

INTE2512 OBJECT-ORIENTED PROGRAMMING FINAL PROJECT

SHINOBU STORE

DO KHOA NGUYEN - S3978796

TRAN THE QUANG MINH - S3979562

NGUYEN SON TUNG - S3979348

BUI ANH MINH - S3979513

NGUYEN MINH KHOA - S3980823

Table of Contents

1. Backend

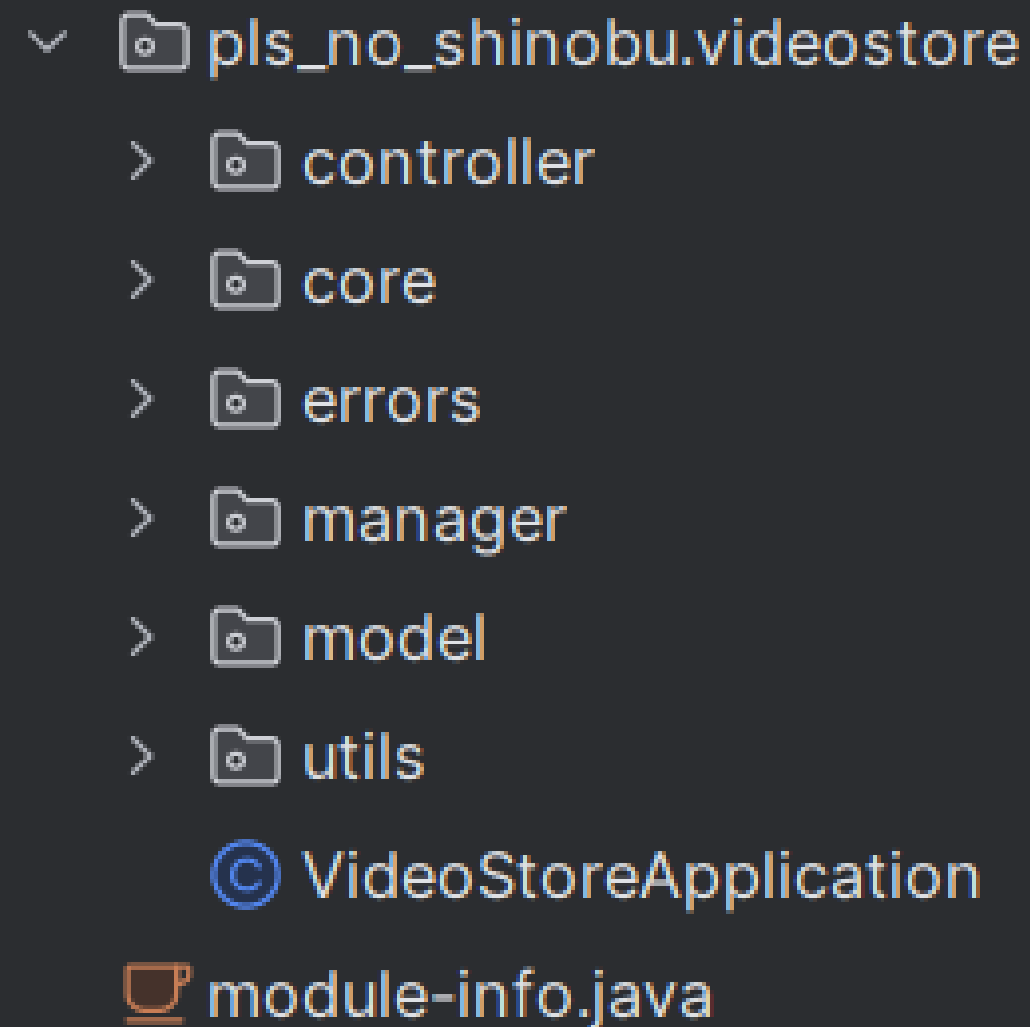
- Folder structure
- Model
- Manager
- Core

2. GUI

- Folder structure
- welcome.fxml
- signup.fxml
- signin.fxml
- userDashboard.fxml
- adminDashboard.fxml
- updateUser.fxml
- addItem.fxml
- updateItem.fxml

Backend - Folder structure

The folder structure:



```

  ✓ [Folder] pls_no_shinobu.videostore
    > [Folder] controller
    > [Folder] core
    > [Folder] errors
    > [Folder] manager
    > [Folder] model
    > [Folder] utils
    © VideoStoreApplication
    [File] module-info.java

```

The screenshot shows a file explorer view of a project named 'pls_no_shinobu.videostore'. The project contains several subfolders: 'controller', 'core', 'errors', 'manager', 'model', and 'utils'. Additionally, there is a 'VideoStoreApplication' package and a 'module-info.java' file.

- **controller:** contains controllers for fxml files.
- **core:** contains classes for core operations.
- **errors:** contains classes for exceptions specified for this application.
- **manager:** contains classer for managing multiple model objects.
- **model:** contains classes for models.
- **utils:** contains classes for utilities for parsing users and items from CSV files and password handling.

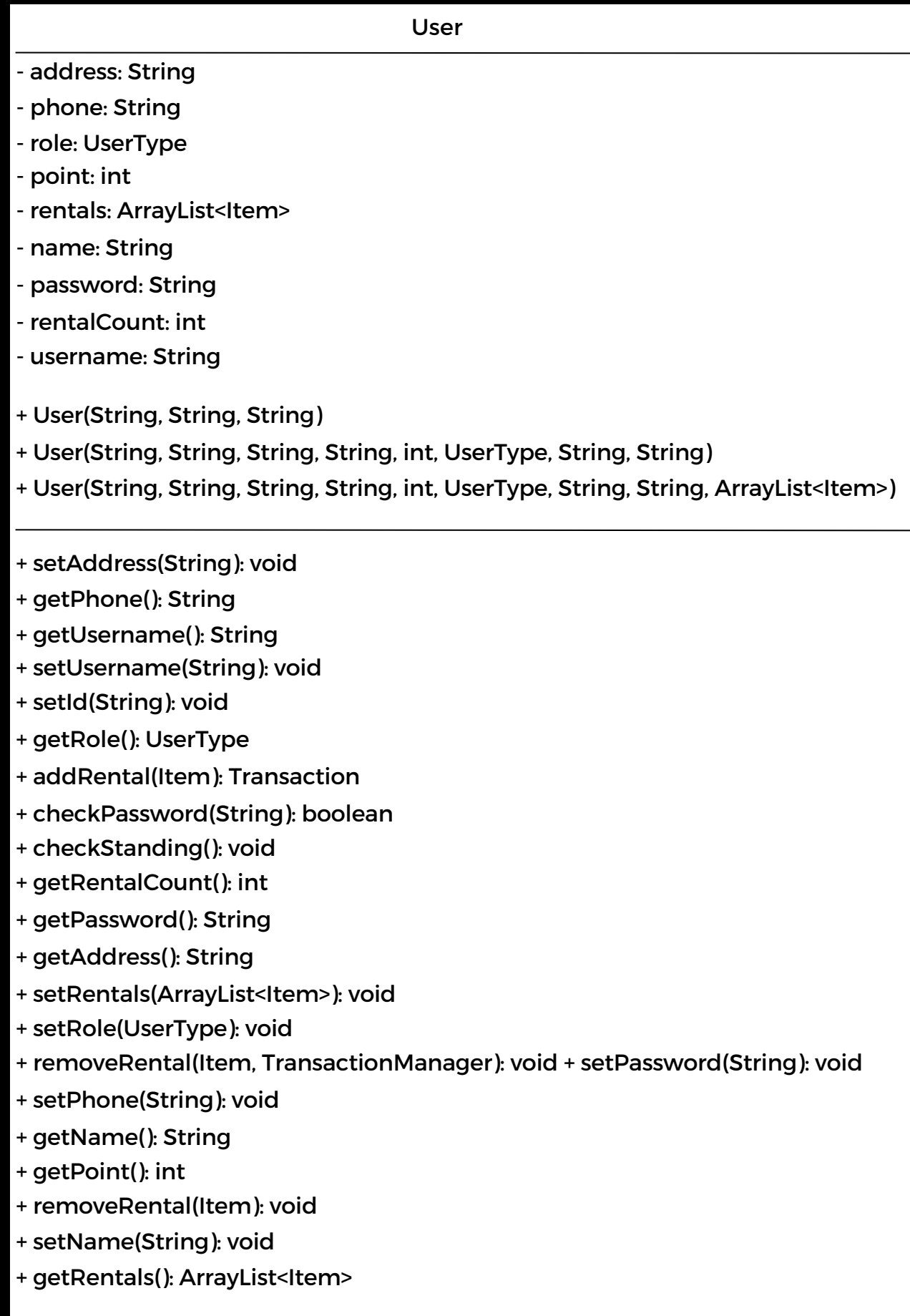
Backend - Model

We started with implementing classes for:

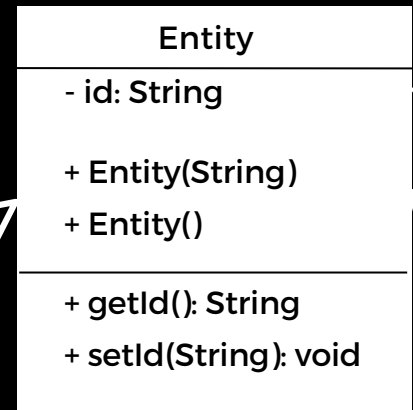
- **User:** Include basic informations, auth informations and rental informations.
- **Item:** Include the item details and the stock with its methods.
- **Transaction:** Store transaction data such as the User, Item, its status and create-resolve timestamp in Instant.

These classes are inherited from the **Entity** class which requires an ID that can be extended.

Backend - Model

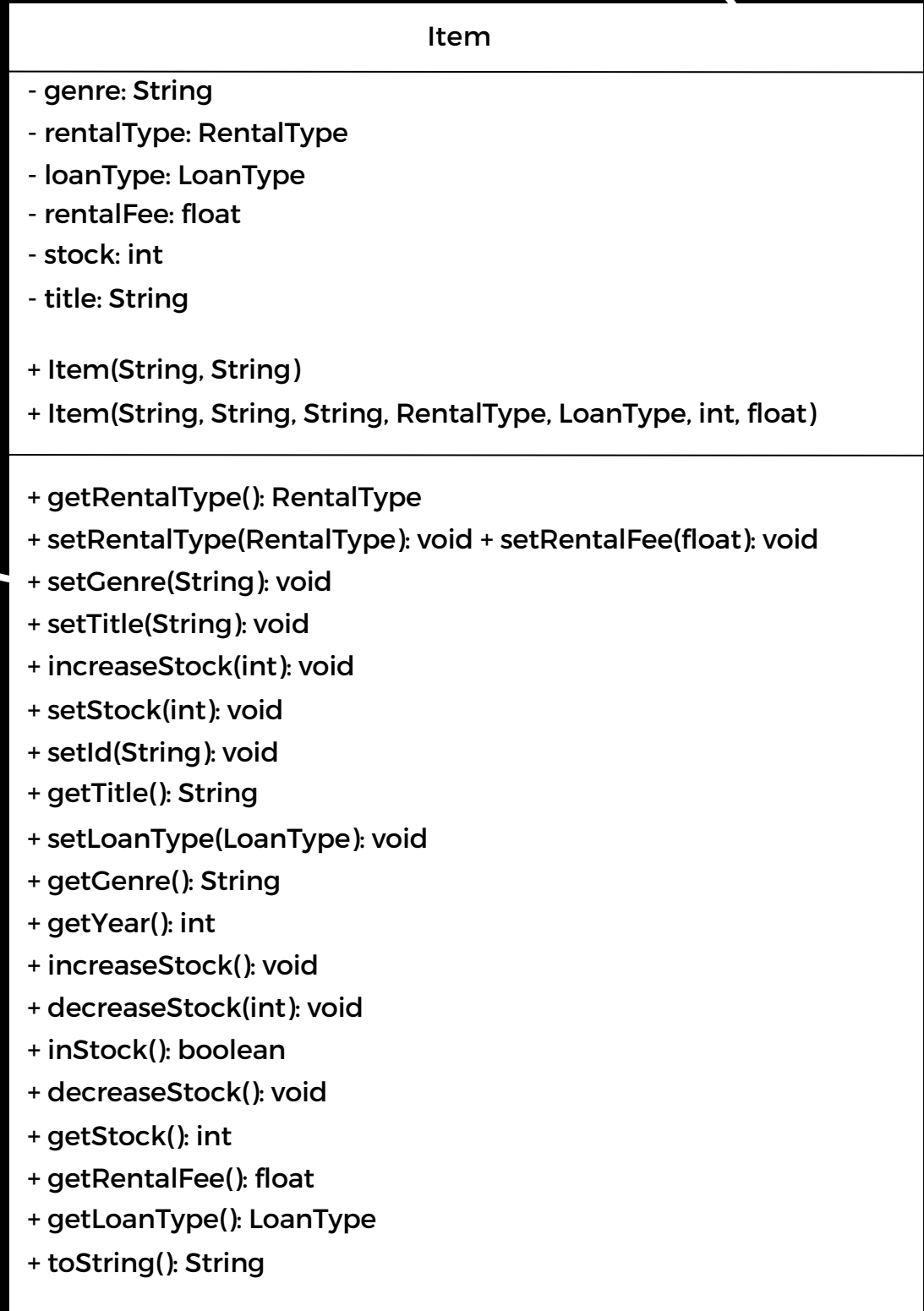


Extends

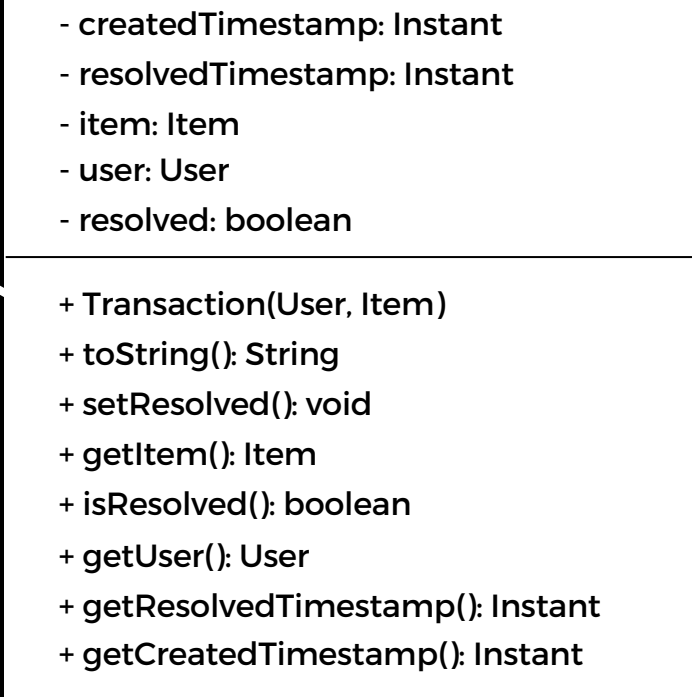


Extends

Extends



Transaction



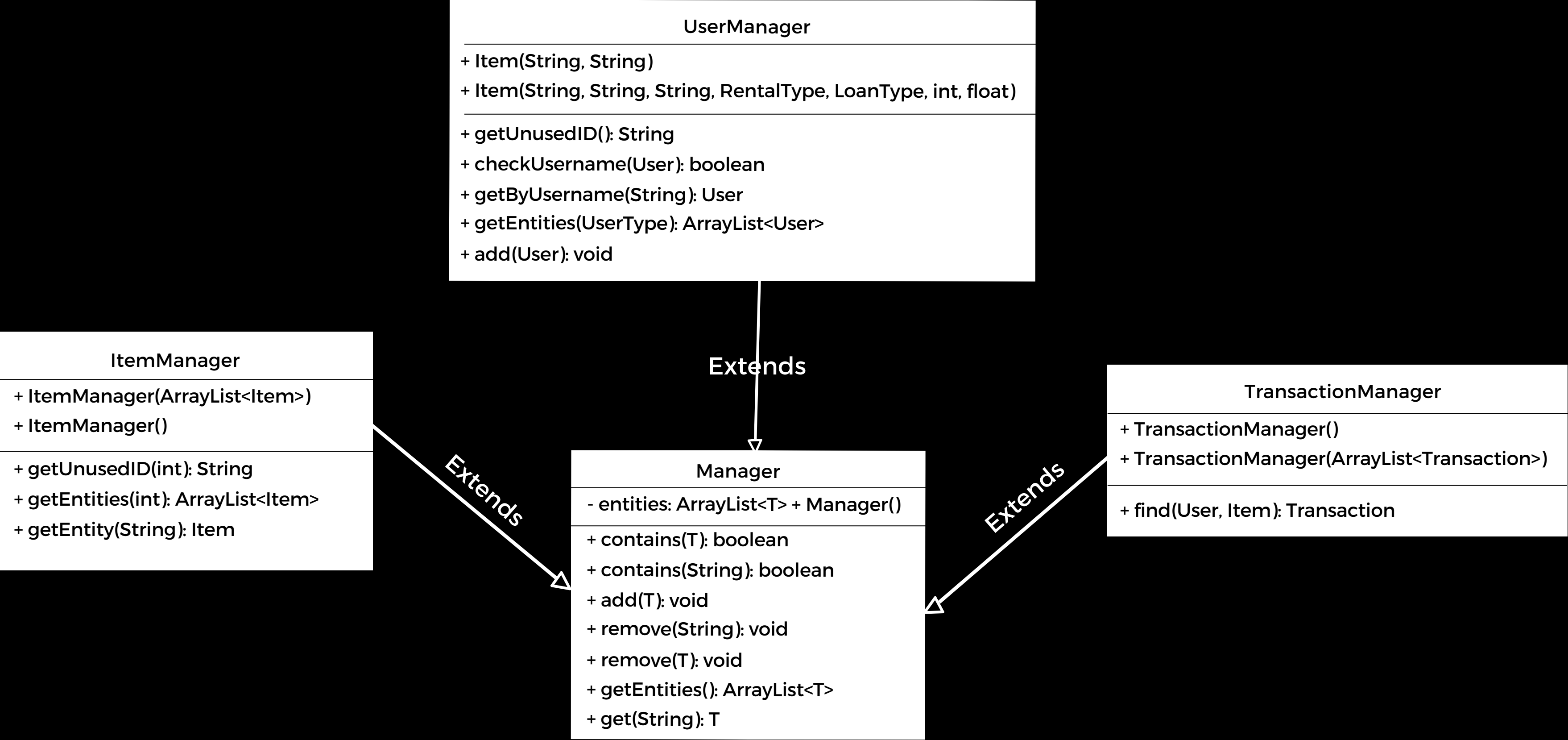
Backend - Manager

We then implement classes for managing multiple objects of the models, in which we can create, delete or to provide information:

- **UserManager**: Class for manage multiple User.
- **ItemManager**: Class for manage multiple Item.
- **TransactionManager**: Class for manage multiple Transaction.

These classes are inherited from the **Manager** class, which is an abstract class for implementing Entity manager.

Backend - Manager

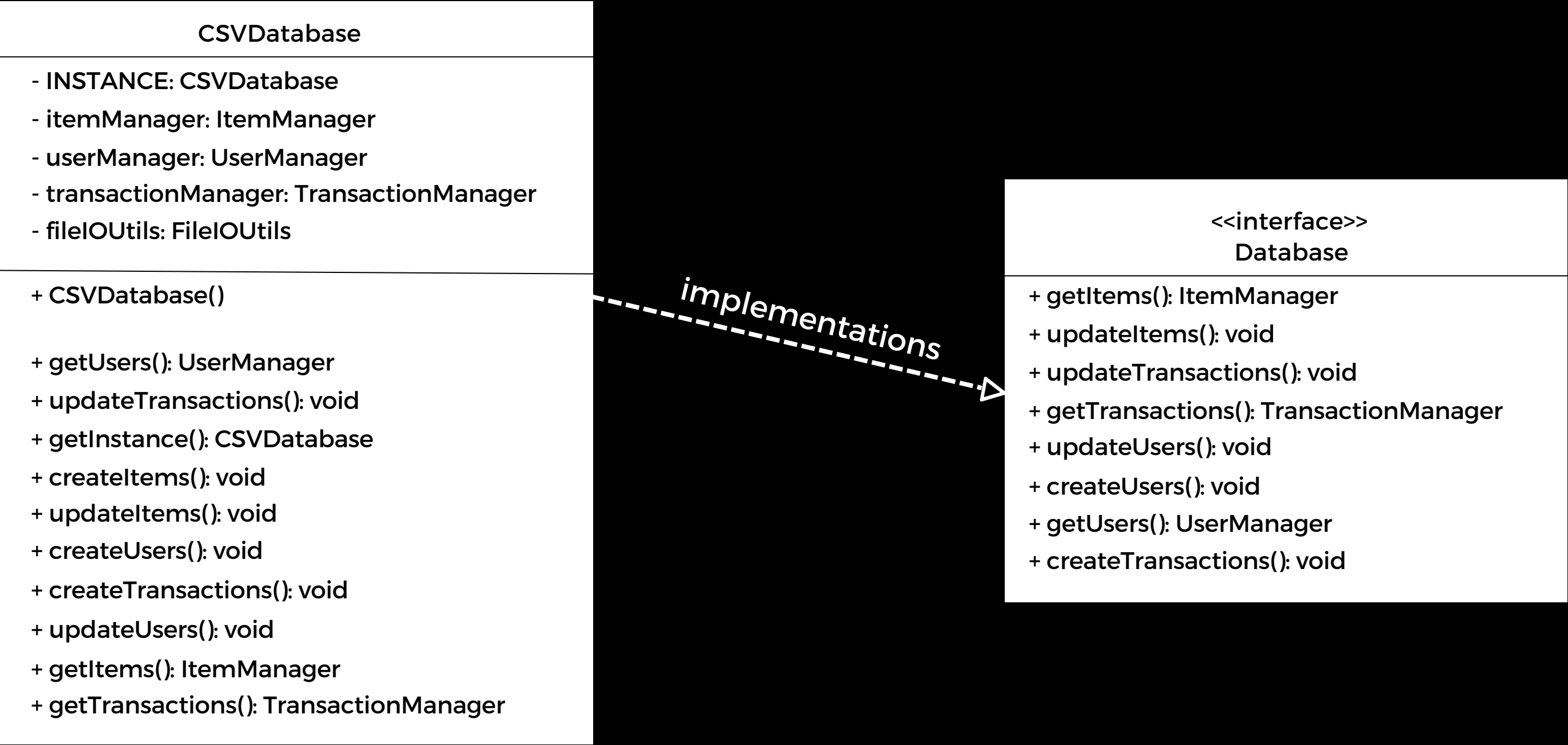


Backend - Core

We then implement classes for the main operations of the application:

- CSVDatabase: implements the Database interface, handle CRUD operations for the CSV database.
- Session: provide credentials for sign in and log out.

Backend - Core



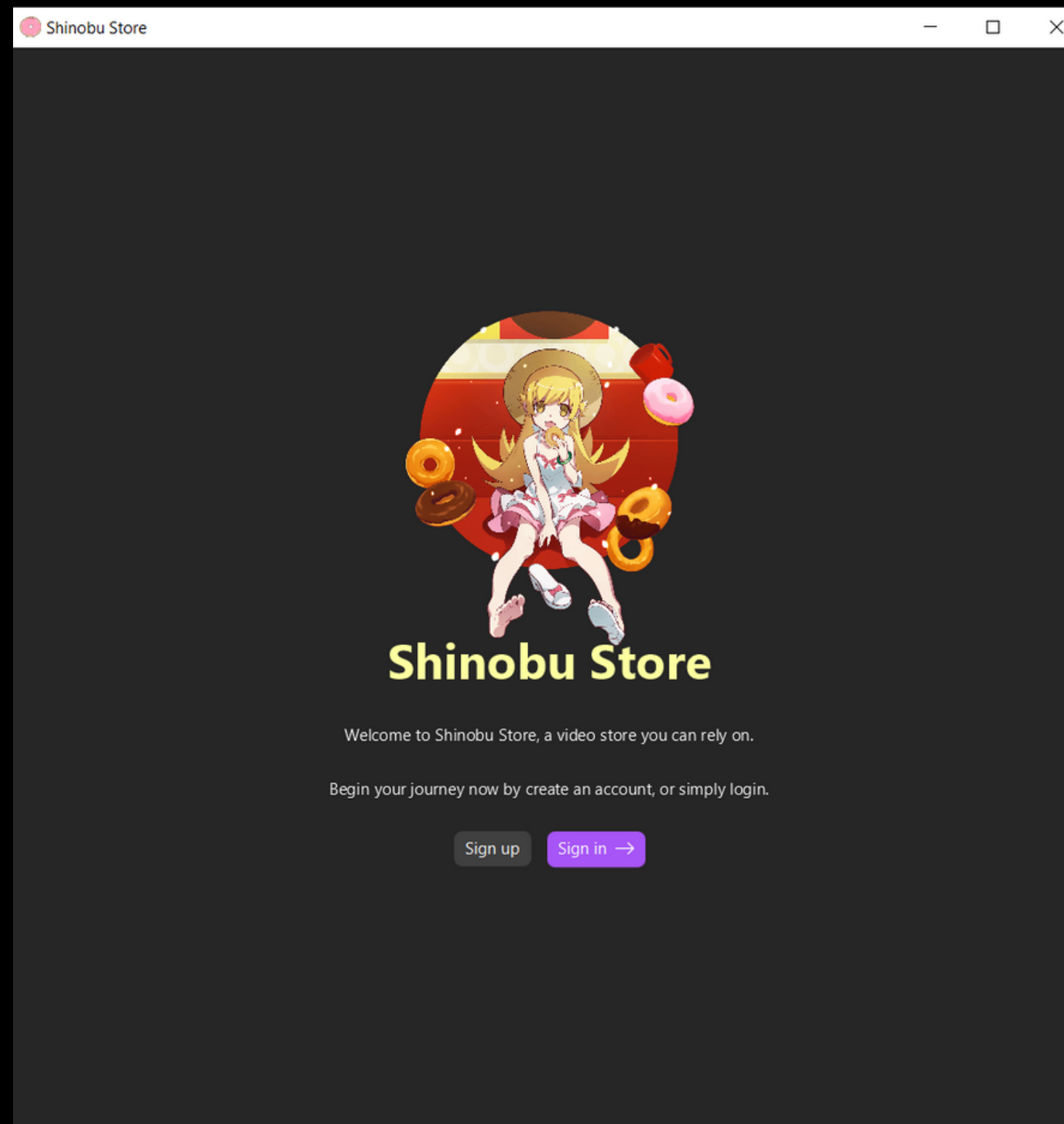
GUI - Folder structure

```

v  pls_no_shinobu.videostore
  >  css
  >  img
    </> addItem.fxml
    </> adminDashboard.fxml
    </> signin.fxml
    </> signup.fxml
    </> updateItem.fxml
    </> updateUser.fxml
    </> userDashboard.fxml
    </> welcome.fxml
```

- **css:** contains css files
- **img:** contains image files
- **8 fxml files**

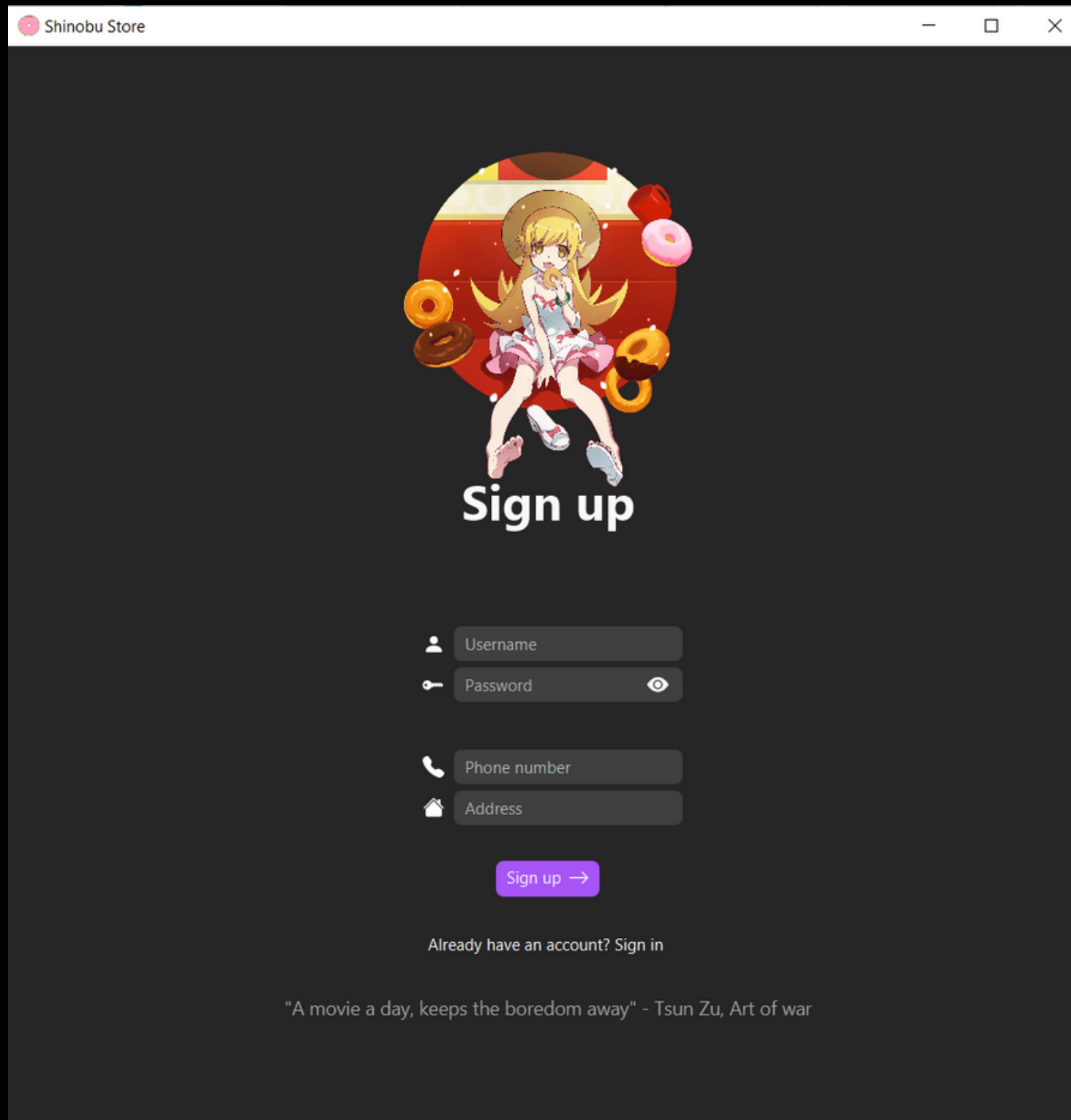
GUI - welcome.fxml



Features:

- Index page of the Shinobu Store
- Buttons link to Sign in and Sign up pages

GUI - signup.fxml



The image shows a screenshot of a web application window titled "Shinobu Store". The window has a dark gray background. At the top center, there is a circular illustration of a blonde anime-style girl with long hair, wearing a pink and white outfit, sitting on a red and yellow striped circular background. Below the illustration, the text "Sign up" is written in a bold, white font. Underneath the text, there are four text input fields, each with a small icon to its left: a person icon for "Username", a key icon for "Password" (which also has an eye icon to its right for toggling visibility), a phone icon for "Phone number", and a house icon for "Address". Below these fields is a blue button with the text "Sign up" and a right-pointing arrow. At the bottom of the form, there is a link that says "Already have an account? Sign in". At the very bottom of the window, there is a quote: "A movie a day, keeps the boredom away" - Tsun Zu, Art of war.

Shinobu Store

Sign up

Username

Password

Phone number

Address

Sign up →

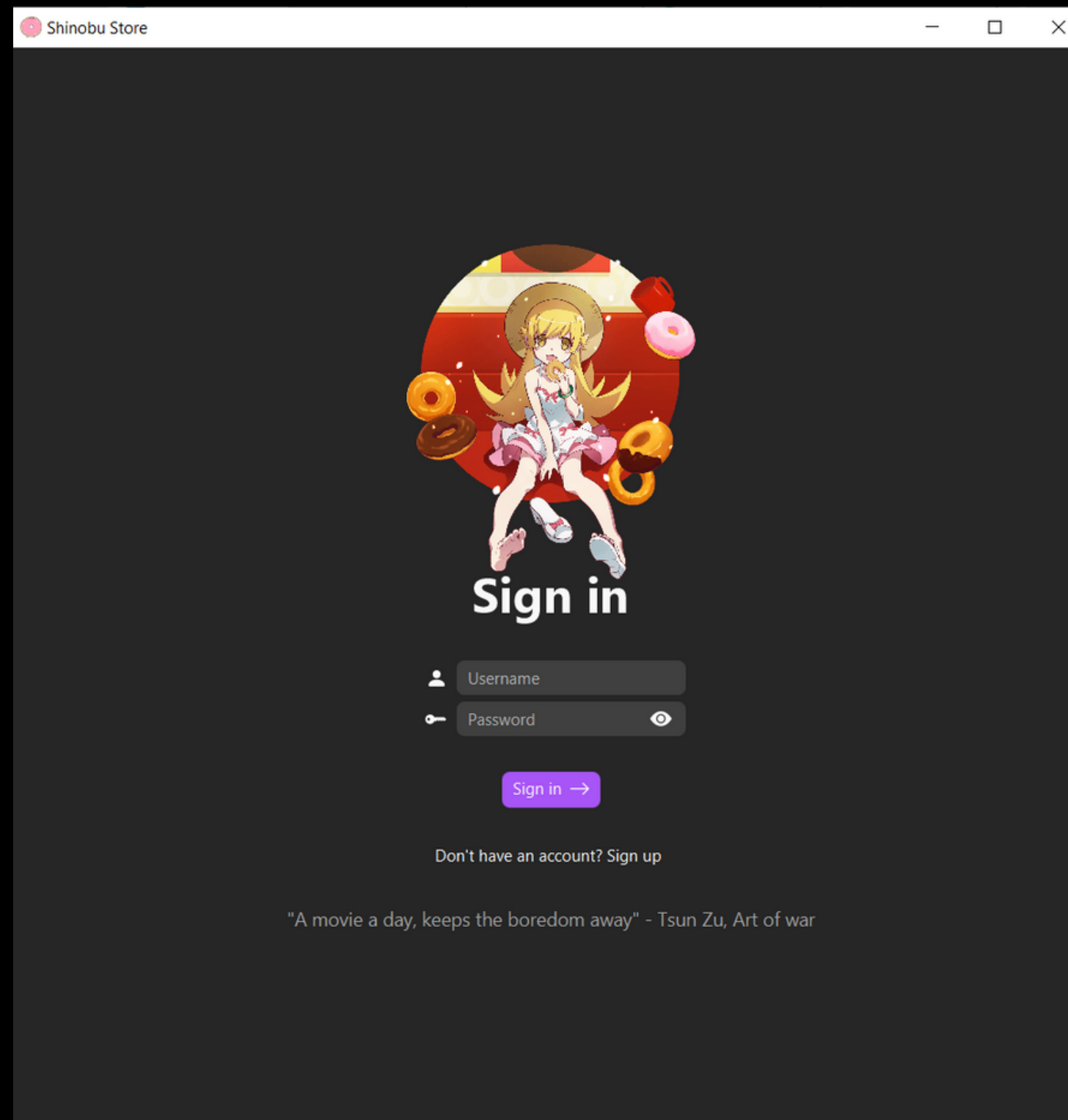
Already have an account? Sign in

"A movie a day, keeps the boredom away" - Tsun Zu, Art of war

Features:

- Text fields for the user to input their information.
- Button to add the user to the database then log the user in.
- All inputs are validated
- Button link to Sign in page.

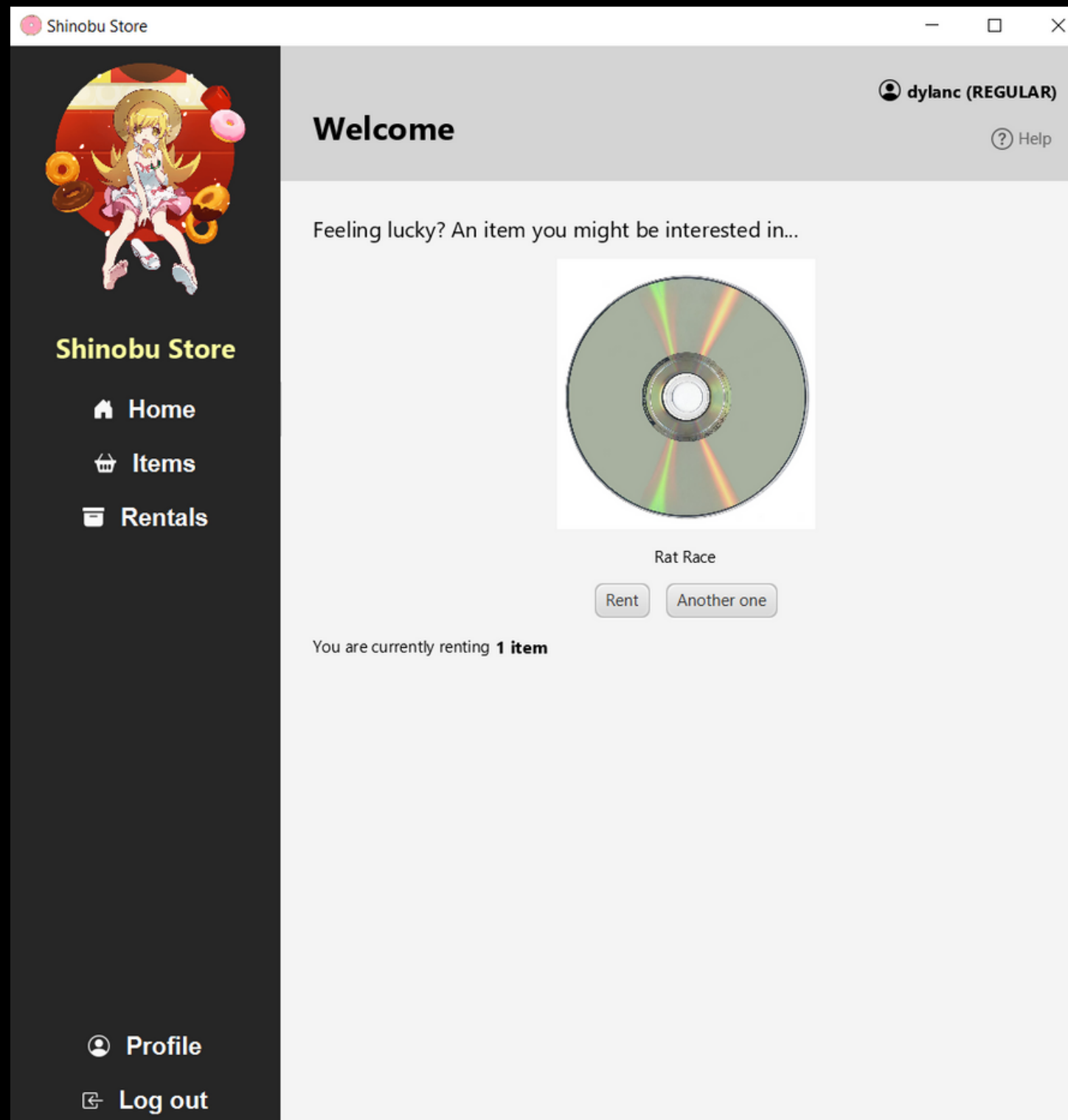
GUI - signin.fxml



Features:

- Text fields for the user to input their username and password.
- An icon for toggling visibility of the password field
- Button to log in and create a session if the provided informations are correct.

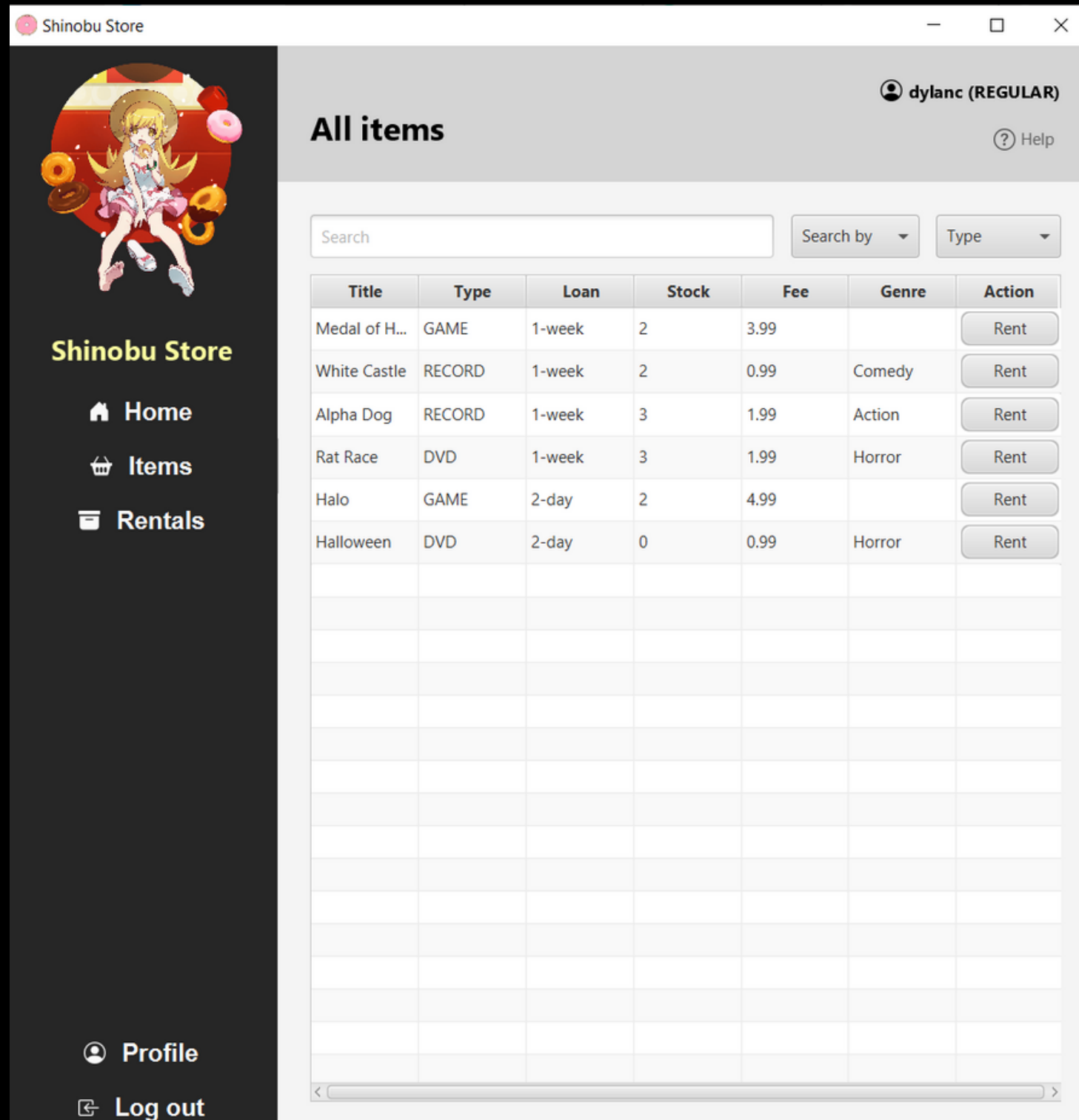
GUI - userDashboard.fxml



Features:

- Welcome page that recommend a random item to the user. Also display the number of item the user currently renting.
- Sideboard with buttons allows the user to navigate the app.
- Display the username and their membership on the top right.
- 'Help' button.

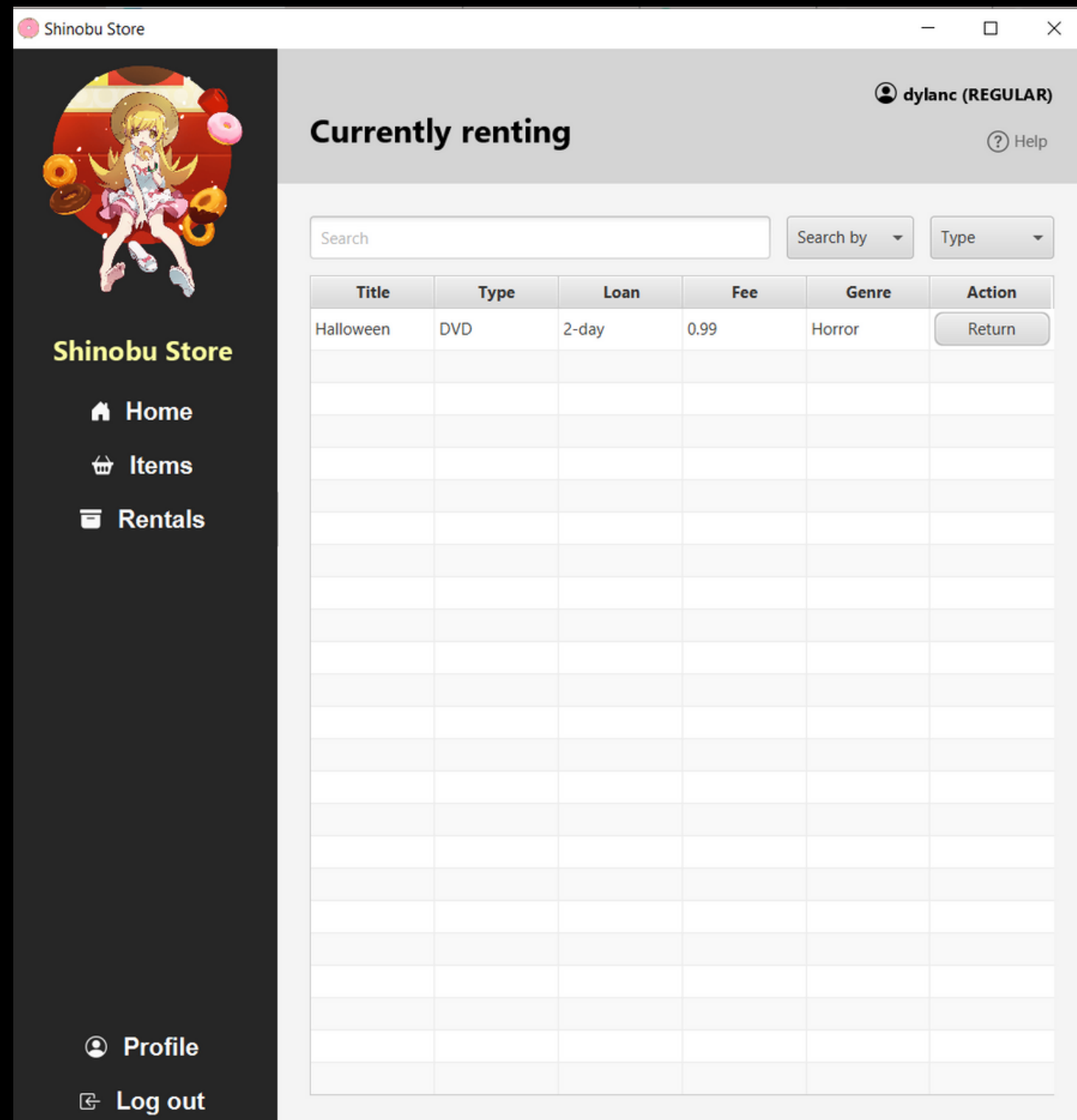
GUI - userDashboard.fxml



Features:

- This page corresponds with the 'Items' button, displays all items currently in the store.
- Search bar, allows the user to search for items by either IDs or names.
- Filter function, displays the items by their type.

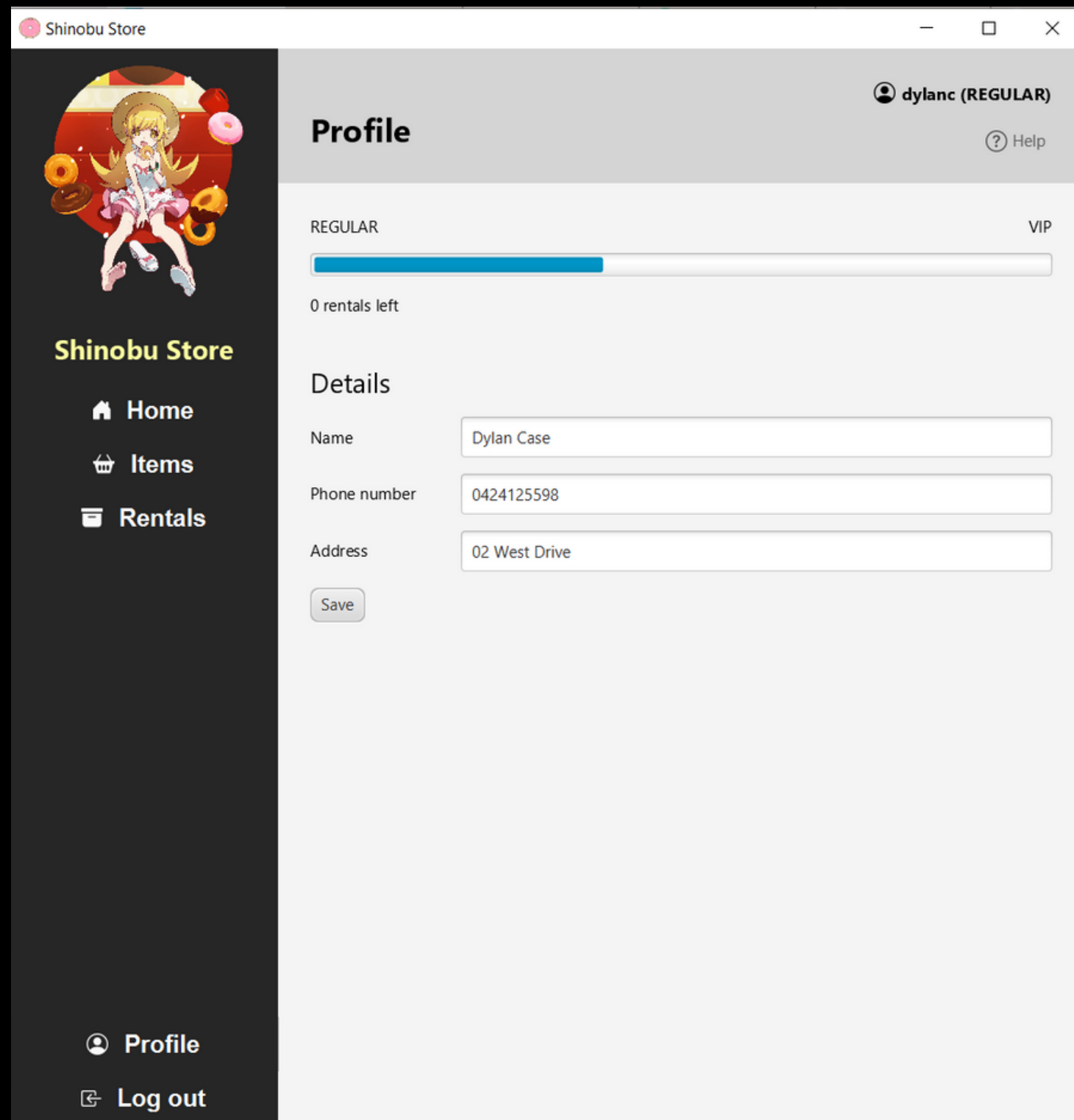
GUI - userDashboard.fxml



Features:

- This page corresponds with the 'Rentals' button, displays all the user's current rented item.
- Same search and filter function as the last page.
- Button that allows the user to return the item.

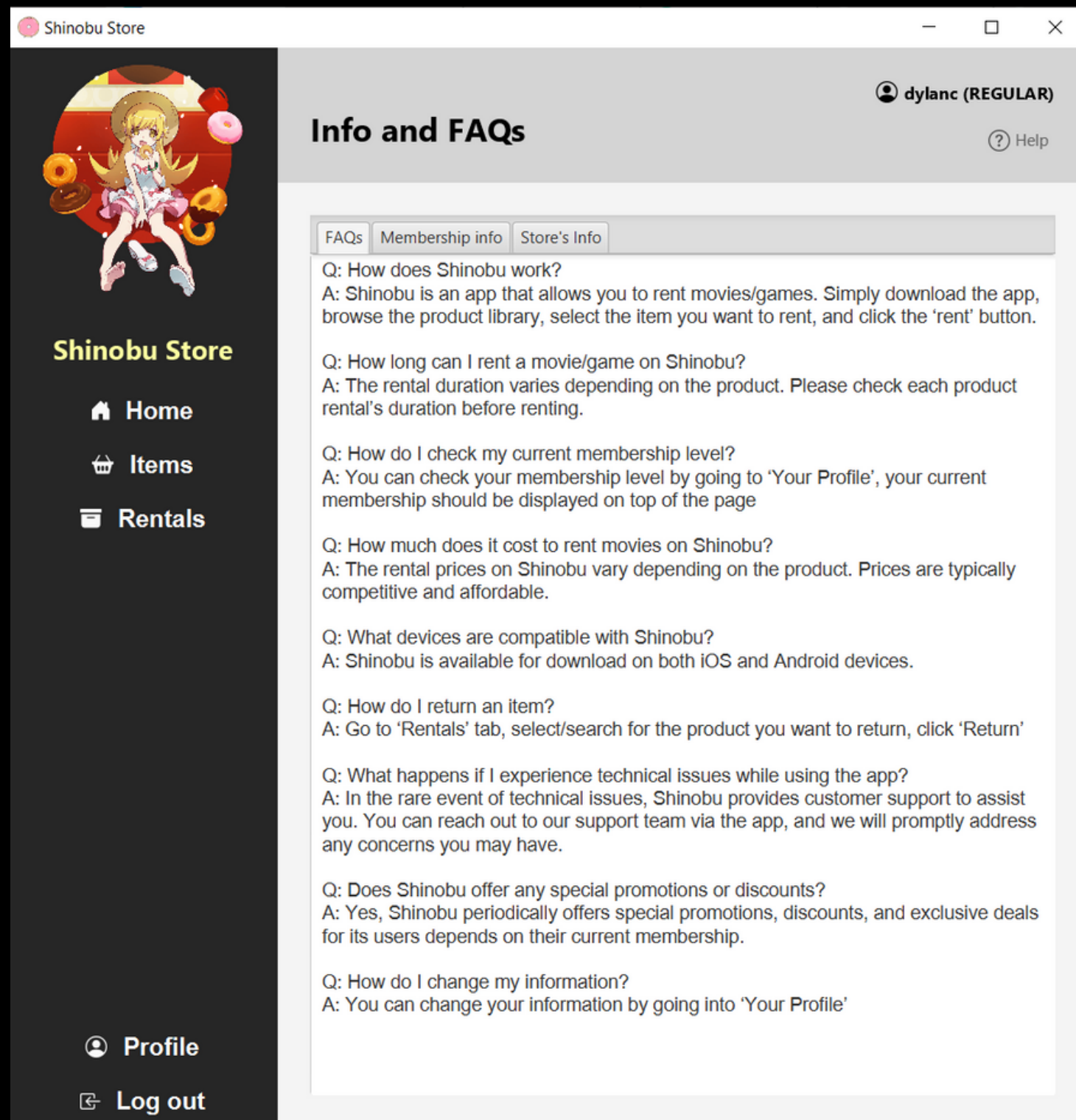
GUI - userDashboard.fxml



Features:

- This page corresponds with the 'Profile' button, displays the user's info and their current membership progression.
- Allows the user to edit their information.

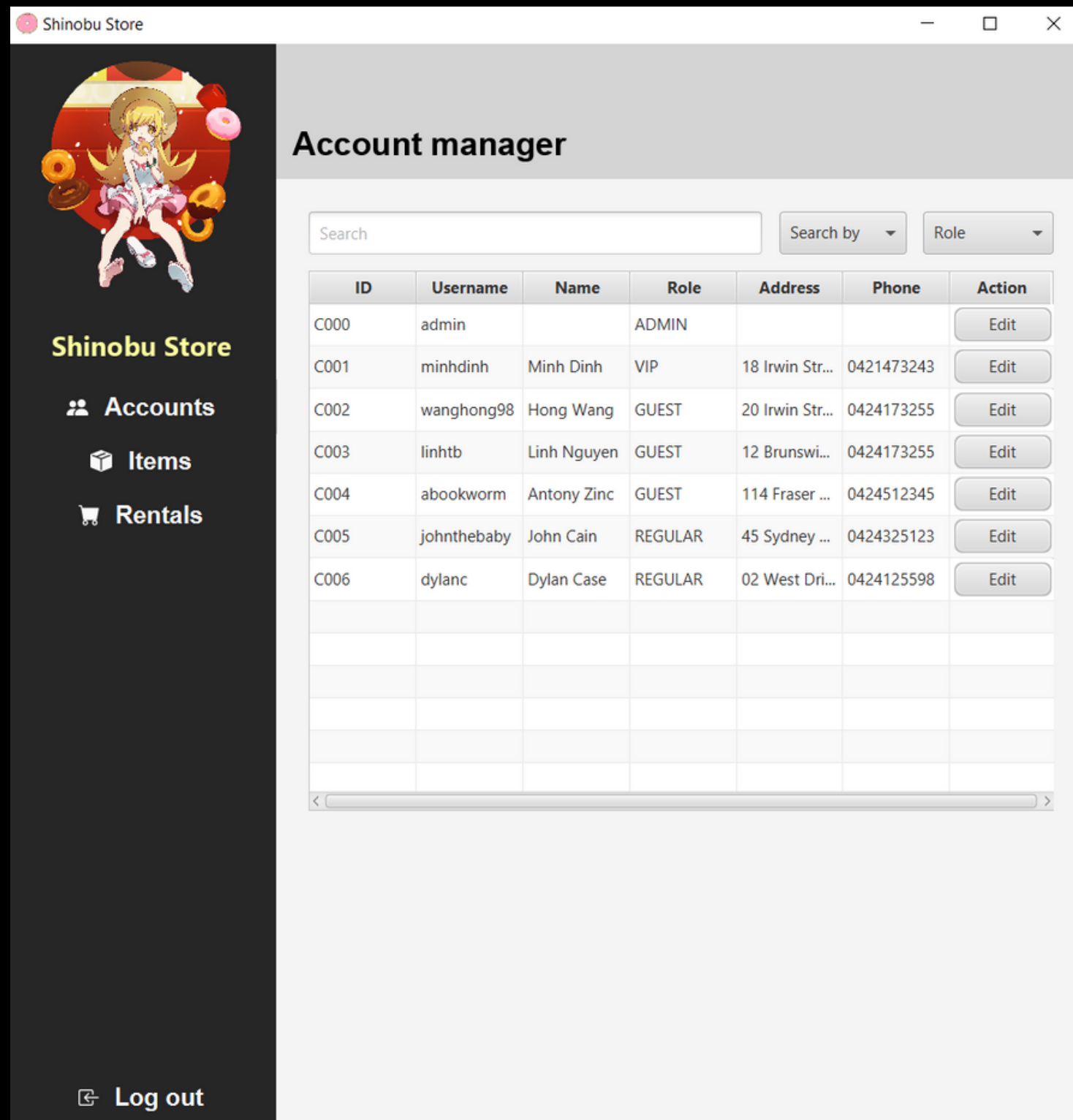
GUI - userDashboard.fxml



Features:

- This page corresponds with the 'Help' button, which displays FAQs, membership rules and store information.

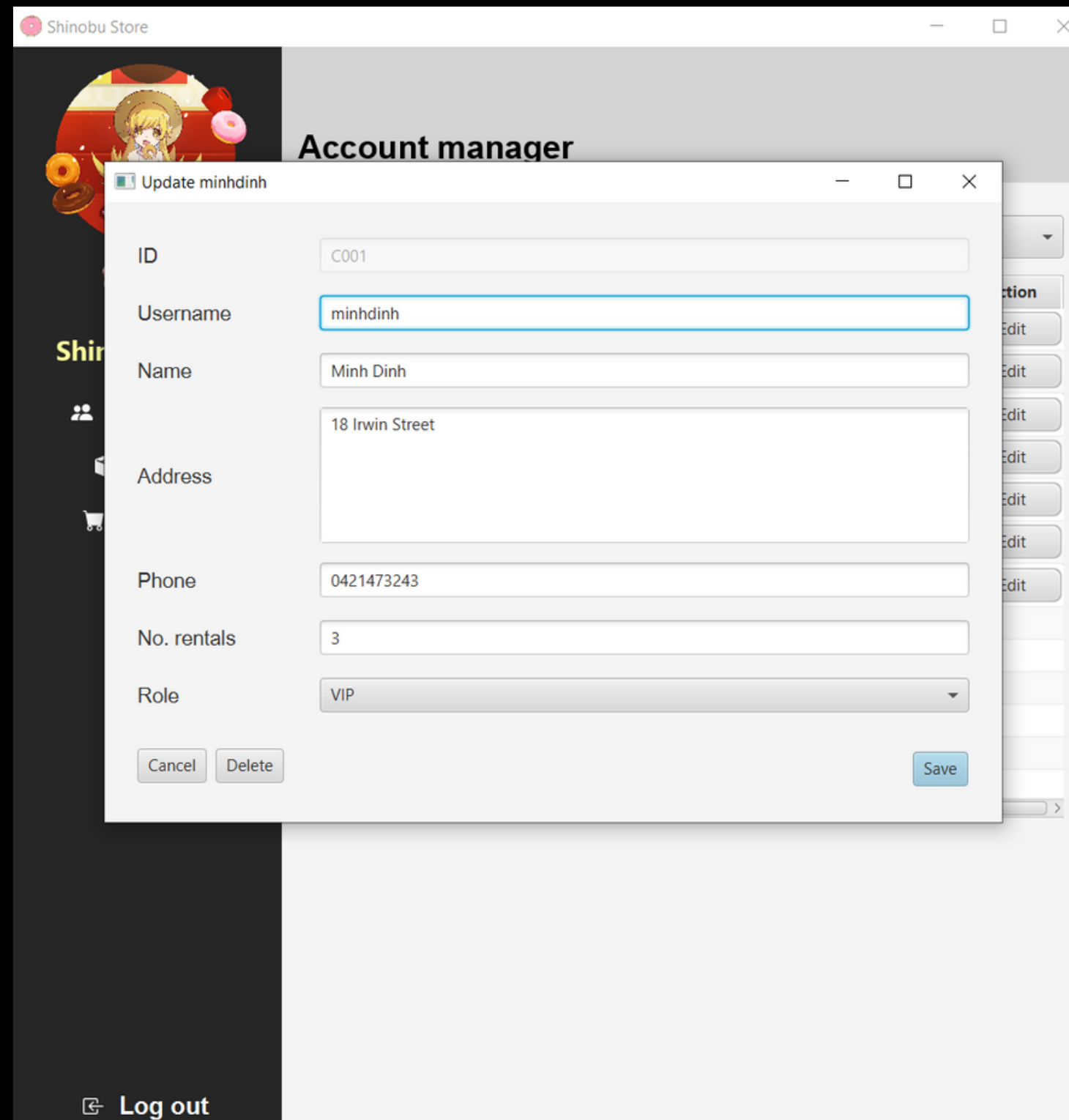
GUI - adminDashboard.fxml



Features:

- Sidebar with buttons to let the admin navigate the app.
- This page corresponds with the 'Accounts' button, displaying all the existing accounts.
- Search bar that allows the admin to search for users by name or ID.
- Filter function which displays the users by their membership level.
- Button to edit the user's info.

GUI - updateUser.fxml



The screenshot displays the 'Shinobu Store' application window. The main content area is titled 'Account manager'. A modal dialog box titled 'Update minhdinh' is open, allowing for user information updates. The dialog contains the following fields and controls:

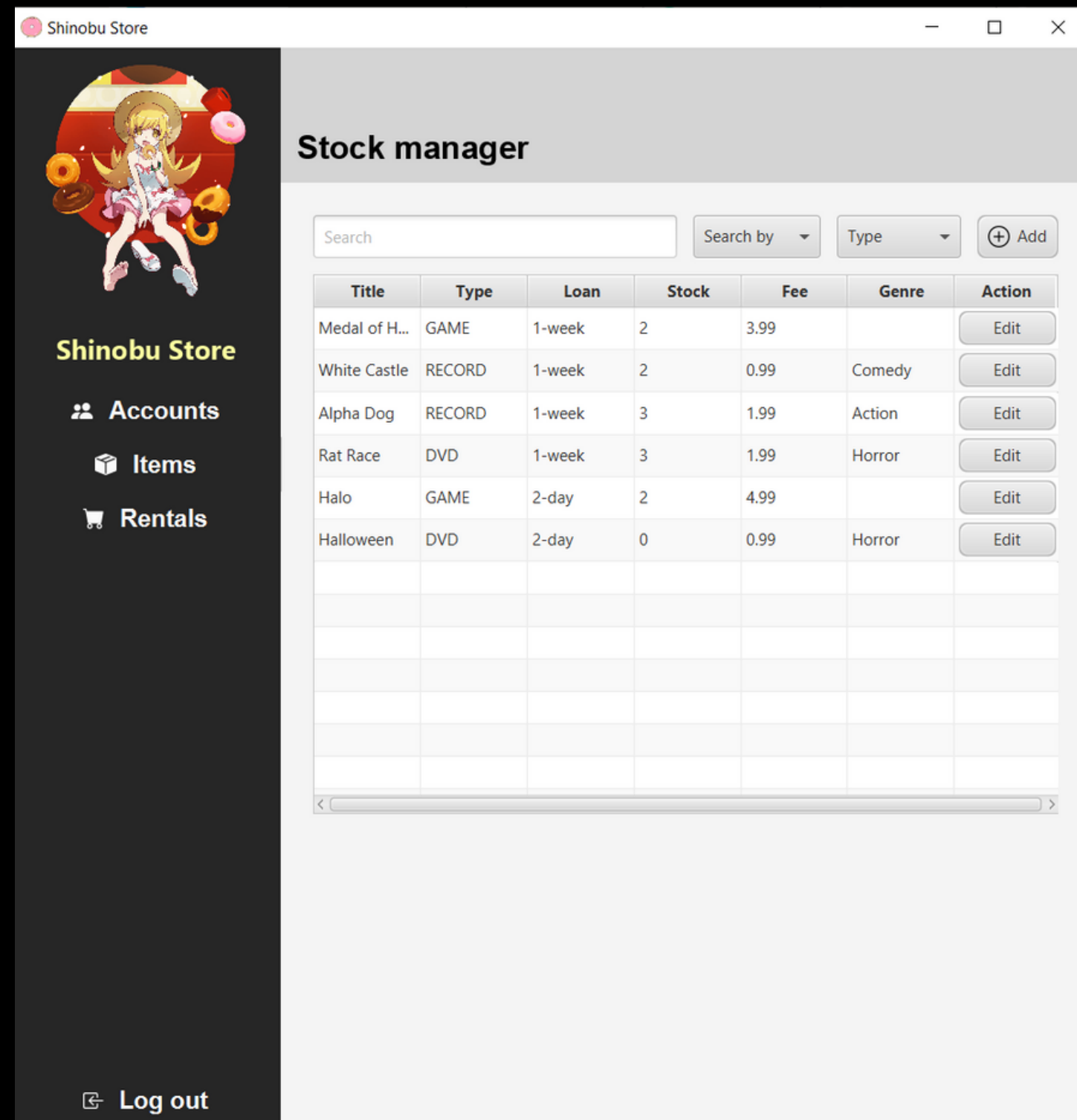
- ID:** Text field containing 'C001'.
- Username:** Text field containing 'minhdinh'.
- Name:** Text field containing 'Minh Dinh'.
- Address:** Text area containing '18 Irwin Street'.
- Phone:** Text field containing '0421473243'.
- No. rentals:** Text field containing '3'.
- Role:** Dropdown menu currently set to 'VIP'.
- Buttons:** 'Cancel', 'Delete', and 'Save' buttons are located at the bottom of the dialog.

The background application window includes a sidebar with a 'Log out' button at the bottom and a list of users on the right, each with an 'Edit' button.

Features:

- Pop up when the admin clicked the 'Edit' button, user's info can be updated here or delete the user out of the database.

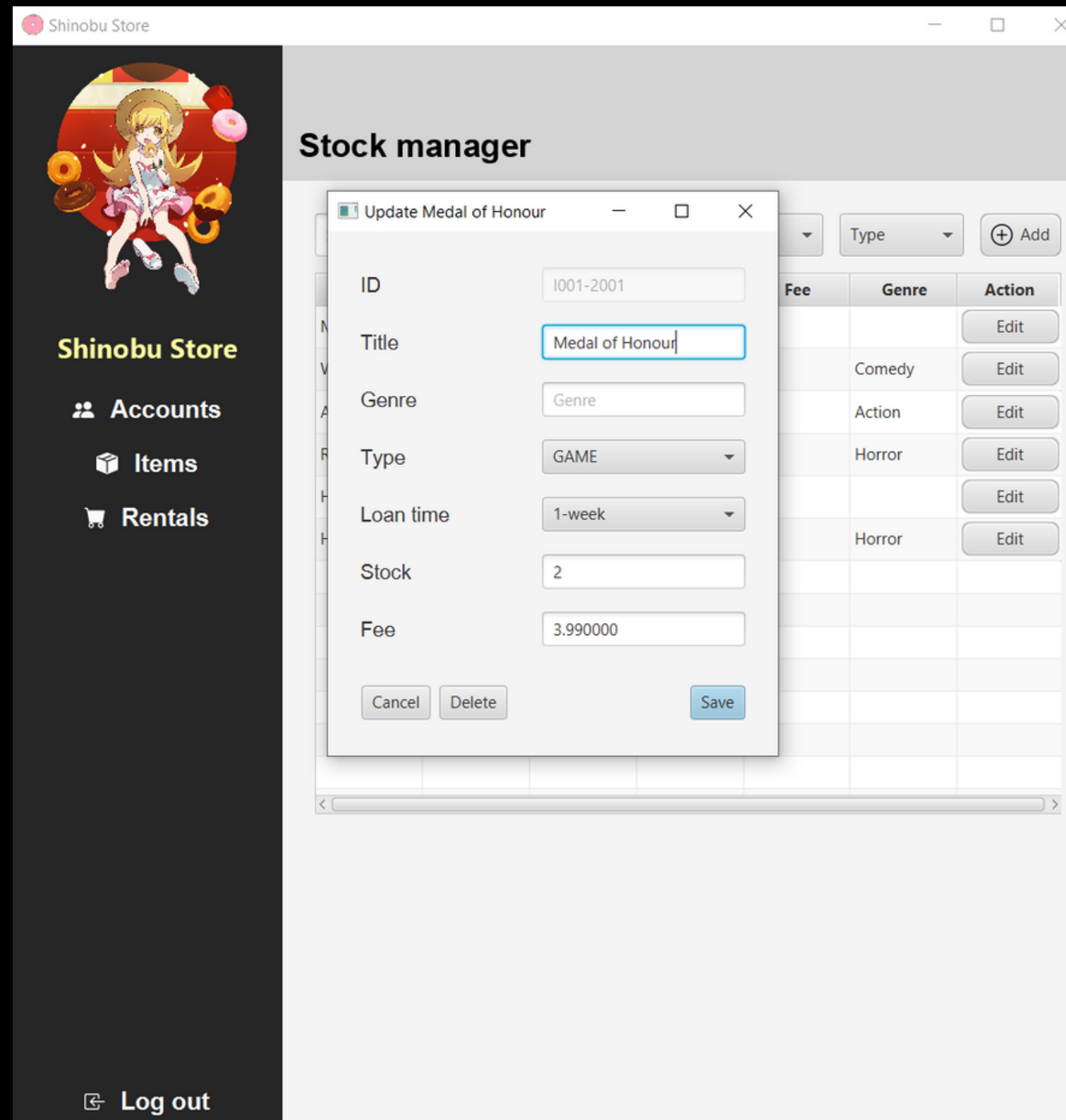
GUI - adminDashboard.fxml



Features:

- This page corresponds with the 'Items' button, allows the admin to manage the store's stock.
- Same search and filter as the user.
- Button to add more item into the stock.
- Button to edit each item's info.

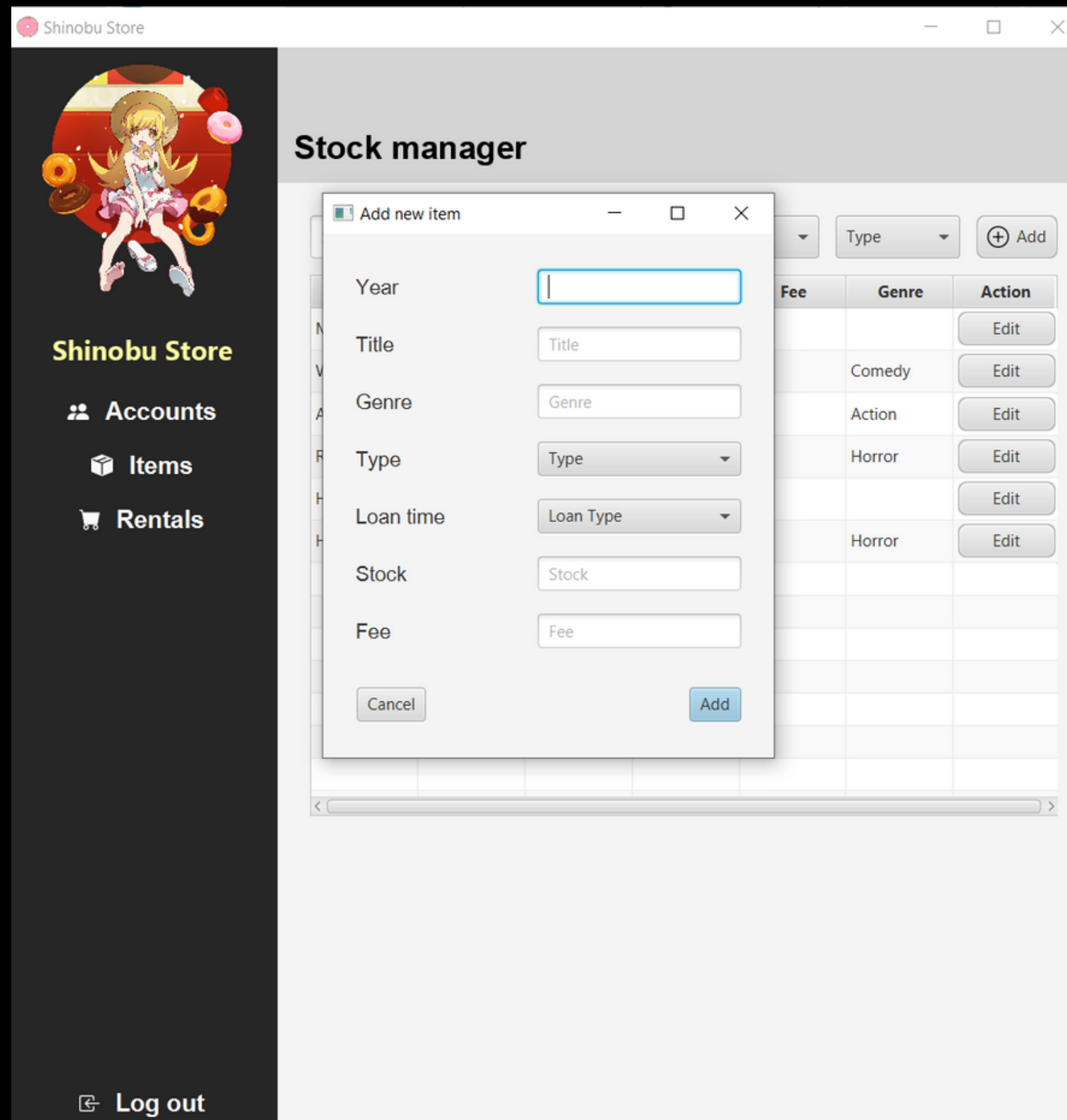
GUI - updateItem.fxml



Features:

- Pop up when the 'Edit' button is clicked, allows the admin to update the item's info or delete them from the database.

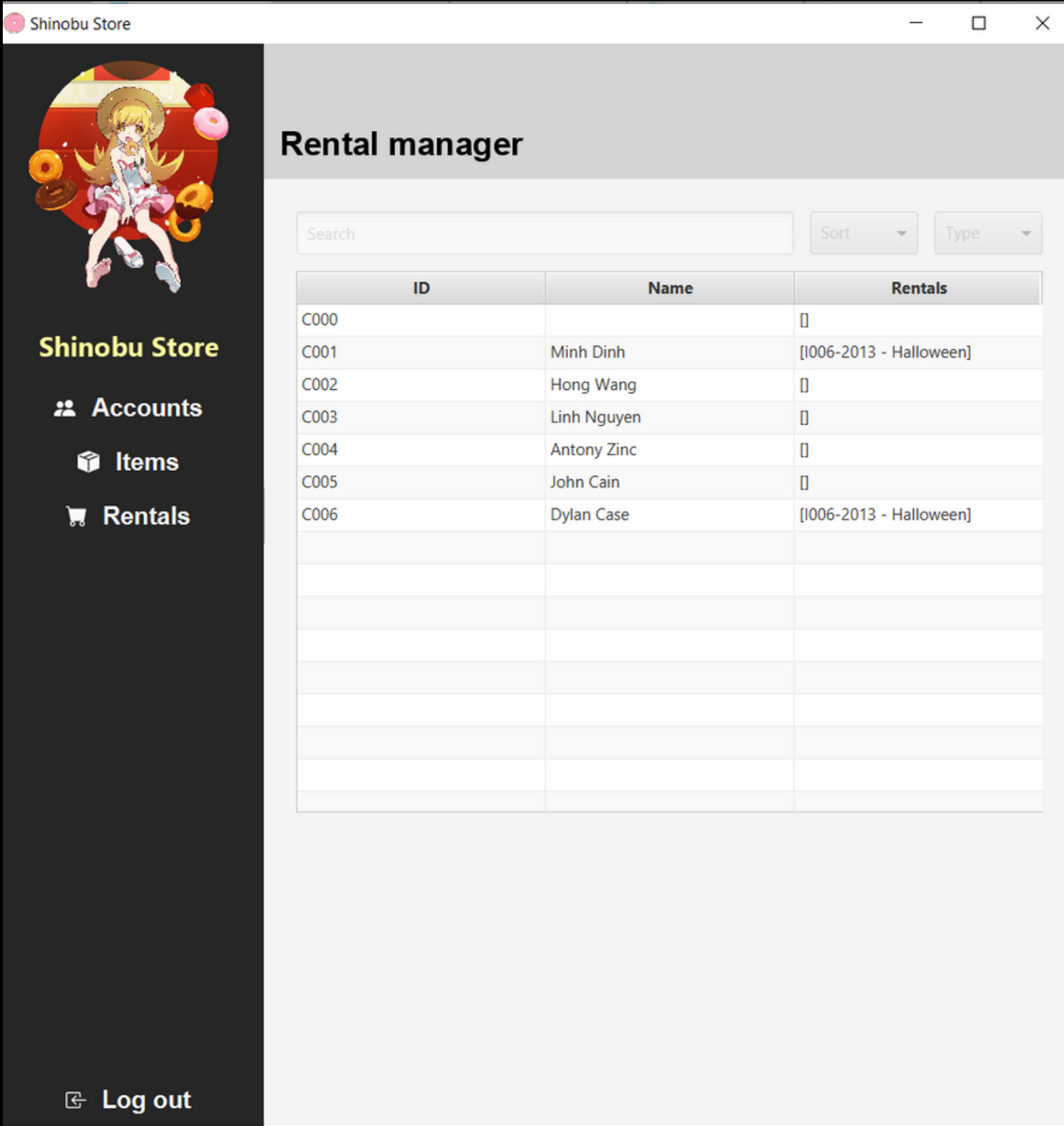
GUI - addItem.fxml



Features:

- Pop up when the 'Add' button is clicked, allows the admin to add more item along with their info into the database.

GUI - adminDashboard.fxml



Features:

- This page corresponds with the 'Rentals' button, displays all users that currently have any item rented.

THANK YOU
