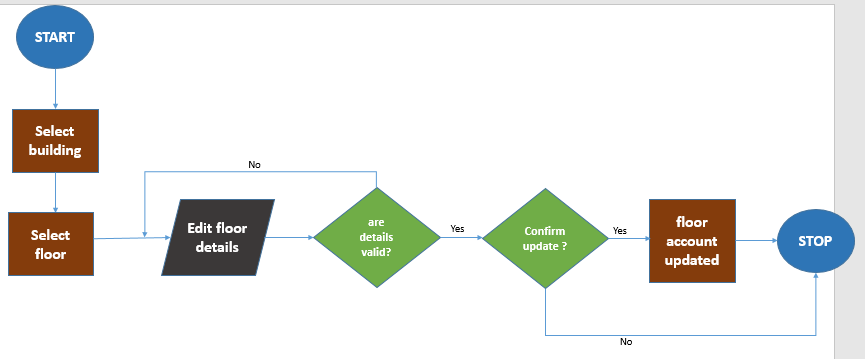
**System Admin**: Initiates the process of registering a floor.

**Preconditions**

Before the system admin edits a floor, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to register floors.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific building and floor for which details need to be modified.
2. The admin modifies the necessary floor details, adhering to the floor data dictionary.
3. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the updated information.
4. The system should enable the admin to confirm the decision to update the building’s details.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the floor editing process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to confirm the floor details update.

**Output**

Upon successful confirmation by the administrator, the expected output is the updated floor details, reflecting the changes made in the system.

### 4.3.3 Remove a floor

The following section describes the process of removing a floor within the system.

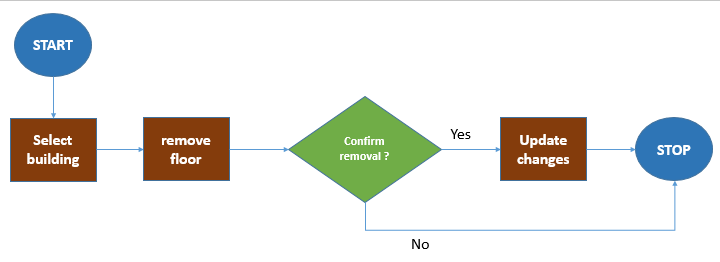
**Participating Actors**

**System Admin**: Initiates the process of removing a floor.

**Preconditions**:

Before the system admin removes a floor, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to remove floors.

**Flow of Events** 

1. Within the interface, the system enables the admin to first select the specific building and floor for removal.
2. The system should enable the admin to confirm the decision to remove the floor account and provide any necessary justifications.

**Output**

Upon successful confirmation by the administrator, the expected output is the updated records reflecting the removal of the floor from the system.

# 5. Office Registration Process

## 5.1 Introduction

The purpose of the office registration process is to systematically record and manage information related to individual offices within a building**.**

## 5.2 Office Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| floor\_id | DROPDOWN | YES | Foreign key referencing the floor table. Representing the floor to which the office belongs to |
| office\_no | STRING | YES | Office number to uniquely identify an office |
| Occupancy\_status | STRING | YES | Occupied  unoccupied |

## 5.3 Processes

### 5.3.1 Adding office

The following section describes the process of adding office details within the system.

**Participating Actors**

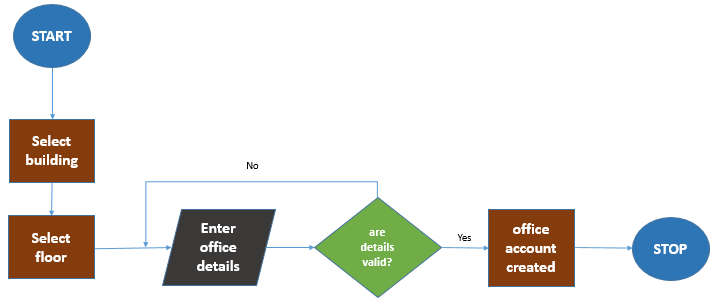
**System Admin**: Initiates the process of adding an office.

**Preconditions**

Before the system admin adds an office, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to add offices.

**Flow of Events**



1. The system admin accesses the office addition interface within the system.
2. Within the interface, the system enables the admin to first select the specific building to which the office will be associated.
3. The system should further allow the admin to select the floor within the chosen building where the office is located.
4. The system then allows the admin to input necessary details for the new office, adhering to the office data dictionary.
5. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the entered information.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the office addition process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the office addition form.

**Output**

Upon successful input of the required details by the system admin, associated with the selected building and floor, the expected output is the creation of an office record within the system.

### 5.3.2 Editing office

The following section describes the process of updating office details within the system.

**Participating Actors**

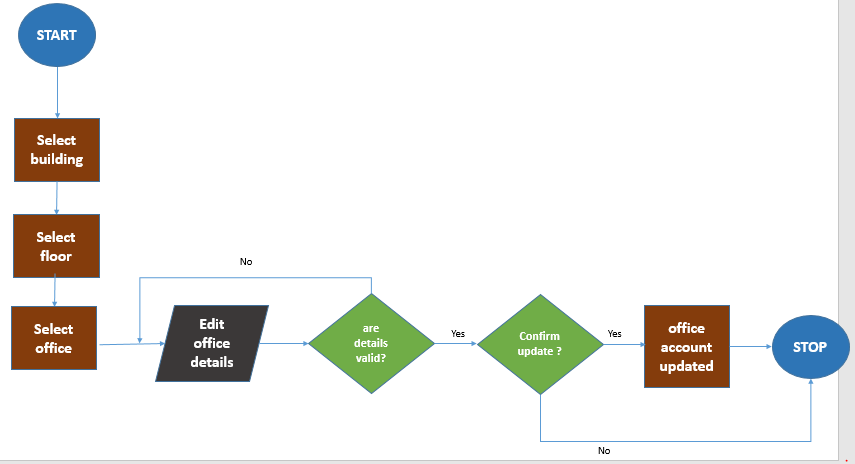
**System Admin**: Initiates the process of editing office details.

**Preconditions**:

Before the system admin edits office details, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to edit offices.

**Flow of Events**



1. The system admin accesses the office editing interface within the system.
2. Within the interface, the system enables the admin to first select the specific building to which the office is associated.
3. The system should further enable the admin selects the floor within the chosen building where the office is located.
4. The system then allows the admin to select the specific office for which details need to be modified.
5. The admin modifies the necessary office details, adhering to the office data dictionary.
6. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the updated information.
7. The system should enable the admin to confirm the decision to update the office’s details

**Alternate Flow of Events:**

1. If the system detects incorrect or incomplete data during the office editing process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to confirm the office details update.

**Output:**

Upon successful confirmation by the administrator, the expected output is the updated office details, reflecting the changes made in the system

### 5.3.3 Removing office

The following section describes the process of deleting office details within the system.

**Participating Actors**

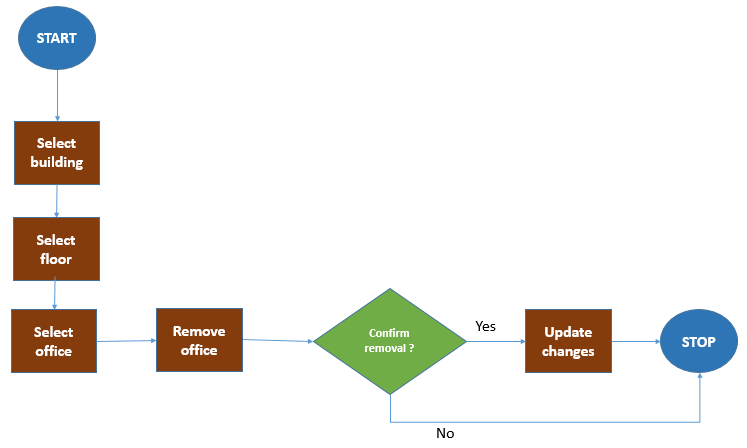
**System Admin**: Initiates the process of removing a floor.

**Preconditions**:

Before the system admin removes a floor, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to remove floors.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific building and floor in which the office belongs.
2. The system should further enable the admin to select the office for removal
3. The system should enable the admin to confirm the decision to remove the office account and provide any necessary justifications.

**Output**

Upon successful confirmation by the administrator, the expected output is the updated records reflecting the removal of the office from the system.

# 6. Physical Card Registration Process

## 6.1 Introduction

The purpose of the physical card registration process is to systematically record and manage information related to physical access cards issued to individuals.

## 6.2 Card Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| building\_id | DROPDOWN | YES | Foreign key referencing the building table. Representing the building to which the personnel belongs to |
| card\_no | STRING | YES | Card number |
| assignment\_status | STRING | YES | Assigned  Unassigned |

## 6.3 Processes

### 6.3.1 Adding physical card

The following section describes the process of adding physical card details within the system.

**Participating Actors**

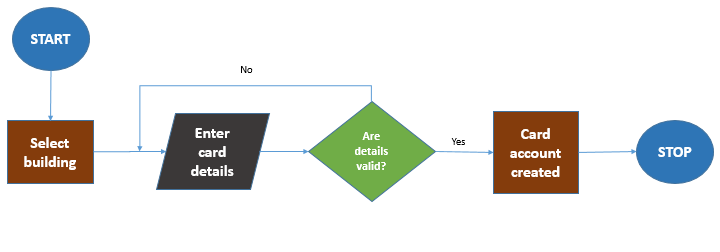
* **System Admin:** Initiates the process of adding a physical card.

**Preconditions**

Before the system admin adds a physical card, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to add physical cards.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific building to which the physical card will be associated.
2. The system should enable admin inputs necessary details for the new physical card, adhering to the card data dictionary.
3. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the entered information.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the physical card addition process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the physical card addition form.

**Output**

Upon successful validation, the expected output is the creation of a physical card record within the system.

### 6.3.2 Editing physical card

The following section describes the process of editing physical card details within the system.

**Participating Actors**

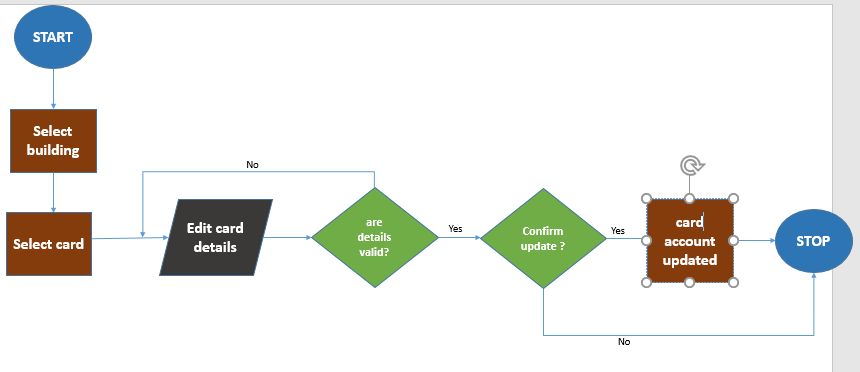
**System Admin**: Initiates the process of editing physical card details.

**Preconditions**

Before the system admin edits physical card details, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to edit physical cards.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific building and physical card for which details need to be modified.
2. The admin modifies the necessary physical card details, adhering to the card data dictionary.
3. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the updated information.
4. The system should enable the admin to confirm the decision to update the physical card’s details.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the physical card editing process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to confirm the physical card details update.

**Output**

Upon successful confirmation by the administrator, the expected output is the updated physical card details, reflecting the changes made in the system.

### 6.3.3 Removing physical card

The following section describes the process of removing physical card details within the system.

**Participating Actors**

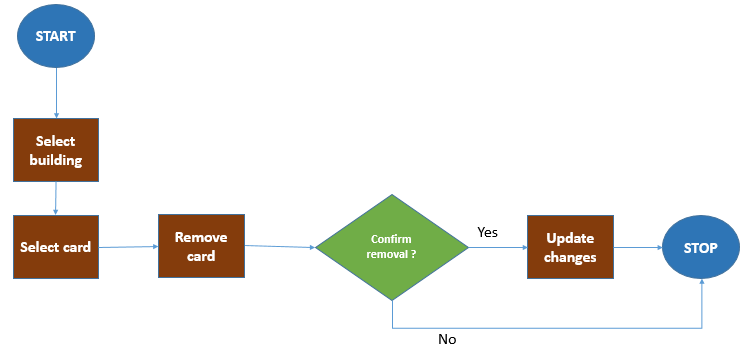
**System Admin**: Initiates the process of removing a physical card.

**Preconditions**

Before the system admin removes a physical card, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to remove physical cards.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific building to which the physical card will be associated.
2. Real-time validations may be conducted by the system to ensure the removal process is valid and authorized.
3. The system should enable the admin to confirm the decision to remove the physical card.

**Output**

Upon successful confirmation by the administrator, the expected output is the updated records reflecting the removal of the physical card from the system

# 7. Host registration

## 7.1 Introduction

A host is an entity in the Office Building Visitor Management system.

This process ensures a structured approach to managing hosts, similar to the process for individuals occupying offices, contributing to streamlined administrative activities within the building.

## 7.2 Host Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| office\_id | DROPDOWN | YES | Foreign key referencing the office table. Representing the office to which the host belongs to |
| availability\_status | DROPDOWN | YES | Available  Not available |
| company\_name | STRING | YES | Name of the company |
| company\_no | STRING | YES | Business registration number for the company. |
| company\_phone | STRING | NO | Phone number of the company. |
| company\_email | DATETIME |  | Email address of the company. |

## 7.3 Processes

### 7.3.1 Adding Host

The following section describes the process of adding host details within the system.

**Participating Actors**

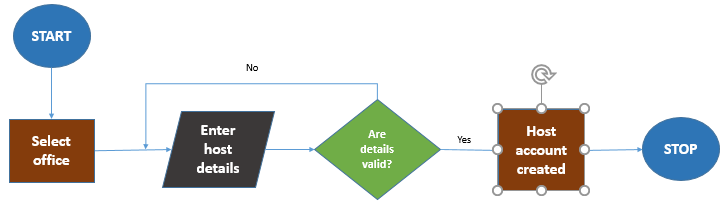
**System Admin**: Initiates the process of add host

**Preconditions**

Before the system admin adds a host, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to add a host.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific office to which the host user will be associated.
2. The system should enable admin inputs necessary details for the new host, adhering to the host data dictionary.
3. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the entered information.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the host addition process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the host addition form.

**Output**

Upon successful validation, the expected output is the creation of a host record within the system.

### 7.3.2 Edit host

The following section describes the process of updating host details within the system.

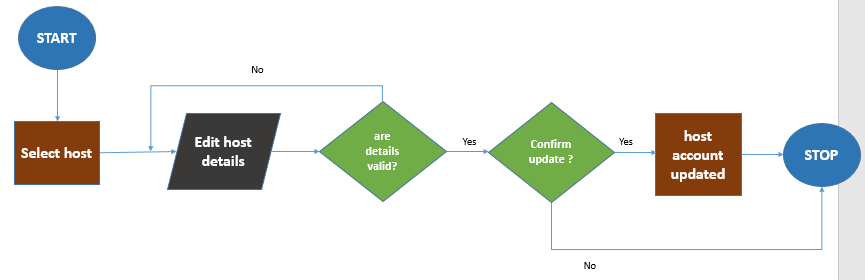
**Participating Actors**

**System Admin**: Initiates the process of editing a host.

**Preconditions**

Before the system admin edits a host, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to edit a host.



**Flow of Events**

1. The system should enable the admin selects the host
2. The admin modifies the necessary host details, adhering to the host data dictionary.
3. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the updated information.
4. The system should enable the admin to confirm the decision to update the host’s details

**Alternate Flow of Events:**

1. If the system detects incorrect or incomplete data during the host editing process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to confirm the host details update.

**Output:**

Upon successful confirmation by the administrator, the expected output is the updated host details, reflecting the changes made in the system

### 7.3.3 Remove Host

The following section describes the process of removing host details from the system.

**Participating Actors**

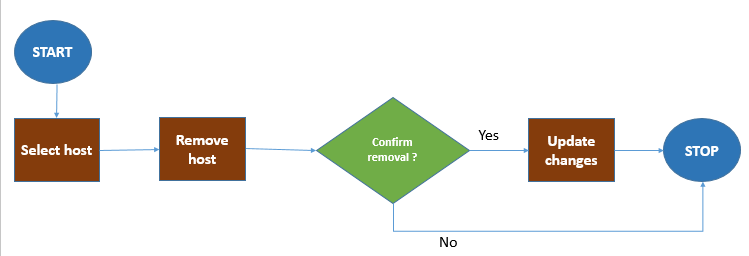
**System Admin**: Initiates the process of removing a host.

**Preconditions**

Before the system admin removes a host, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to remove a host.

**Flow of Events**



1. Within the interface, the system enables the admin to first select the specific host.
2. Real-time validations may be conducted by the system to ensure the removal process is valid and authorized.
3. The system should enable the admin to confirm the decision to remove the host

**Output**

Upon successful confirmation by the administrator, the expected output is the updated records reflecting the removal of the host from the system

# 8. Host Users registration

## 8.1 Introduction

A host user is an entity in the Office Building Visitor Management system.

This process ensures a structured approach to managing hosts, similar to the process for individuals occupying offices, contributing to streamlined administrative activities within the building.

## 8.2 Host Users Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| host\_id | DROPDOWN | YES | Foreign key referencing the host table. Representing the host to which the host users belongs to. |
| user\_id | DROPDOWN | YES | Foreign key referencing the users table. |

## 8.3 Processes

### 8.3.1 Adding host user

The following section describes the process of adding host user within the system.

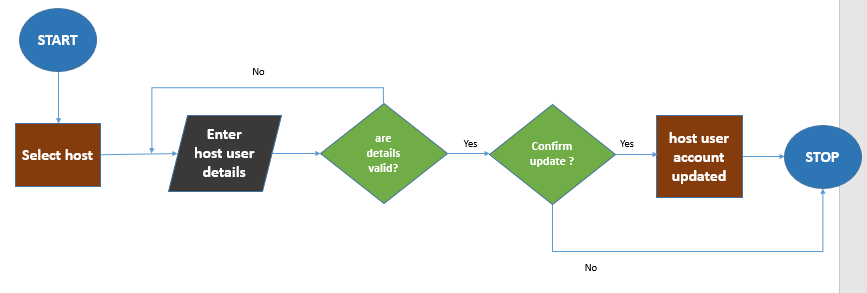
**Participating Actors**

**System Admin**: Initiates the process of add host user

**Preconditions**

Before the system admin adds a host, it is assumed that:

* The system is operational and accessible.
* The admin is logged in and has the necessary authorization to add a host.



1. Within the interface, the system enables the admin to first select the host to which the host user will be associated.
2. The system should enable admin inputs necessary details for the new host user, adhering to the host user data dictionary.
3. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the entered information.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the host user addition process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.
4. Once the corrections are made, the system re-initiates the validation process before allowing the admin to resubmit the host user addition form.

**Output**

Upon successful validation, the expected output is the creation of a host user record within the system.

# 9. Visitor management process

## 9.1 Introduction

The visitor management process typically involves several stages, from pre-registration to check-in and finally to check-out.

## 9.2 Visit Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD | Data Type | Required | Description |
| Full\_name | STRING | YES | Name of the visitor |
| phone | INT | YES | Valid Phone number of the visitor |
| id\_number | INT | YES | 1. Should be 8 digits |
| building | DROPDOWN | YES | identifying the building associated with the visit |
| card | DROPDOWN | YES | identifying the card used by the visitor |
| purpose | DROPDOWN | YES | identifying purpose of the visit |
| pass\_code | STRING | YES | Pass code associated with the visit |
| date\_check\_in\_by\_building | DATETIME | YES | Date and time of visitor check-in by the building management personnel. |
| date\_check\_out\_by\_host | DATETIME | NO | Date and time of visitor check-out initiated by the host. |
| date\_check\_out\_by\_ building | DATETIME | NO | Date and time of visitor check-out initiated by the building management personnel. |
| expected\_arrival\_time | DATETIME | NO | Expected date and time of visitor's arrival. |
| pre\_registered | DROPDOWN | NO | Indicates if the visit is pre-registered. |

## 9.3 Processes

### 9.3.1 Visitor pre-registration process

**Introduction**

The purpose of the visitor pre-registration process is to allow for the efficient and organized entry of visitors into a building by enabling them to register in advance. This process streamlines the check-in procedure, providing a convenient and secure experience for both visitors and building management personnel.

**Participating Actors**

**Host:** Initiates the pre-registration process for a visitor.

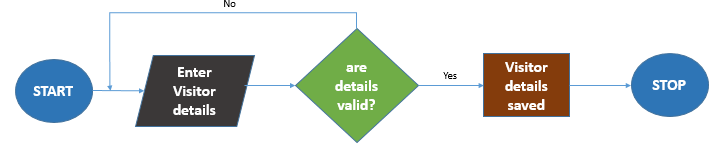
**Visitors:** Provides necessary details for the pre-registration process

**Preconditions**

Before the host user adds a visitor, it is assumed that:

* Visitor has not yet arrived but intends to visit
* The system is operational and accessible.
* The host user is logged in and has the necessary authorization to pre-register a visitor.

**Flow of Events**



1. The system enables the host to enter visitor's personal information, purpose of the visit, expected arrival time.
2. The system should save the visitor's details in the system.
3. The system should enable host user inputs necessary details for the visitor, adhering to the visit data dictionary.
4. Real-time validations are conducted by the system to ensure accuracy and compliance with predefined formats for the entered information.

**Alternate Flow of Events**

1. If the system detects incorrect or incomplete data during the pre-registration process, it immediately identifies the issues.
2. Error messages are generated within the user interface, guiding the admin on necessary corrections.
3. The admin reviews the error messages and rectifies the information as guided by the system.

**Output**

Upon successful validation of the visitor details, the system will create a visitor record

### 9.3.2 Visitor check-in process

**Introduction**

The purpose of the visitor check-in process is to systematically record and manage the entry of visitors into a building.

**Participating actors**

**Visitor:**

Individuals who are entering the building for specific purposes.

**Host User:**

Individuals authorized to host visitors within the building.

**Authorized Building Management Personnel:**

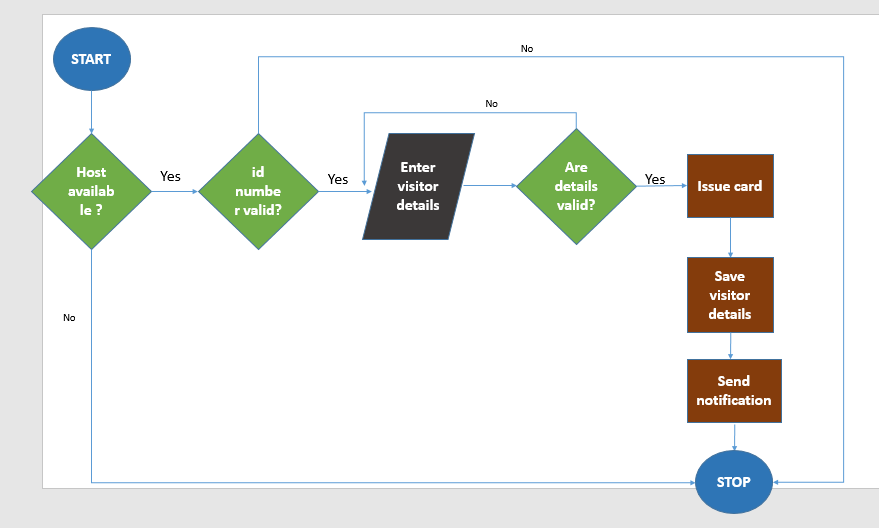
Individuals responsible for overseeing and managing visitor-related activities.

**Preconditions**

* System is operational.
* Authorized personnel is logged in.
* The visitor has arrived at the location.

**Flow of Events**

1. **New Visitor**



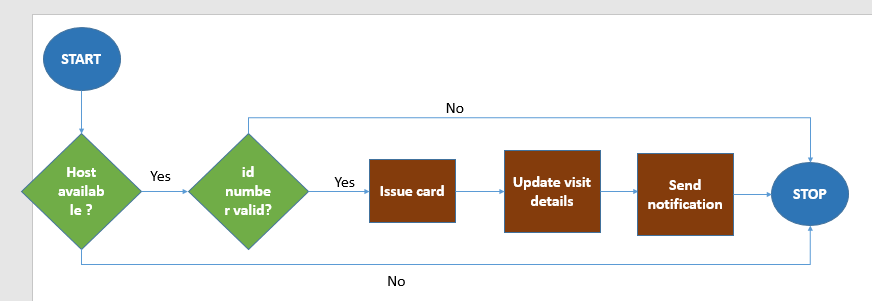
1. The system should verify if the host exists in the system
2. The system should then verify if the host is available to receive visitors.
3. The system should enable the authorized building management personnel to input visitor’s details in the form as per the visitors and visit data dictionary as per described above.
4. The system should send a passcode to the visitor’s phone.
5. The system should then assign a card to the visitor.
6. The system should save the visitors details in the system.
7. The system should then send notification to both host and building management personnel about the successful check-in of a visitor.

**Output**

Upon successful validation of the visitor details, the system will create a visitor record

1. **Pre-registered Visitor**

**Flow of Events**



1. The system should then verify if the host is available to receive visitors.
2. The system should enable the authorized building management personnel to if the id number provided by the visitor matches with the one in the pre-registered in the system
3. The system should then assign a card to the visitor.
4. The system should save the visitors details in the system.
5. The system should then send notification to both host and building management personnel about the successful check-in of a visitor

**Output**

Upon successful validation of the visitor details, the system will update the visitor check in time.

### 9.3.3 Host check-out Process.

The purpose of the host check-out process is to systematically record and manage the departure of visitors associated with a host from a specific office within a building. This process ensures a structured approach to visitor management and enhances security by confirming the departure of visitors from the office.

**Participating Actors**

**Host User**:

Initiates the check-out process and confirms the visitor's departure from the office.

**Visitor**:

Associated with a specific host user, and their check-out is confirmed by the host.

**Preconditions**

* Visitor is ready to leave the location.
* System is operational.
* Authorized personnel is logged in.

**Flow of Events**



1. The process begins when a host user decides to check out the visitor from the office.
2. The system requires the host user to scan the physical card to confirm the visitor's check-out.
3. The system should then send notification to both host and building management personnel about the successful check-out of a visitor from the office

**Output**

The system should record the departure time of the visitor from the office

### 9.3.4 Building check-out Process

**Preconditions**

* Visitor is ready to leave the location.
* System is operational.
* Authorized personnel is logged in.

**Participating Actors**

**Visitor:**

Individuals who have completed their visit and are leaving the building.

**Authorized Building Management Personnel:**

Checkout the visitor from the building

**Flow of Events**

1. Upon successful return of the physical card, the system proceeds to update the assigned status
2. A record of the visitor's check-out is stored in the system, including details such as the time and date.
3. The system should then send a notification to both host and building management personnel about the successful check-out of a visitor from the building

**Output**

The system should record the departure time of the visitor from the building.