

Table of Contents

1	Introduction	3
2	User Stories	4
2.0	Authentication	4
2.1	Admin role – Add property.....	4
2.2	Admin role – Delete property.....	4
2.3	User role – View all properties	4
2.4	User role – Filter properties by Type.....	5
2.5	User role – Filter properties by Beds	5
3	Demo of User Stories	5
3.1	Admin role – Add property.....	6
3.2	Admin role – Delete property.....	7
3.3	User role – View all properties	7
3.4	User role – Filter properties by Type.....	8
3.5	User role – Filter properties by Beds	8
4	Evaluation	9
4.1	Points of improvement	9

1 - Introduction

Property Management App

The Property Management App is a **console-based Java application** designed to manage property records. It supports two user roles: **Admin** and **User**, with specific functionalities for each. The application utilizes inheritance, interfaces, exception handling, and method overloading to meet both functional and technical requirements. There are features for data persistence via plain text files, which allows for records to be saved and loaded from text files.

Core Features Implemented

i. Inheritance

- A base class Property defines common attributes.
- The House class extends Property and adds house-specific attributes (e.g., number of beds, type).

ii. Interfaces

- The PropertyOperations interface defines methods for adding, viewing, and deleting properties.

iii. Exception Handling

- Custom exceptions (AuthenticationException, UnauthorizedAccessException) enhance error handling.

iv. Authentication

- Users log in with credentials stored in users.txt.
- Admins can modify data, while users have read-only access.

v. Data Persistence

- Property records are saved to and loaded from houses.csv.

vi. User Interaction

- The admin menu supports adding, viewing, filtering, and deleting properties.
- The user menu allows viewing and filtering properties with a looping menu system.

vii. Filtering with Method Overloading

- Overloaded methods in AdminPropertyManager filter properties by type or number of bedrooms.

2 - User Stories

2.0 As an Admin, I want to be authenticated

As an admin, I want to be authenticated via username and password.

Acceptance Criteria:

- I can log in using registered username and password.
- I can not log in by using incorrect username and/or password.

2.1 As an Admin, I want to add property details

As an admin, I want to input details for a new property (e.g., size, price, location, type) so that I can maintain an up-to-date list of properties.

Acceptance Criteria:

- Admin selects the "Add Property" option.
- System collects all necessary details via prompts (e.g., property ID, location, size, type).
- The property is saved to the list and persisted in the file.

2.2 As an Admin, I want to delete a property

As an admin, I want to delete a property from the system using its unique ID so that I can remove outdated or invalid entries.

Acceptance Criteria:

- Admin selects the "Delete Property" option.
- System prompts for the property ID.
- If the property exists, it is removed, and the updated list is saved to the file.
- If the property does not exist, an error message is displayed.

2.3 As a User, I want to view all properties

As a user, I want to see a list of all properties, including their details (e.g., type, location, price, beds), so that I can browse available options.

Acceptance Criteria:

- User selects the "View All Properties" option.
- System displays details of each property in a readable format.
- If no properties are available, a message is displayed indicating an empty list.

2.4 As a User, I want to filter properties by type

As a user, I want to filter the list of properties by their type (e.g., Apartment, Detached) so that I can focus on properties that match my preferences.

Acceptance Criteria:

- User selects the "View Properties by Type" option.
- System prompts for a type choice (e.g., Apartment, Detached).
- System displays all properties of the selected type.
- If no properties match, a message is displayed indicating no results.

2.5 As a User, I want to filter properties by the number of bedrooms

As a user, I want to filter the list of properties by the number of bedrooms so that I can find properties that meet my size requirements.

Acceptance Criteria:

- User selects the "View Properties by Number of Beds" option.
- System prompts for the desired number of bedrooms.
- System displays all properties that match the criteria.
- If no properties match, a message is displayed indicating no results.

3 - Demo of User Stories

User story 2.0

[illegible]

Figure 1 - Admin login - success

```
Only authenticated users can access this app!
=====

Enter username:
admin
Enter password:
abcdef00000
Authentication failed: Invalid username or password!

Process finished with exit code 0
```

Figure 2 - Admin login - Failed

```
Only authenticated users can access this app!
=====

Enter username:
user1
Enter password:
user123
Login successful! Role: USER

User Options:
1. View All Properties
2. View Properties by Type
3. View Properties by Number of Beds
4. Exit
Enter your choice:
```

Figure 3 - User login - success

User story 2.1

```
Enter your choice: 1
Enter Property ID:
78
Enter Location:
Athlone
Enter Size (in sqm):
90
Enter Price:
400000
Enter Number of Beds:
3
Enter Number of Baths:
3
Does it have a yard? (true/false):
true
Enter House Type (1-Apartment, 2-SemiDetached, 3-Detached):
2
Property added successfully and saved to file!
```

Figure 4 - Admin - Add property

User story 2.2

```
Property ID: 78
Location: Athlone
Size: 90.0 sqm
Price: €400000.0
Bedrooms: 3
Bathrooms: 3
Yard: Yes
House Type: typeSemiDetached

Admin Options:
1. Add Property
2. View All Properties
3. View Properties by Type
4. View Properties by Number of Beds
5. Delete Property
6. Exit
Enter your choice: 5
Enter Property ID to delete:
78
Property deleted successfully and changes saved to file!
```

Figure 5 - Admin - Delete property

User story 2.3

```
Enter your choice: 1
Property ID: 45
Location: Dublin
Size: 120.0 sqm
Price: €78000.0
Bedrooms: 3
Bathrooms: 2
Yard: Yes
House Type: typeSemiDetached

Property ID: 56
Location: Cork
Size: 78.0 sqm
Price: €250000.0
Bedrooms: 2
Bathrooms: 2
Yard: No
House Type: typeApartment
```

Figure 6 - User - View all properties

User story 2.4

```
User Options:
1. View All Properties
2. View Properties by Type
3. View Properties by Number of Beds
4. Exit
Enter your choice: 2
Enter House Type (1-Apartment, 2-SemiDetached, 3-Detached):
1
Property ID: 56
Location: Cork
Size: 78.0 sqm
Price: €250000.0
Bedrooms: 2
Bathrooms: 2
Yard: No
House Type: typeApartment

User Options:
1. View All Properties
2. View Properties by Type
3. View Properties by Number of Beds
4. Exit
Enter your choice:
```

Figure 7 - User - Filter by Type

User story 2.5

```
User Options:
1. View All Properties
2. View Properties by Type
3. View Properties by Number of Beds
4. Exit
Enter your choice: 3
Enter Number of Beds:
3
Property ID: 45
Location: Dublin
Size: 120.0 sqm
Price: €78000.0
Bedrooms: 3
Bathrooms: 2
Yard: Yes
House Type: typeSemiDetached

User Options:
1. View All Properties
2. View Properties by Type
3. View Properties by Number of Beds
4. Exit
Enter your choice:
```

Figure 8 - User - Filter by Beds

4 - Evaluation

Overall, the project adhered to all initial requirements and user stories. I utilized various Java object oriented programming principles. The authentication system allows users to log in and perform role-based operations. Admins can add, delete, and view properties. While regular users can view and filter properties based on specific criteria. Lastly, data persistence was achieved effectively using plain text files.

4.1 Points of improvement

Currently, passwords are stored as plain text in the users.txt file. Introducing password hashing (e.g. SHA-256) would enhance security and protect user credentials.

Introducing relational database (e.g., MySQL or PostgreSQL) can provide better data structure, security.

As a Java console application, the UI is functional but lacks visual appeal. Introducing a graphical user interface (GUI e.g. Swing) can make the application more user-friendly.

A web-based interface could also be developed in future versions.