

**Requirement & Design Specification**

**Resident Hub System(RHS)**

**Version: 1.0**

**Group 4**

**HE180734 - Truong Minh Hoang**

**HE186662 - Nguyen Lam Hoang**

# Table of content

**[Table of content 2](#_r9oy2mq87el8)**

**[Record of Changes 7](#_pgkz2okp1oqm)**

**[I. Overview 10](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.b1hely8tnkh6)**

[1. User Requirements 10](#_8digyqu41akm)

[1.1 Actors 10](#_ctq4c4kcgaba)

[1.2 Use Cases 10](#_jqxi7joal0b)

a. Diagram(s) 10

b. Descriptions 13

[2. Overall Functionalities 15](#_pgkz2okp1oqm)

[2.1 Screens Flow 15](#_7hh2stx90wng)

[2.2 Screen Descriptions 20](#_7q6rw18my1ei)

[2.3 Screen Authorization](#_phchw11wm6bt) [24](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.p5srnn12erj)

[3. System High Level Design 24](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.adbblm56siyp)

[3.1 Database Design 24](#_ctvnqvi2795p)

a. Database Schema 25

[b. Table Descriptions](#_iy31a7w9ie9c)

**[II. Requirement Specifications 27](#_pmtajeouz2pl)**

[1. Common Functions 27](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.xlx9594ze8xb)

[1.1 UC1\_Login 27](#_wt3oqb179moh)

[1.2 UC2\_Sample Manage 28](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.aux471lfzqcw)

[1.3 UC3\_Vehicle Manage](#_nf1ojnml6s0d)

1.4 UC4\_ABC Manage

1.5 UC5\_Staff Manager

1.6 UC6\_Manage User

1.7 UC7\_Manage Notification [28](#_nf1ojnml6s0d)

**[III. Design Specifications 52](#_3nqbbzhbnvmh)**

Citizen

Traffic Police

Admin

**[IV. Appendix 92](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.47qqxd8838t3)**

[1. Assumptions & Dependencies 92](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.9dexybrcuq58)

[2. Limitations & Exclusions 92](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.ngz2woyutcj6)

[3. Business Rules 92](https://docs.google.com/document/d/177bhh4dR9YhgyBvfdosf-nZCsaGKIpD8eDWOCqDoKGw/edit#heading=h.x5pagdt4e93s)

# I. Overview

The Traffic Violation Reporting Management Application aims to create a platform that allows citizens to report traffic violations quickly and transparently. The system supports different types of users, including citizens, traffic violators, and traffic police. The application helps raise awareness of traffic law compliance and assists authorities in handling violations more effectively.

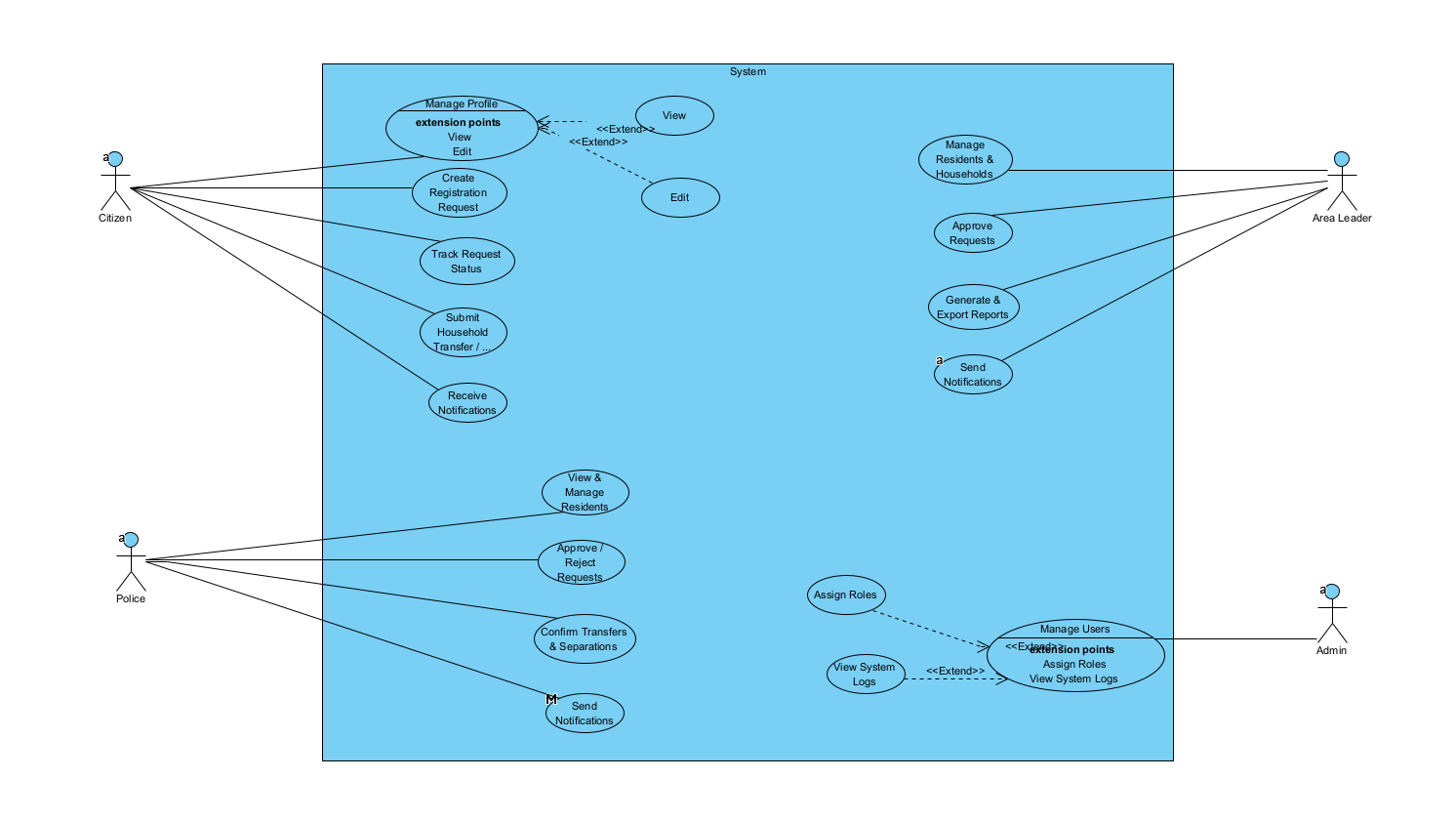
## 1. User Requirements

### 1.1 Actors

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Citizen | Citizens can log in, manage their profiles, submit registration requests, upload documents, track request status, and receive system notifications. |
| 2 | Area Leader | Area leaders oversee residents, review and approve registration requests, confirm household transfers and separations, and send notifications. |
| 3 | Police | Police officers manage resident and household data, approve requests, generate reports, and send official notifications. |
| 4 | System Administrator | Admins manage user accounts, assign roles, configure system settings, and ensure security. |

### 1.2 Use Cases

#### *a. Diagram(s)*

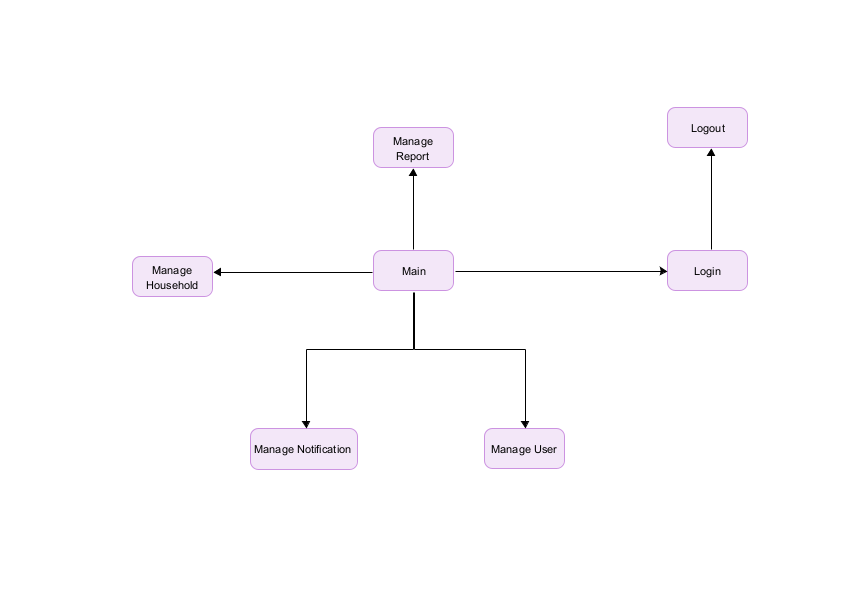


#### *b. Descriptions*

| **ID** | **Feature** | **Use Case** | **Use Case Description** |
| --- | --- | --- | --- |
| 01 | Citizen | Manage Profile | View and update personal information |
| 02 | Create Registration Request | Submit a new citizen registration request |
| 03 | Upload Documents | Attach and submit required documents |
| 04 | Track Request Status | View progress of submitted requests |
| 05 | Submit Household Transfer | Request to transfer household registration |
| 06 |  | Submit Household Separation | Request to separate from a household |
| 07 |  | Receive Notifications | Get updates and alerts from the system |
| 08 | Area Leader | View & Manage Residents | See and manage citizen records |
| 09 | Approve / Reject Requests | Review and decide on registration requests |
| 10 |  | Confirm Transfers & Separations | Validate household changes |
| 11 |  | Send Notifications | Notify citizens about important updates |
| 12 | Police | Manage Residents & Households | Add, edit, or remove resident and household data |
| 13 | Approve Requests | Approve or reject citizen registration requests |
| 14 | Generate & Export Reports | Create and download statistical reports |
| 15 |  | Send Notifications | Notify citizens about approvals and regulations |
| 16 | Admin | Manage Users | View and manage system user accounts |
| 17 | Assign Roles | Set user roles and permissions them |
| 18 |  | View System Logs | Monitor system activities and logs |

## 2. Overall Functionalities

### 2.1 Screens Flow



### 2.2 Screen Descriptions

### 

### 

| **#** | **Feature** | **Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | Login | Login | Allows users to log in using email/username and password. Role-based access control (Citizen, AreaLeader, Police). |
| 2 | Logout | Logout | Ends the current user session. |
| 3 | View Personal Information | Account Management | Displays user details (Full Name, Email, Address, Role). |
| 4 | Update Personal Information | Account Management | Allows users to update their Full Name, Email, and Address. Role cannot be changed. |
| 5 | Create Registration Request | Registration | Users submit registration requests with details like type, start date, and end date. Default status is "Pending". |
| 6 | Upload Documents | Registration | Users can upload files (ID card, proof of residence) and store details in the "Comments" field. |
| 7 | Track Request Status | Registration | Displays a list of submitted requests with status updates. |
| 8 | Submit Household Transfer Request | Registration | Users can request household transfers, which require approval from authorities. |
| 9 | Submit Household Separation Request | Registration | Users request household separation, specifying members to be separated. |
| 10 | Receive Notifications | Notifications | Users can view system notifications and mark them as read. |
| 11 | View List of Residents | Resident Management | Area Leaders and Police can see a list of residents filtered by area. |
| 12 | View List of Households | Household Management | Displays household details and members. |
| 13 | |  | | --- |  | View List of Registration Requests | | --- | | Request Management | Authorities filter and review registration requests within their area. |
| 14 | Approve/Reject Requests | Request Management | Authorities update request status (Approved/Rejected) and add comments. Notifications are sent to users. |
| 15 | Confirm Household Transfer | Request Management | Authorities approve or reject household transfer requests. |
| 16 | Confirm Household Separation | Request Management | Authorities verify and update household information upon approval. |
| 17 | Send Manual Notifications | Notifications | Area Leaders can manually send notifications to users or police. |
| 18 | Add/Update/Delete Resident Information | Resident Management | Police can perform CRUD operations on residents. |
| 19 | Add/Update/Delete Households | Household Management | Police can create, update, or remove household records. |
| 20 | View Resident Report | Reports | Generates statistics on residents by type (Permanent, Temporary, Temporary Stay). |
| 21 | View Household Report | Reports | Generates statistics on households and members. |
| 22 | Export Reports | Reports | Exports data to PDF/Excel for administrative use. |
| 23 | Send Notifications | Notifications | Police can create notifications for users and area leaders. |
| 24 | User Account Management | Admin Panel | Admins can add, edit, and delete user accounts. |
| 25 | View System Logs | Admin Panel | Displays system activity logs for monitoring. |
| 26 | System Configuration | Admin Panel | Allows setting system parameters (e.g., request limits, notification formats). |

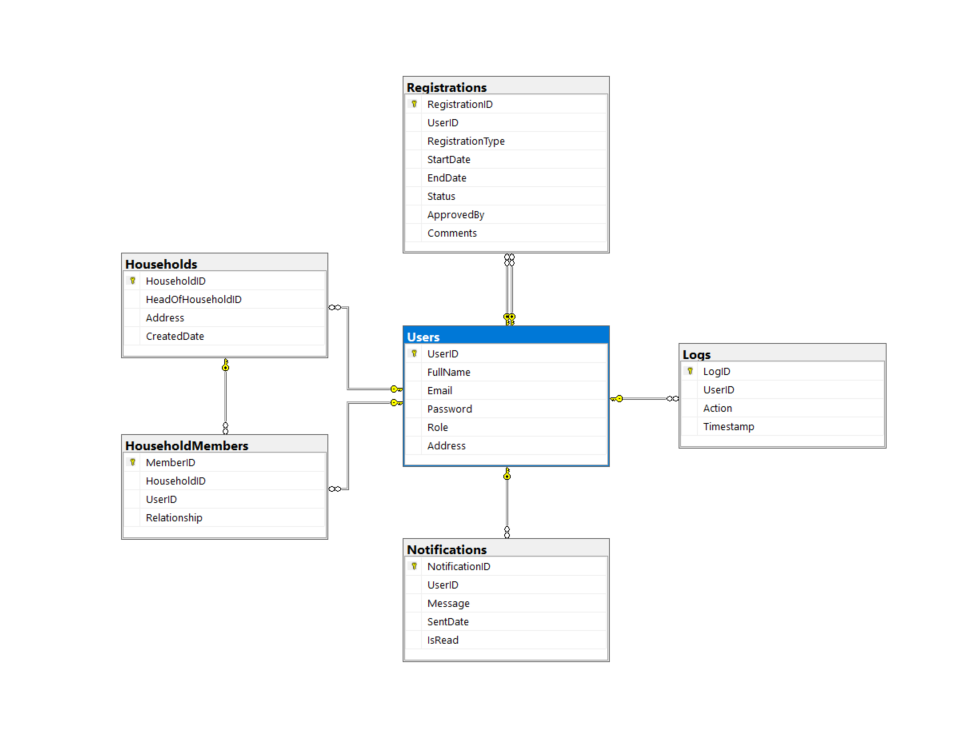
### 2.3 Screen Authorization

| **#** | **Screen** | **Admin** | **Area Leader** | **Traffic Police** | **Citizen** |
| --- | --- | --- | --- | --- | --- |
| 1 | Login / Register |  |  |  | X |
| 2 | Home | X | X | X | X |
| 3 | View Notifications |  | X | X | X |
| 4 | View & Update Personal Info |  |  |  | X |
| 5 | Create Registration Request |  |  |  | X |
| 6 | Upload Documents |  |  |  | X |
| 7 | Track Request Status |  |  |  | X |
| 8 | Submit Household Transfer Request |  |  |  | X |
| 9 | Submit Household Separation Request |  |  |  | X |
| 10 | View & Manage Residents |  | X | X |  |
| 11 | View & Manage Households |  | X | X |  |
| 12 | View & Manage Registration Requests |  | X | X |  |
| 13 | Approve/Reject Requests |  | X | X |  |
| 14 | Confirm Household Updates |  | X | X |  |
| 15 | Manage Reports & Export Data | X |  | X |  |
| 16 | Send Notifications | X | X | X |  |
| 17 | User Account Management | X |  |  |  |
| 18 | System Configuration | X |  |  |  |

### 3. System High Level Design

### 3.1 Database Design

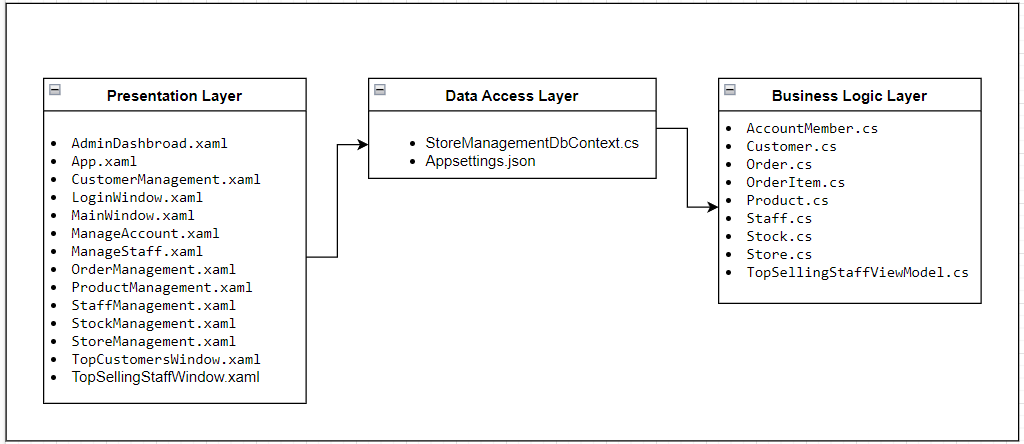
#### *Database Schema*



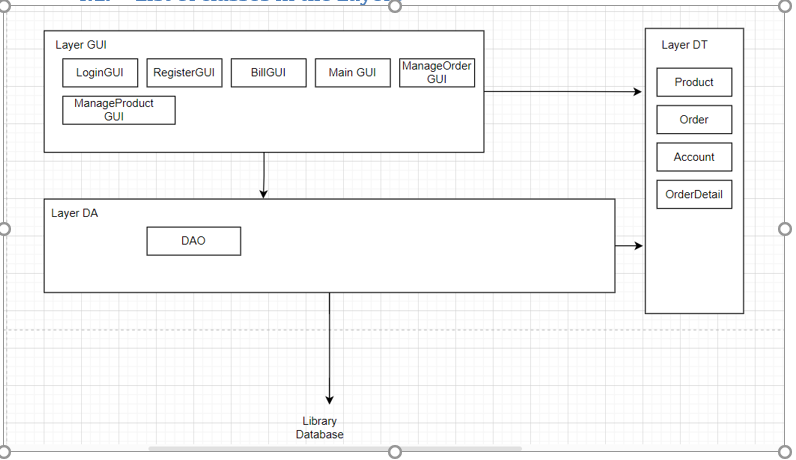
#### *b. Table Descriptions*

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | Users | * Manages user accounts with fields for UserID, FullName, Email, Password, Role, Address. * **Primary key:** UserID. |
| 02 | Vehicles | * Stores user registration requests with fields for RegistrationID, UserID, RegistrationType, StartDate, EndDate, Status, ApprovedBy, Comments. * **Primary key:** RegistrationID. * **Foreign key:** UserID (references UserID in Users). |
| 03 | Households | * Stores household information with fields for HouseholdID, HeadOfHouseholdID, Address, CreatedDate. * **Primary key:** HouseholdID. * **Foreign key:** HeadOfHouseholdID (references UserID in Users). |
| 04 | HouseholdMembers | * Manages household members with fields for MemberID, HouseholdID, UserID, Relationship. * **Primary key:** MemberID. * **Foreign keys:** HouseholdID (references HouseholdID in Households), UserID (references UserID in Users). |
| 05 | Notifications | * Stores notifications with fields for NotificationID, UserID, Message, SentDate, IsRead. * **Primary key:** NotificationID. * **Foreign key:** UserID (references UserID in Users). |
| 06 | Logs | * Stores system logs with fields for LogID, UserID, Action, Timestamp. * **Primary key:** LogID. * **Foreign key:** UserID (references UserID in Users). |

4. Class diagram,

*4.1. *

*4.2. Class diagram*

**

# II. Requirement Specifications (OPTIONAL)

## 1.1 UC1\_ Login

**Functional Description Template**

| ID and Name: | UC-1\_ User Login | | |
| --- | --- | --- | --- |
| Created By: | HE180734  Truong Minh Hoang | Date Created: 3/25/2025 |  |
| Primary Actor: | User (Citizen, AreaLeader, Police) | Secondary Actors: |  |
| Description: | This use case describes the authentication process for users by verifying their email and password. | | |
| Trigger: | User attempts to log in. | | |
| Preconditions: | PRE-1: User has an existing account. | | |
| Postconditions: | * POST-1: If authentication is successful, the user is logged into the system and granted access based on their role. * POST-2: If authentication fails, the system displays an error message. | | |
| Normal Flow: | 1. The user enters their Email and Password. 2. The system verifies the credentials against the database. 3. If the credentials are valid, the system logs the user in and redirects them to the appropriate dashboard based on their role. 4. The system records the login action in the activity log. 5. If the credentials are invalid, the system displays an error message and prompts the user to retry. | | |
| Alternative Flows: | AF-1: If the user clicks "Forgot Password", they are redirected to the Password Reset page.  AF-2: If the user enters incorrect credentials three times, the system temporarily locks the account and sends a notification email. | | |
| Exceptions: | EX-1: Database connection error prevents authentication.  EX-2: Unexpected system failure occurs during login. | | |
| Priority: | High | | |
| Frequency of Use: | Frequently used by all users. | | |
| Business Rules: | BR-1: Users can only access functionalities based on their assigned role (Citizen, AreaLeader, Police).  BR-2: Default admin credentials should not be used in production.  BR-3: Passwords must meet security requirements (e.g., minimum length, complexity). | | |
| Other Information: | All successful and failed login attempts are recorded in the system logs. | | |
| Assumptions: | Users have registered and activated their accounts before attempting to log in. | | |

1.2 UC2\_Register

a. Functional Description

| ID and Name: | UC-2\_ Register | | |
| --- | --- | --- | --- |
| Created By: | HE180734  Truong Minh Hoang | Date Created: | 25-03-2025 |
| Primary Actor: | Citizen | Secondary Actors: | None |
| Description: | A Citizen registers for an account to access additional features. | | |
| Trigger: | Citizen clicks on the "Register" button in the Login Screen | | |
| Preconditions: | The Citizen does not already have an account with the provided email address. | | |
| Postconditions: | The Citizen successfully creates an account and becomes a registered user | | |
| Normal Flow: | 1.Citizen clicks on the "Register" button.  2.The system displays a registration form, requesting information such as name, email address, password, and contact details.  3.The Citizen fills out the form and submits it.  4.The system validates the input data (e.g., checks for a valid email format and ensures the email is not already in use).  5.The system creates a new account for the Citizen and stores the information in the database. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | Usually | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

## 

1.3 **UC-3\_Online Registration Management**

a. Functional Description

| ID and Name: | UC-2\_ **Online Registration Management** | | |
| --- | --- | --- | --- |
| Created By: | HE180734  Truong Minh Hoang | Date Created: | 25-03-2025 |
| Primary Actor: | User (Citizen) | Secondary Actors: | AreaLeader, Police |
| Description: | This use case describes the process for users to create and manage registration requests related to residence matters. | | |
| Trigger: | User initiates a new registration request or tracks an existing request. | | |
| Preconditions: | PRE-1: User has a valid account and is logged into the system.  PRE-2: User has the necessary information and documents for the registration request.. | | |
| Postconditions: | POST-1: Registration request is created and stored in the system with a default status of "Pending".  POST-2: Uploaded documents are linked to the corresponding registration request.  POST-3: User can track the status of their request in the system.  POST-4: Requests requiring approval are sent to the appropriate authority (AreaLeader/Police). | | |
| Normal Flow: | 1.User navigates to the registration request page.  2.User fills in the required registration details (RegistrationType, StartDate, EndDate).  3.User submits the request, which is stored in the Registrations table with status "Pending".  4.User uploads relevant documents (e.g., ID card, proof of residence), which are stored in the system.  5.User can view a list of their past and current requests, including their status and comments from authorities.  6.If the request involves a household transfer, the system routes it to the AreaLeader or Police for approval.  7.If the request involves household separation, the user provides details of members to be separated, which are stored in the system. | | |
| Alternative Flows: | AF-1: If a document upload fails, the system prompts the user to retry.  AF-2: If the request is missing mandatory details, the system prevents submission and highlights the missing fields. | | |
| Exceptions: | EX-1: Database connection error prevents request submission.  EX-2: System failure occurs during document upload.  EX-3: User submits an invalid file format for upload. | | |
| Priority: | High | | |
| Frequency of Use: | Frequently used by citizens submitting registration requests. | | |
| Other Information: | The system maintains a history of all registration requests submitted by a user.  Authorities can update the status and provide comments on requests. | | |
| Assumptions: | Users have the required documents and information before starting a registration request.  AreaLeader and Police have appropriate access to review and approve requests. | | |

## 

## 1.4.**Request Verification**

a. Functional Description

| ID and Name: | UC-4\_ **Request Verification** | | |
| --- | --- | --- | --- |
| Created By: | HE180734  Truong Minh Hoang | Date Created: | 25-03-2025 |
| Primary Actor: | AreaLeader, Police | Secondary Actors: | System, Citizen |
| Description: | This use case describes the process for authorities (AreaLeader and Police) to view, approve, or reject registration requests submitted by citizens. | | |
| Trigger: | An authority (AreaLeader or Police) accesses the system to verify pending registration requests. | | |
| Preconditions: | PRE-1: Authority has a valid account and is logged into the system.  PRE-2: Authority has the required permissions to review and update requests.  PRE-3: The system has pending registration requests that need verification. | | |
| Postconditions: | POST-1: The status of a request is updated to "Approved" or "Rejected".  POST-2: A notification is sent to the citizen regarding the request decision.  POST-3: If approved, the system updates residence records accordingly. | | |
| Normal Flow: | 1. Authority navigates to the request verification page. 2. Authority filters the list of registration requests by their managed area. 3. Authority selects a request to review. 4. Authority views request details, including the citizen’s provided information and uploaded documents. 5. Authority decides to approve or reject the request. 6. If approved:    * The system updates the request status to "Approved".    * The system sends a notification to the citizen.    * If applicable, residence records are updated. 7. If rejected:    * The system updates the request status to "Rejected".    * Authority enters a rejection reason in the comments.    * The system notifies the citizen of the rejection and reason. | | |
| Alternative Flows: | AF-1: Authority wants to search for a specific request based on citizen name, registration type, or status.  AF-2: If a document is missing or unclear, the authority may request additional information before making a decision. | | |
| Exceptions: | EX-1: Database connection error prevents request retrieval or update.  EX-2: Authority tries to approve/reject a request outside their managed area.  EX-3: System fails to send a notification to the citizen. | | |
| Priority: | High | | |
| Frequency of Use: | Frequently used by authorities to review and process citizen registration requests. | | |
| Other Information: | Authorities can view past approved/rejected requests.  The system should provide a summary dashboard for authorities to track request statistics. | | |
| Assumptions: | Authorities have the required permissions and responsibilities for request verification.  Citizens provide accurate and complete information when submitting requests.  The system correctly maps requests to the appropriate authority based on the managed area. | | |

## 

## 

# III. Design Specifications

## Citizen

### 1. Citizen logs in with the designated account

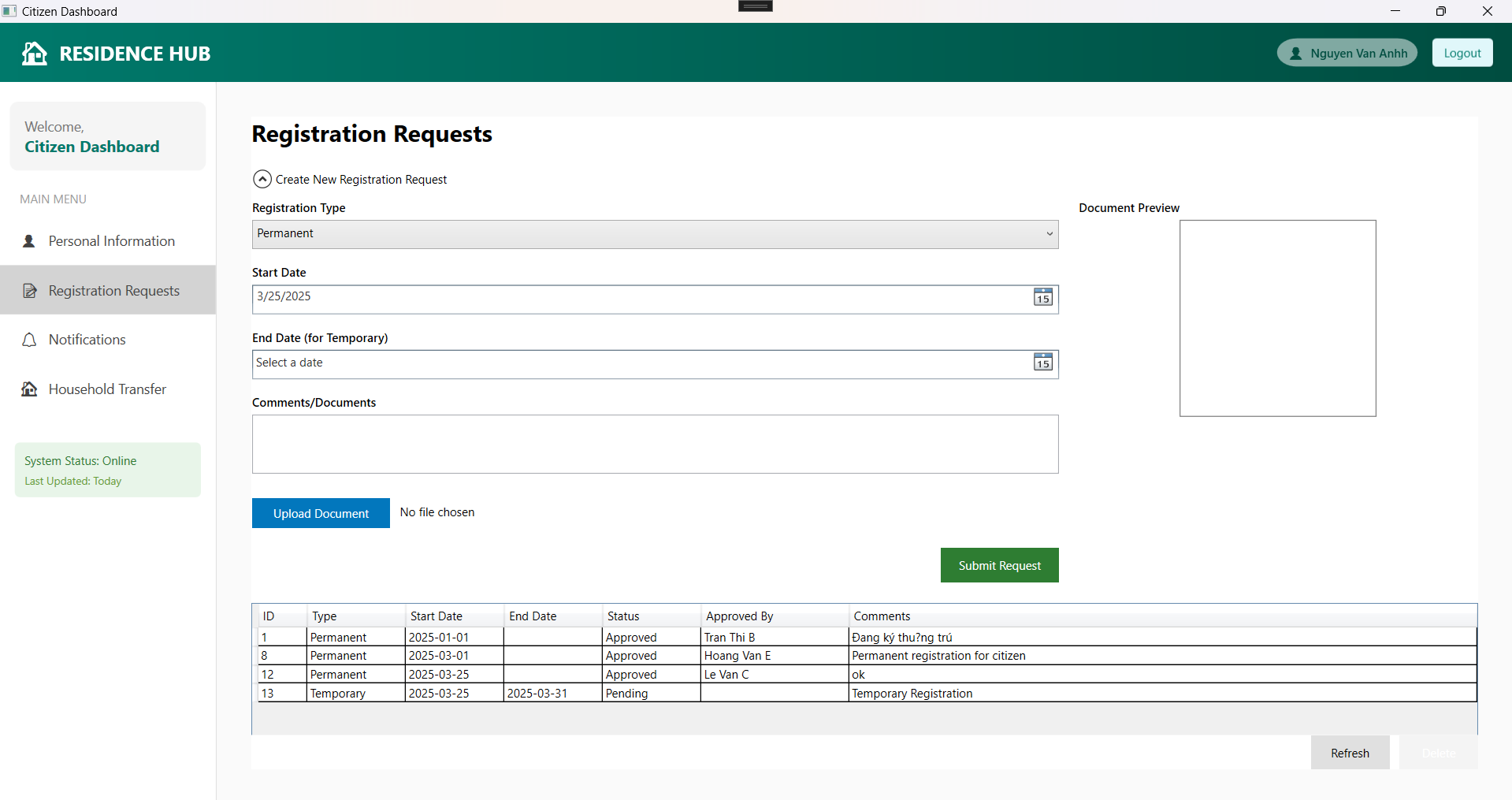
## 

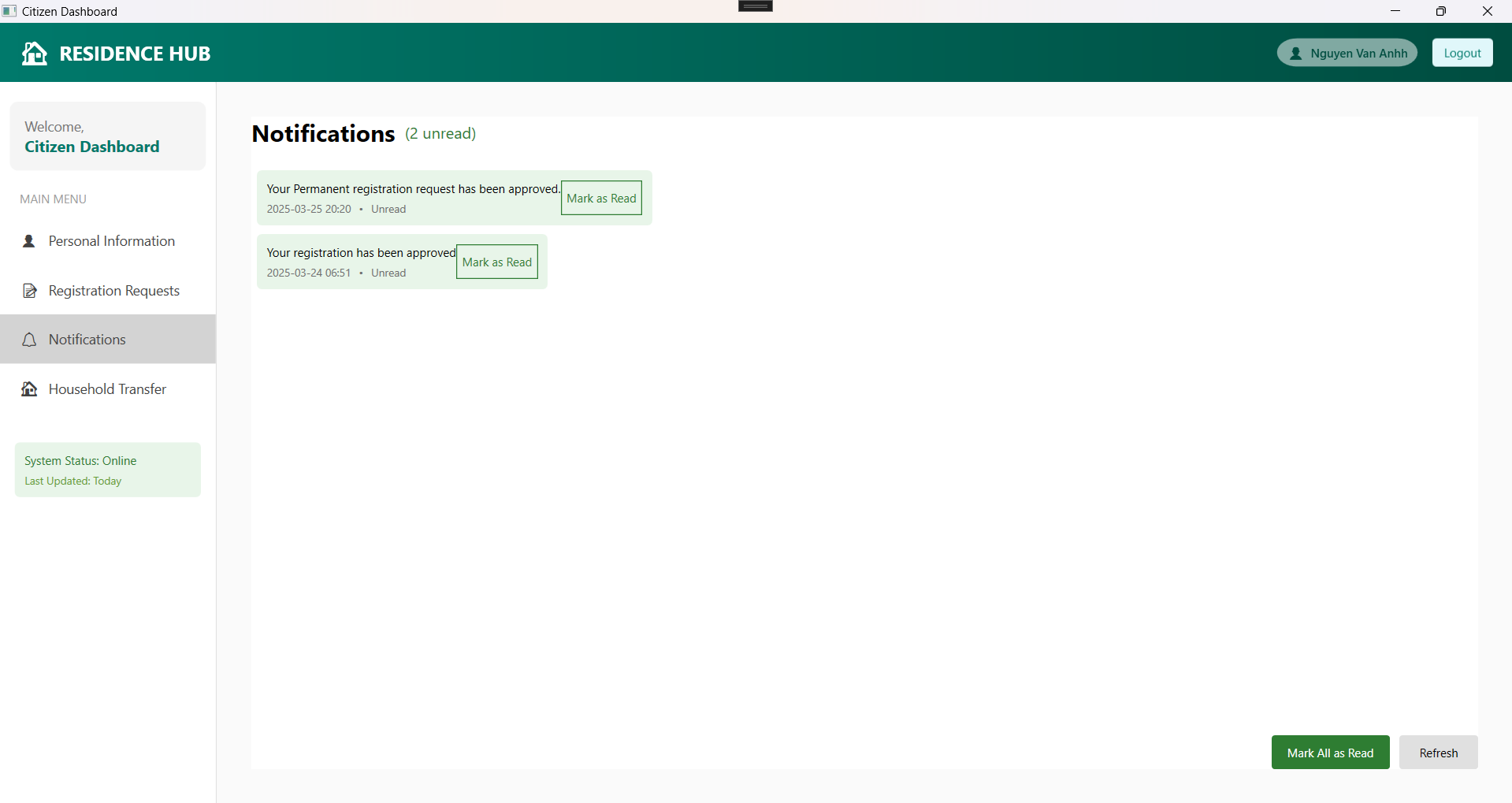
## 2. Citizen screen after logging in

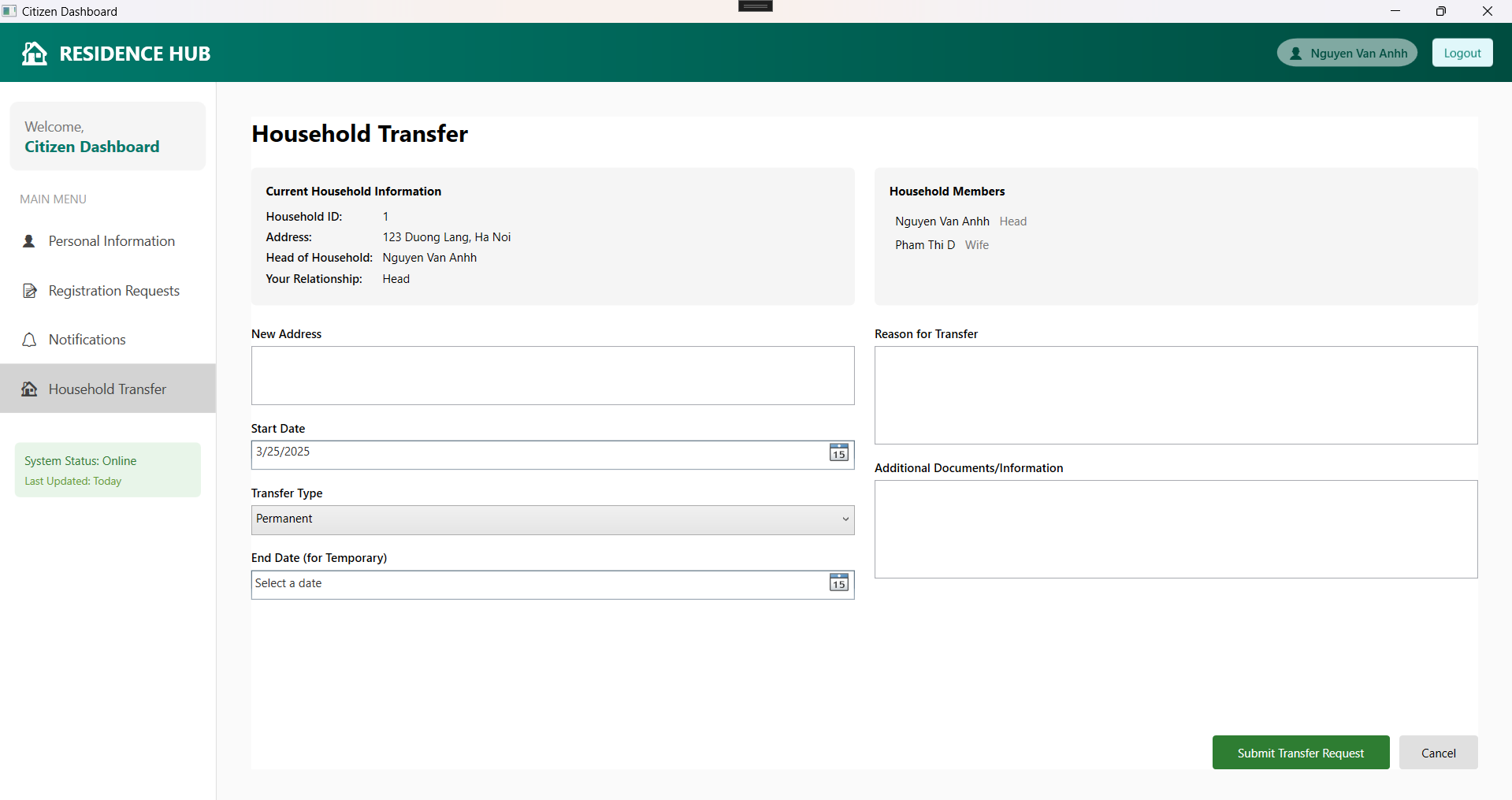
## 3. Citizen can view and edit their information.

## 

### 

4. Citizen can view their report and Registration request.  
  


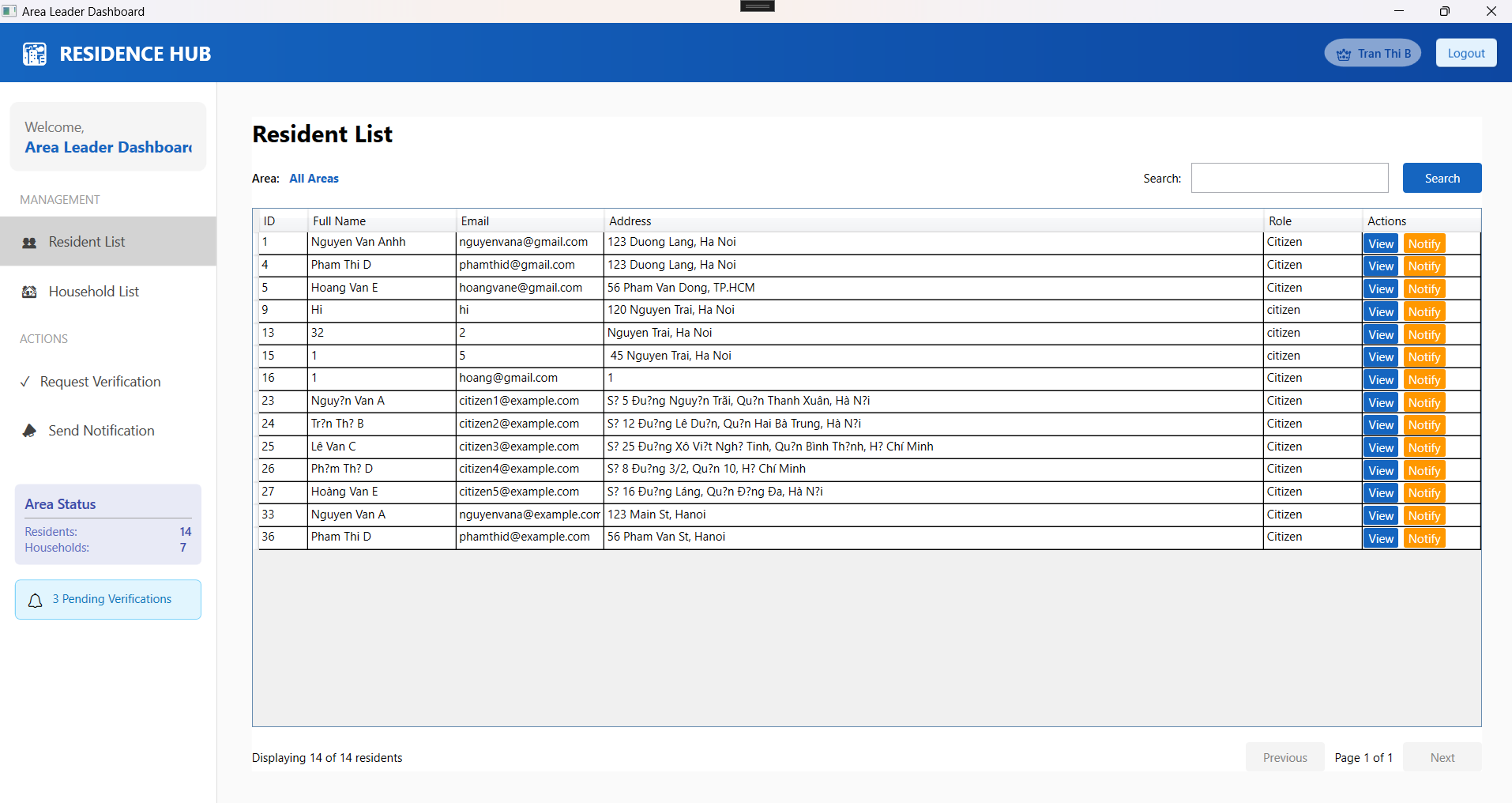
5. Citizen can receive notification.  
  


6. Citizen can submit transfer report ****

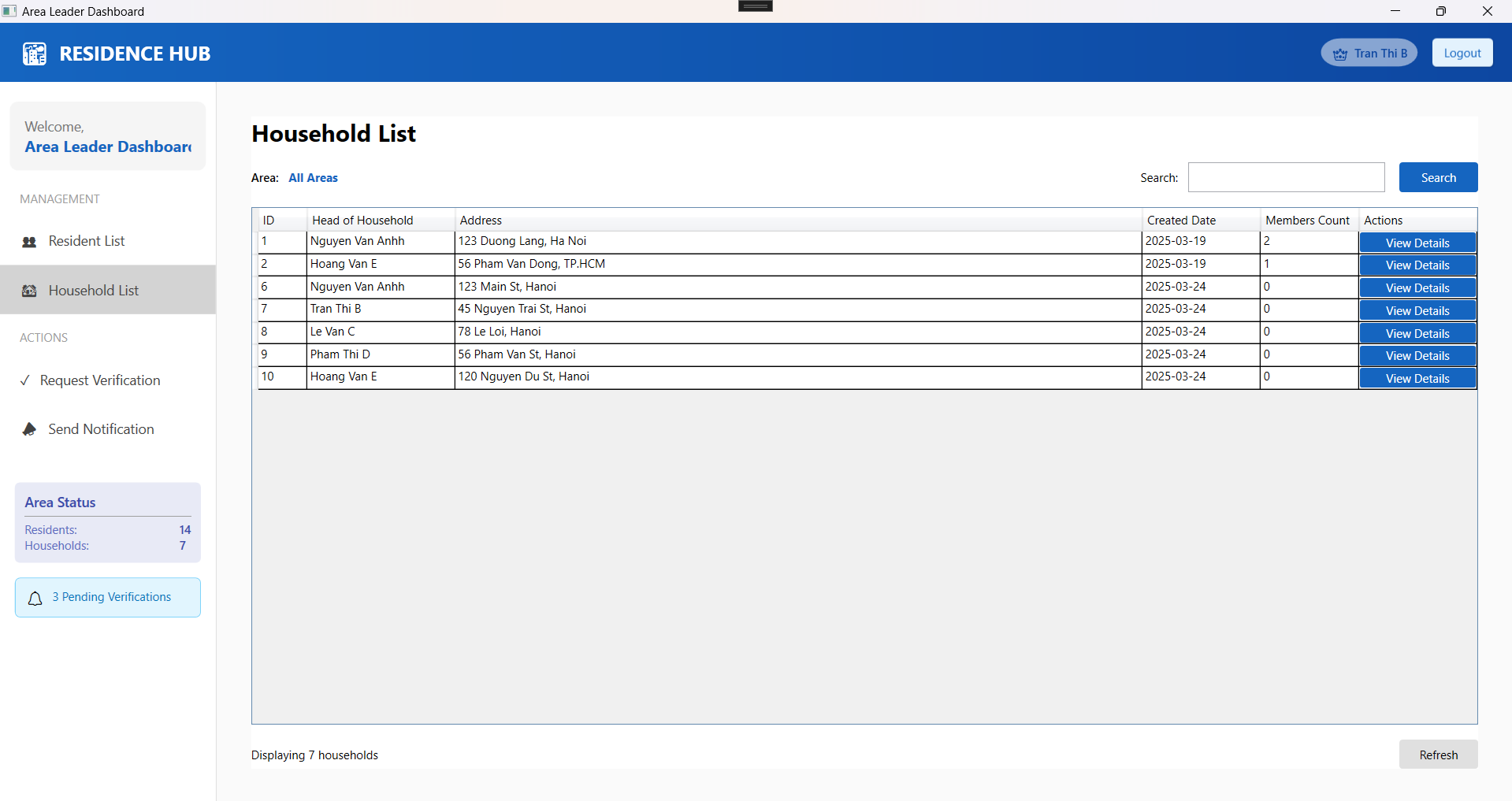
## Area Leader

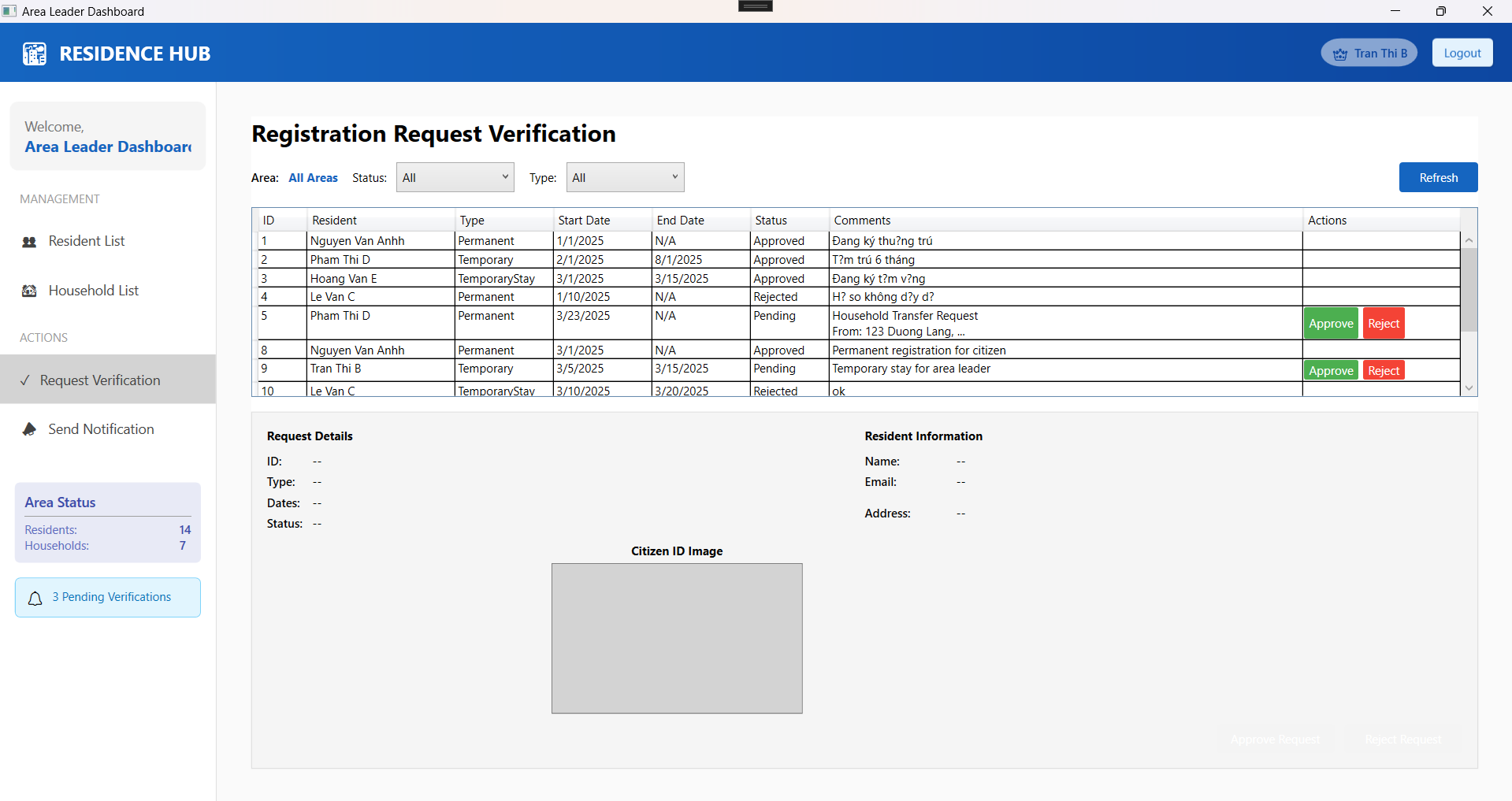
### 1. Citizen logs in with the designated account

## 2. Area Leader screen after logging in

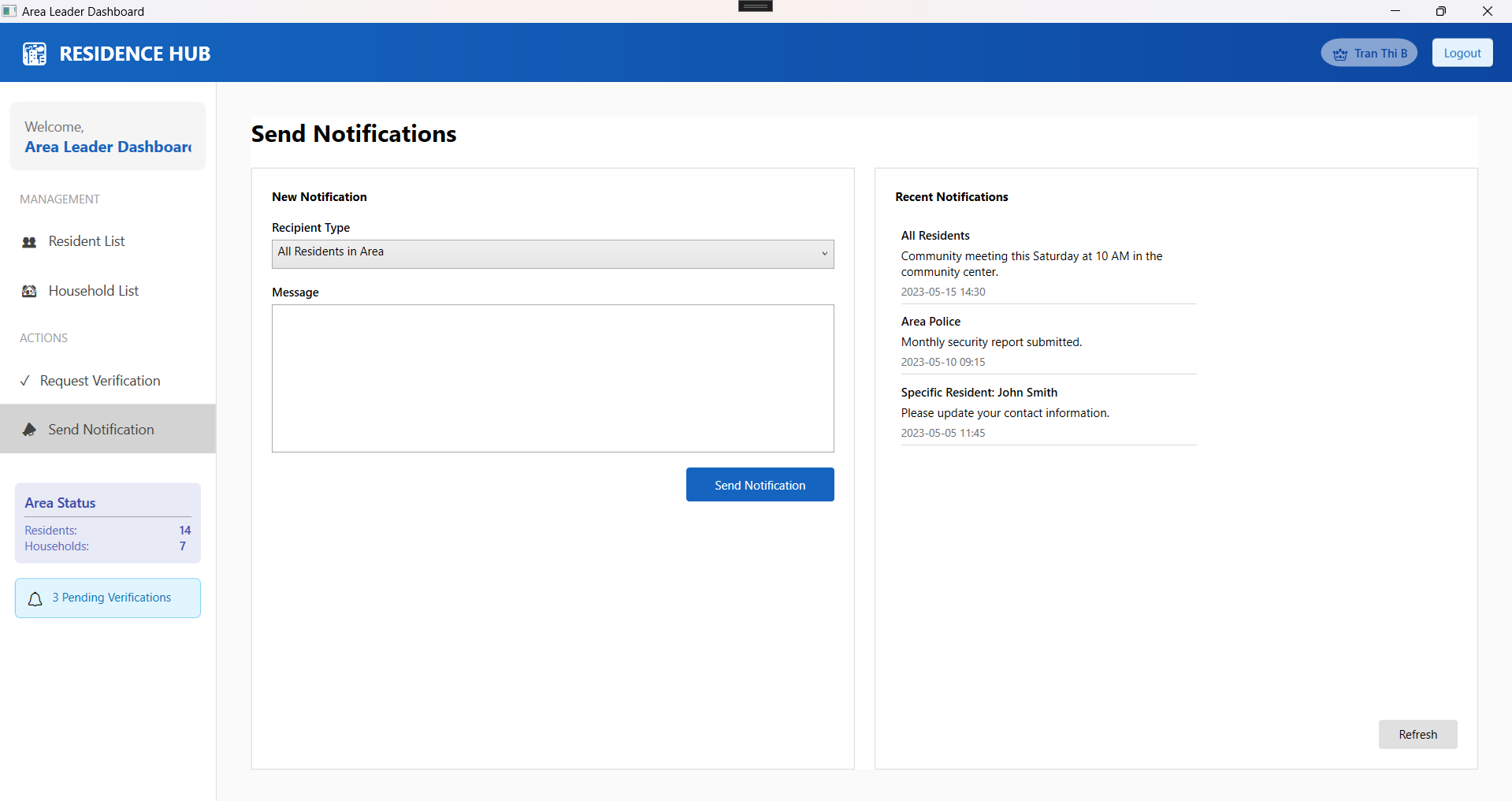
3. Area Leader can view list resident  


4. Area Leader can view list household



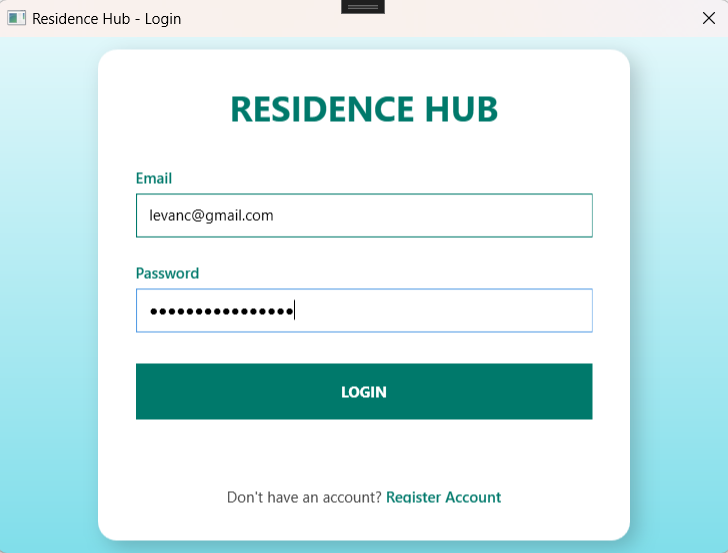
5. Area Leader can view request verification  
  


6. Area Leader can view sent verification



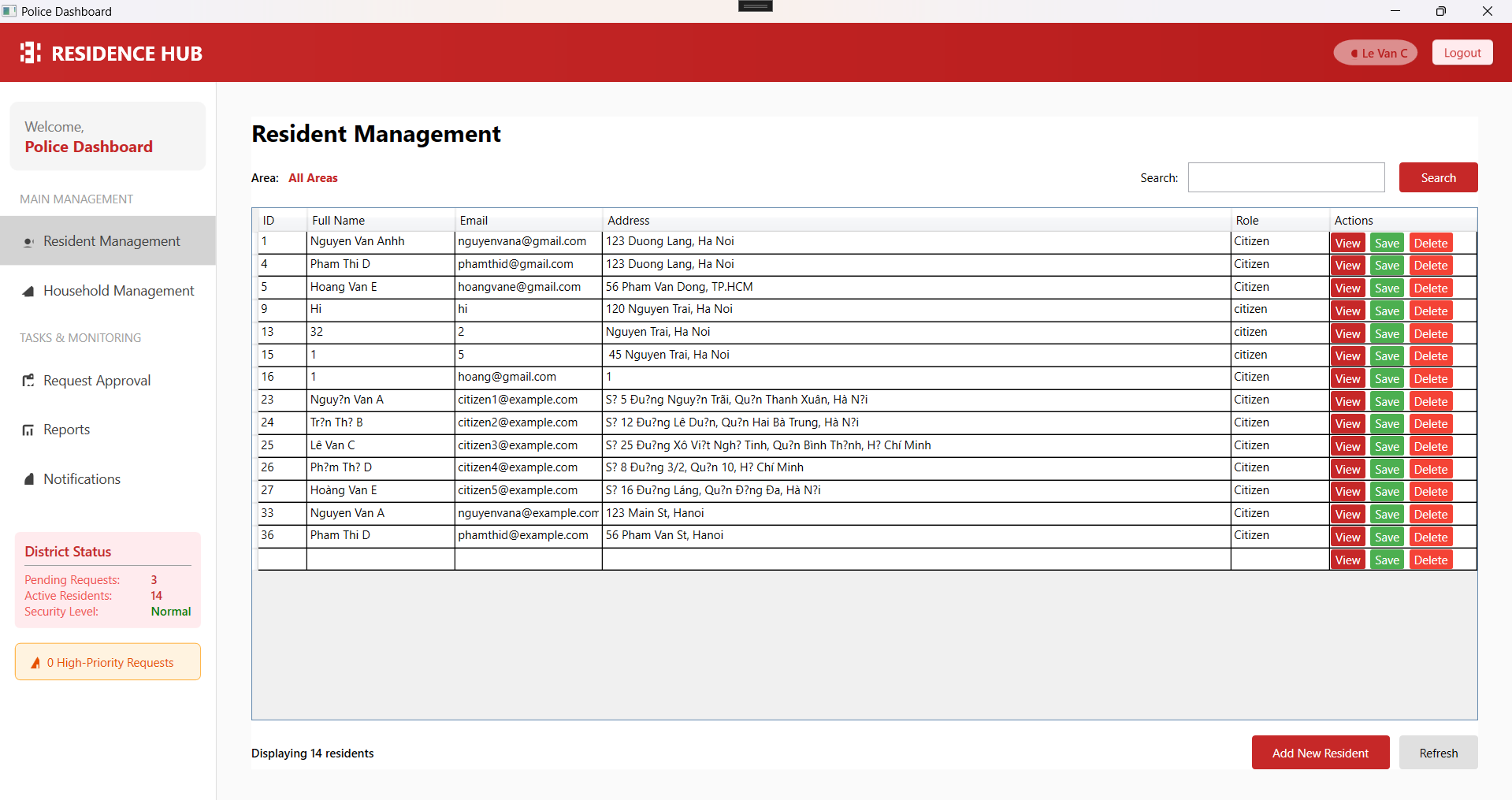
## Police

### 1. Police logs in with the designated account

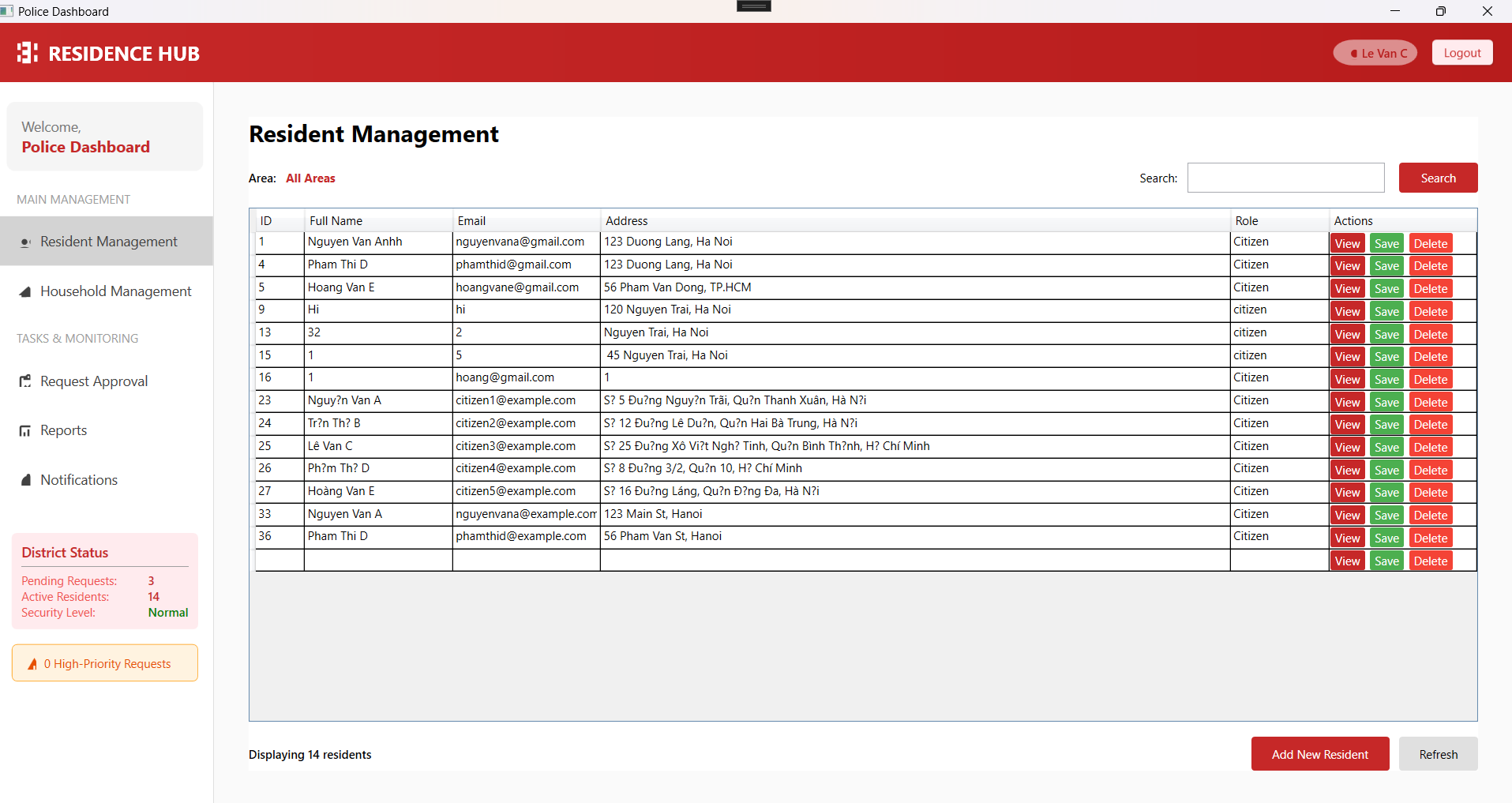
****

#### 

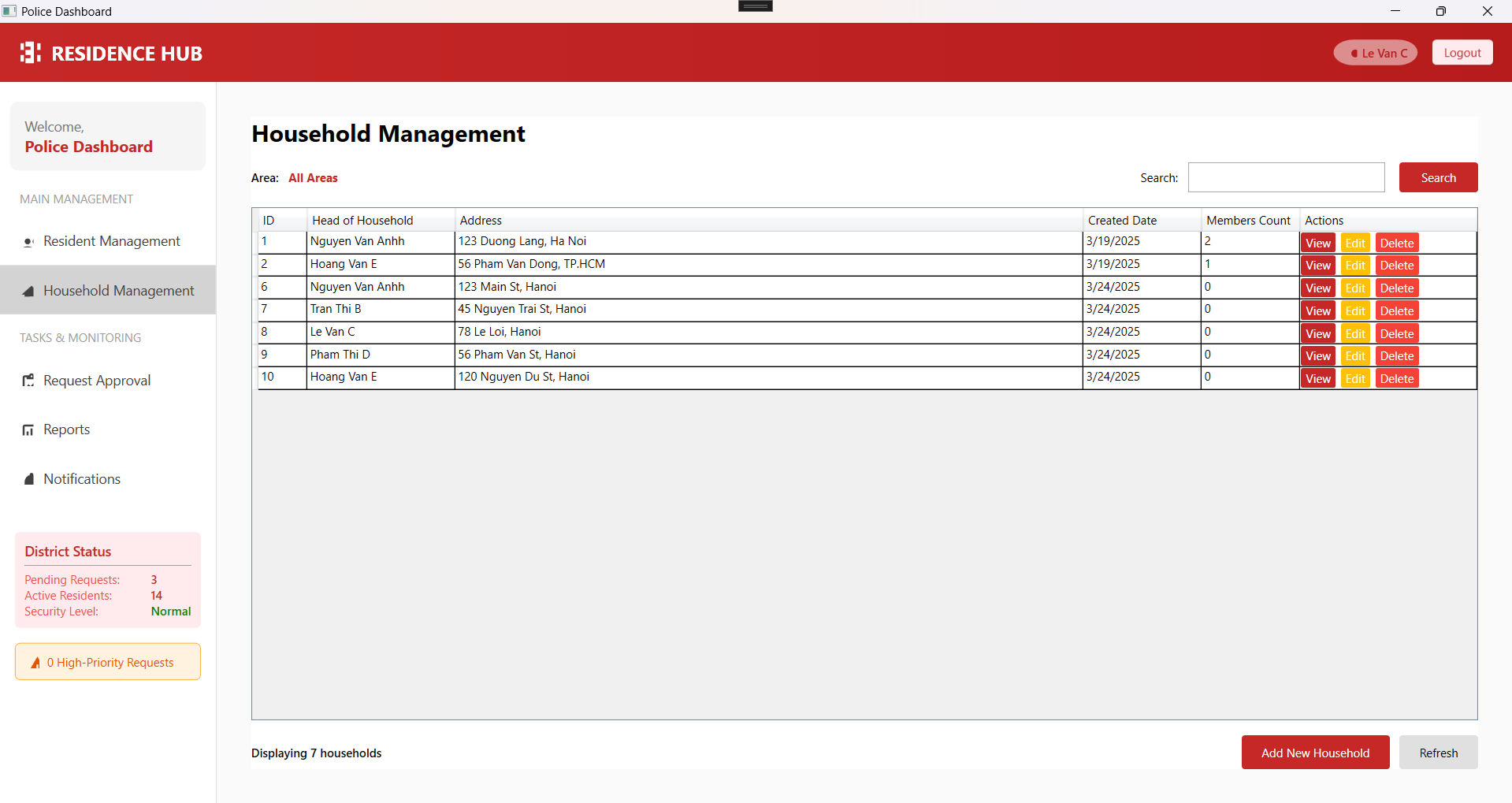
## 2. Police screen after logging in



3. Police can view and manage list resident



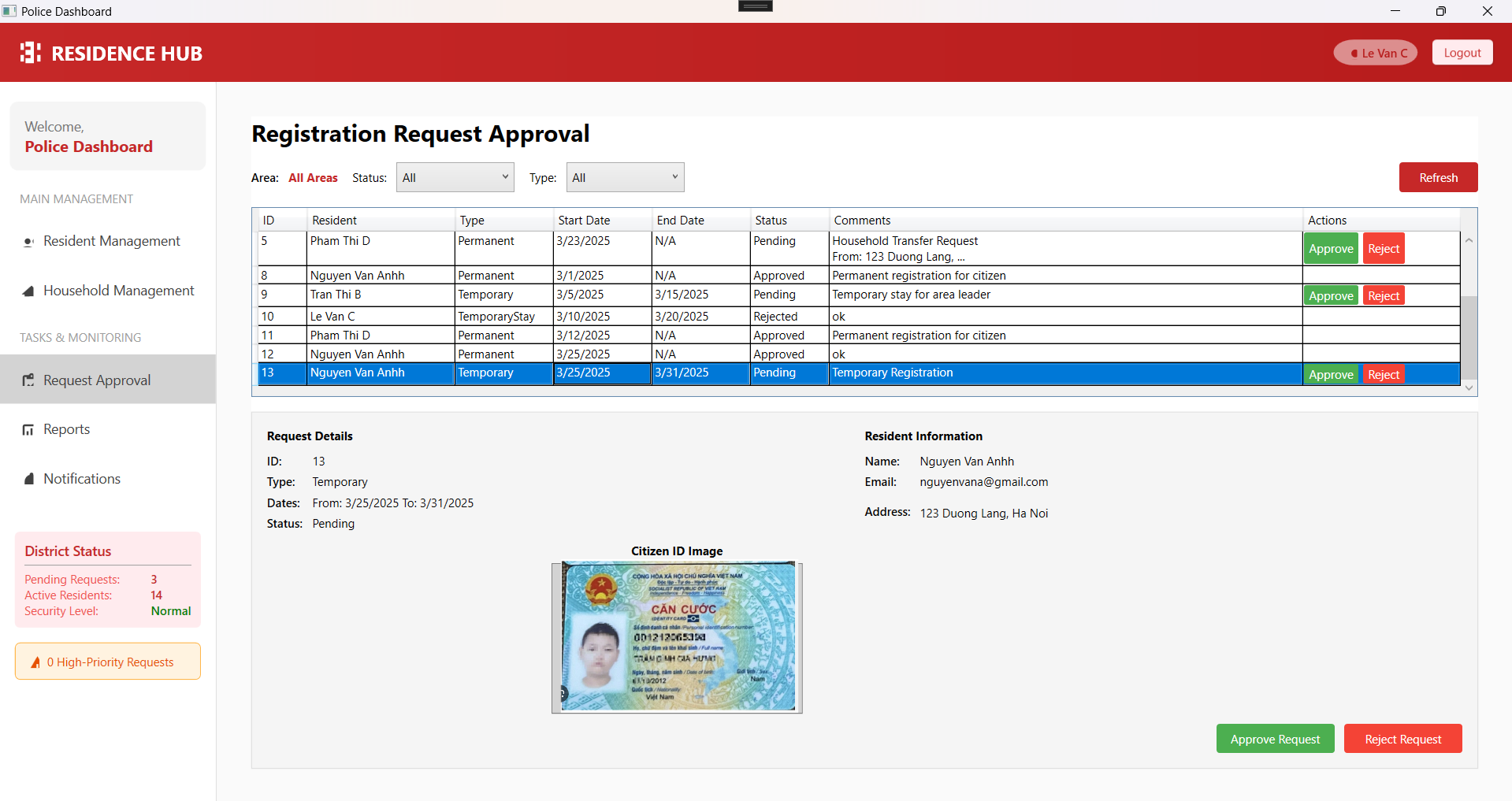
4. Police can view and manage list household



## 

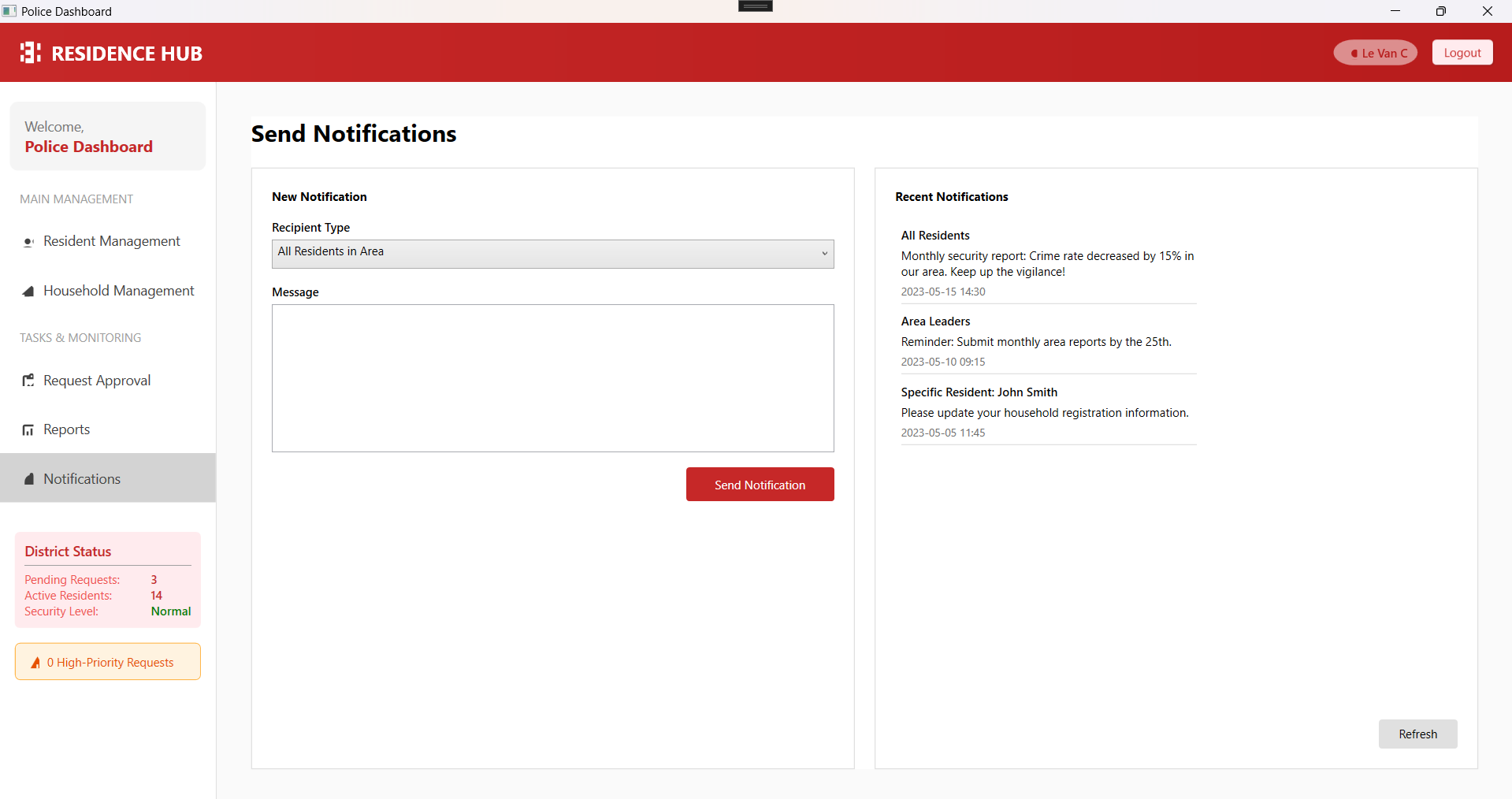
## 

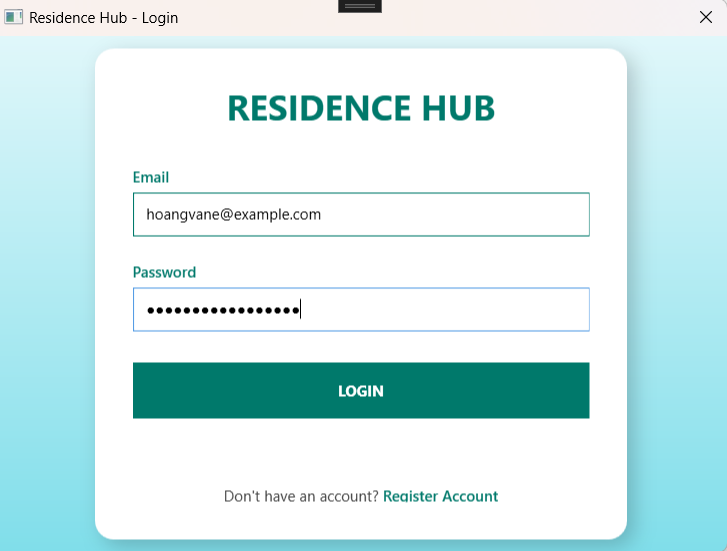
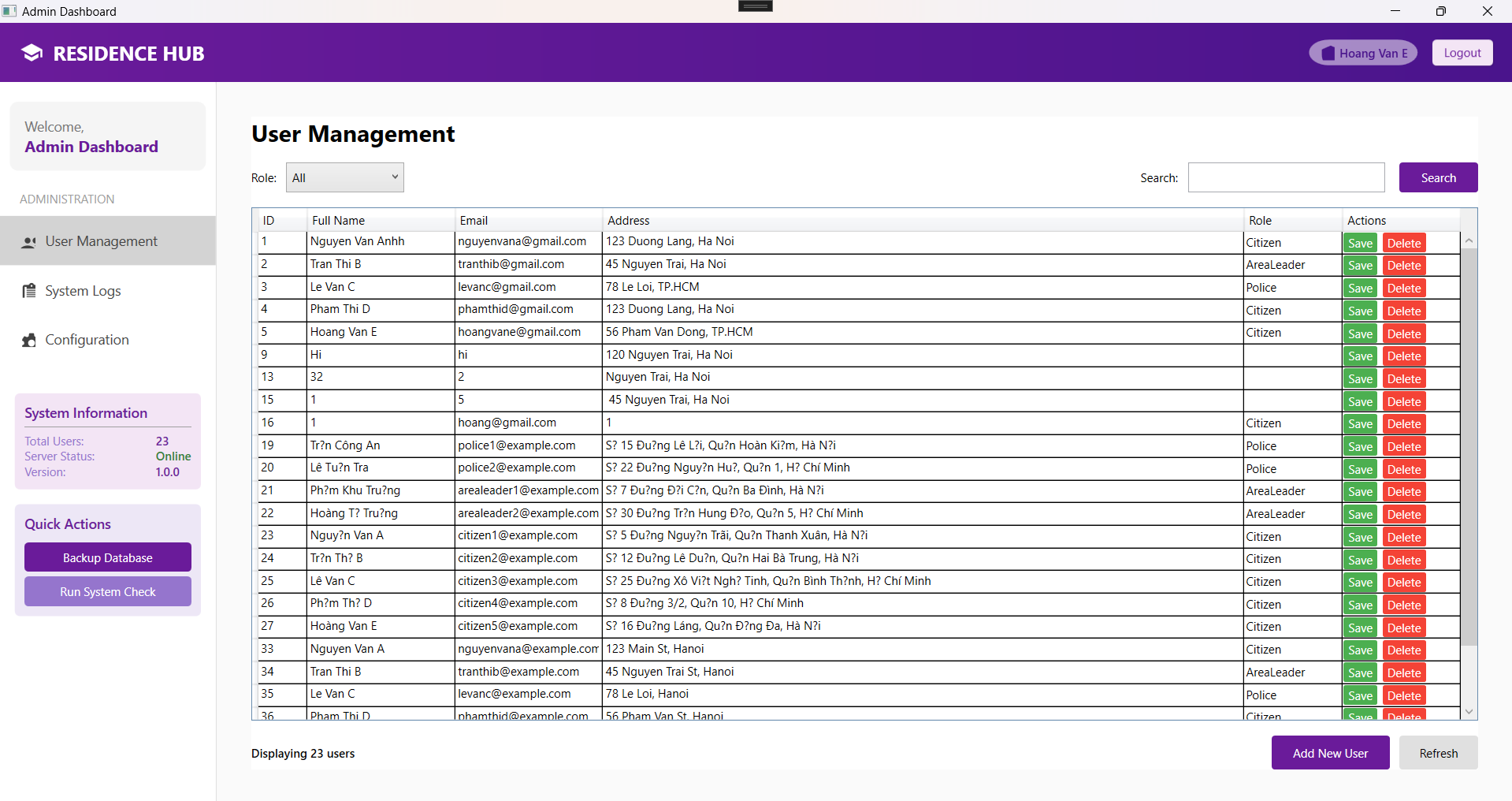
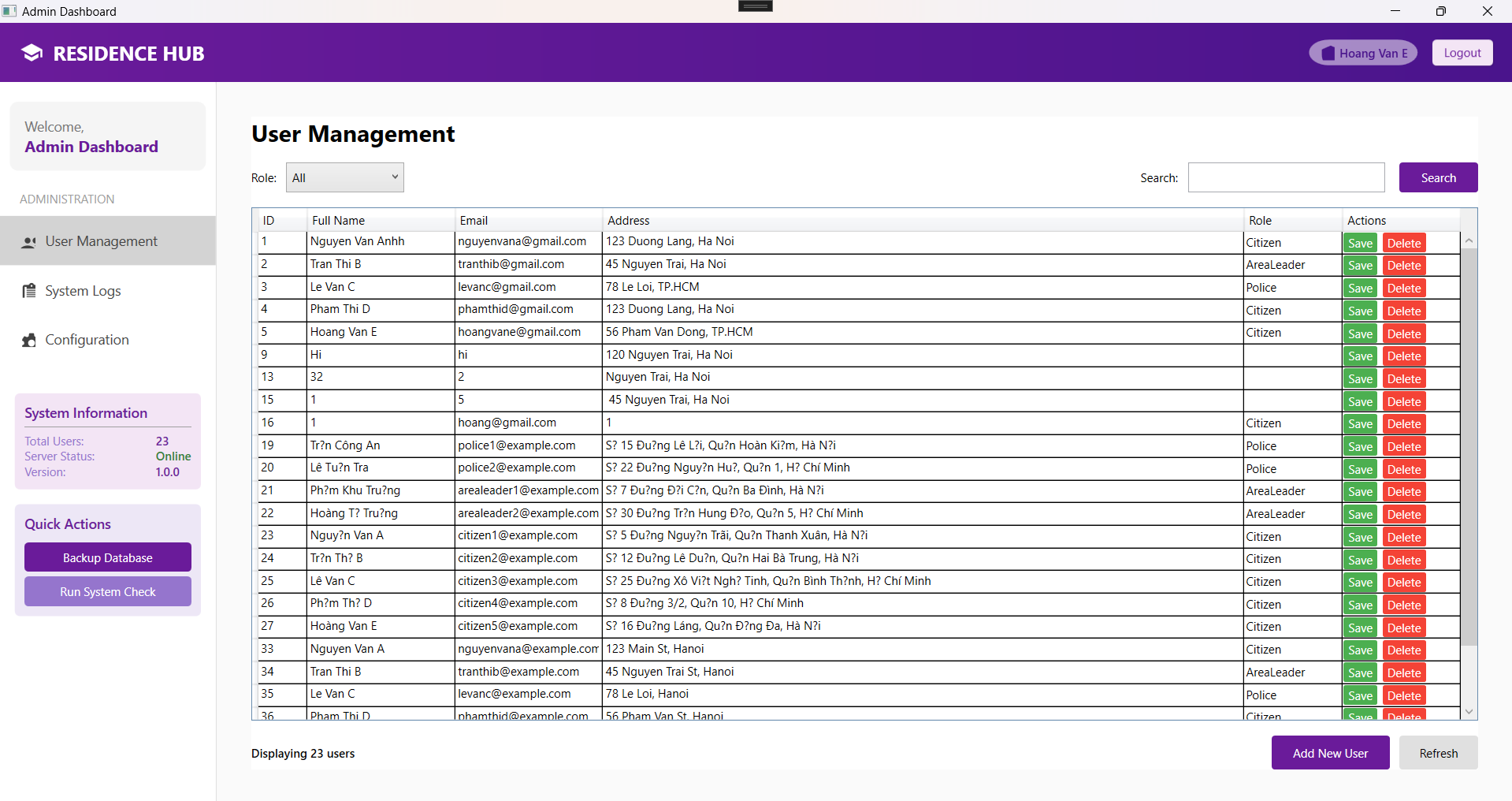
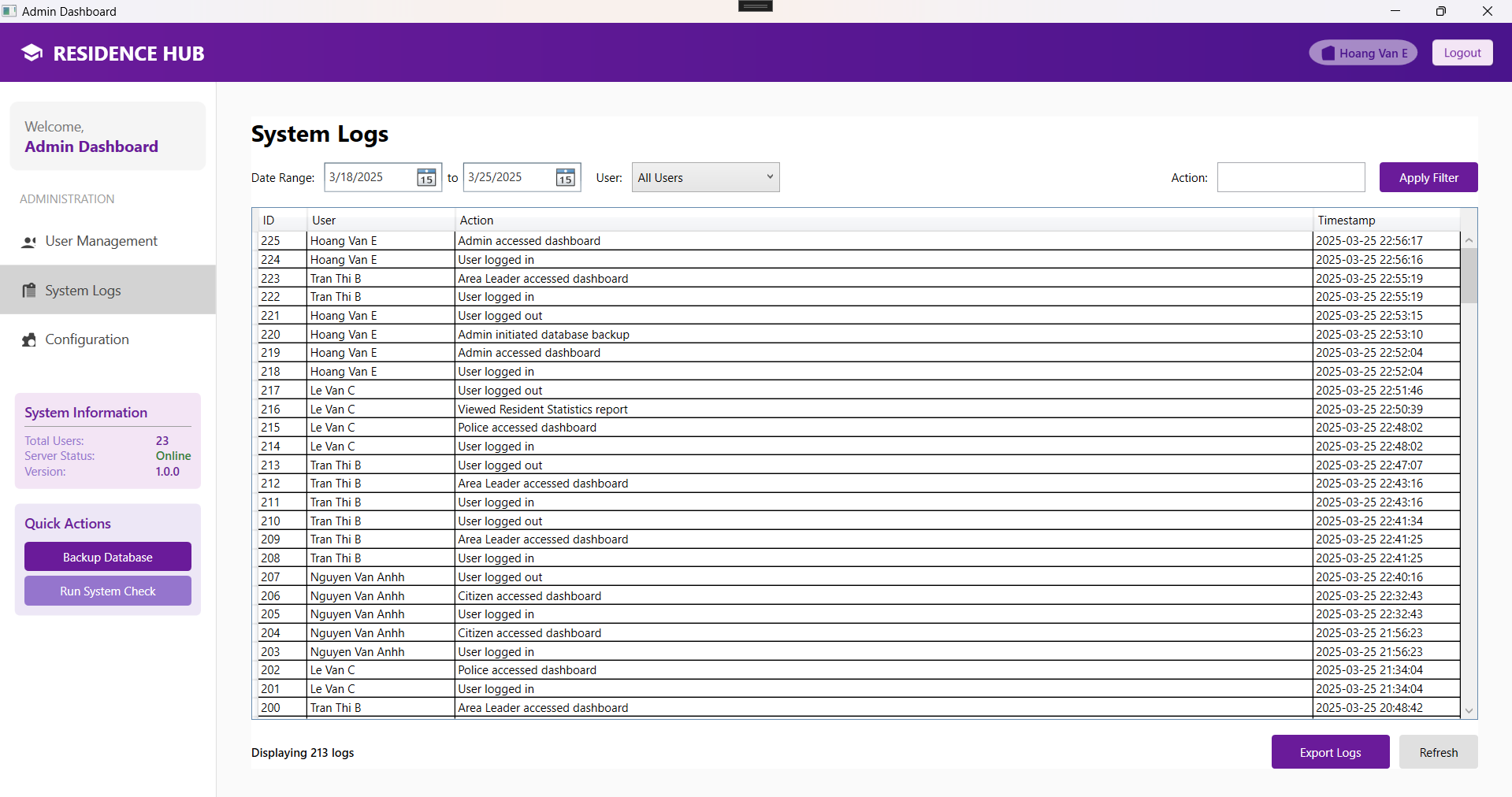
5. Police can view request approval

****

6. Police can view report

### 

7. Police can send notifications.  
  
  
  
Admin

1. Admin logs in with the designated account  
  
  
  
2, After Login  
  
  
  
  
  
3, Admin can manage all users  
  
  
  
  
  
  
  
  
  
4, Admin can view history login  
  
  
  
5, Admin can config system  
