Java Programming

CSCI-2210-01

The Knicks Team

Bachelor's in Computer Science and Cybersecurity

Final Project Report

University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING, West Haven, CT

Spring 2021

Submitted To: Prof. Reza Sadeghi

Contents

1 Overview

1.1 Team

Name of the Team: The Knicks

Members:

1. Andrew Mahr (Team Head) - amahr1@unh.newhaven.edu
2. Jake Intravaia (Team Member) - jintr1@unh.newhaven.edu
3. Andrew Manfredi (Team Member) - amanf1@unh.newhaven.edu
4. Spencer Wondolowski (Team Member) - swond1@unh.newhaven.edu
5. Antonio Lopez (Team Member) - alope7@unh.newhaven.edu

Github Repository Link: https://github.com/acompsec1/CSCI-2210-Java-Warehouse-Managment-System.git

1.2 Objective

The goal of our project is to create a Warehouse Management System that allows users to borrow/rent/favorite items that are stored within the warehouse. Admin users will have the ability to add/delete and edit users, items, and favorites; as well as accept or reject borrow/rent requests. All of these functions will be able to be carried out by the user through a clean and easy-to-use Graphical User Interface. All of the changes made to users or items or requests within our Warehouse Management System will be added into our SQL database and stored for future use by the user.

1.3 Warehouse Management System Outline

1. Login Portal
   (a) The first interface of the warehouse database management system will be a login portal for registered users of the website, such as admins or regular users. To log in, a user will enter a username or password, which will be verified by checking for the account information within the SQL database. Additionally, there is a registration page for users to sign up and create an account. Based on the user access level (Admin or User), upon login there will be a different page displayed based on the restrictions for each user status.
2. SQL Database

   (a) This database will contain data used throughout the management system such as usernames, password hashes, information about items, borrow requests, and items favorited by the user.

3. Admin Dashboard

   (a) This interface will contain the buttons and functions need for Admin level users to create, edit, and delete items and users. It also allows them to accept or reject borrow requests. The Interface is the main dashboard for if the current user signed into the program is registered as a Admin. Admin Users have the authority to add items to the database and also delete said items from the database. The Admin Dashboard differed from the main User Dashboard by adding the extra fields for the User to add and to search for an item in the database which can be viewed through the tables directly in the dashboard.

4. User Dashboard

   (a) This page will be the dashboard for non-admin users. It will allow them to select buttons in order to show all the items available in the system, make a borrow or rent request, see previous borrow history, and make a list of favorite items.

2 Modules of our Warehouse Management System

2.1 Structure of our Database

During our project, we used MySQL and MySQL Workbench mys to work with our SQL database. MySQL Workbench allowed us to quickly alter our database structure if we needed to add more information.

2.1.1 Primary Keys

- FAVORITEID - Unique identifier for each f̈avoriteïtem in the table
- ROLEID - Unique identifier for each role (admin or user)
- ID - Unique identifier for each user in the users table
- BORROW_REQUEST - Unique identifier for each borrow in the borrowed_items table
- ITEMID - Unique identifier for each item in the items table

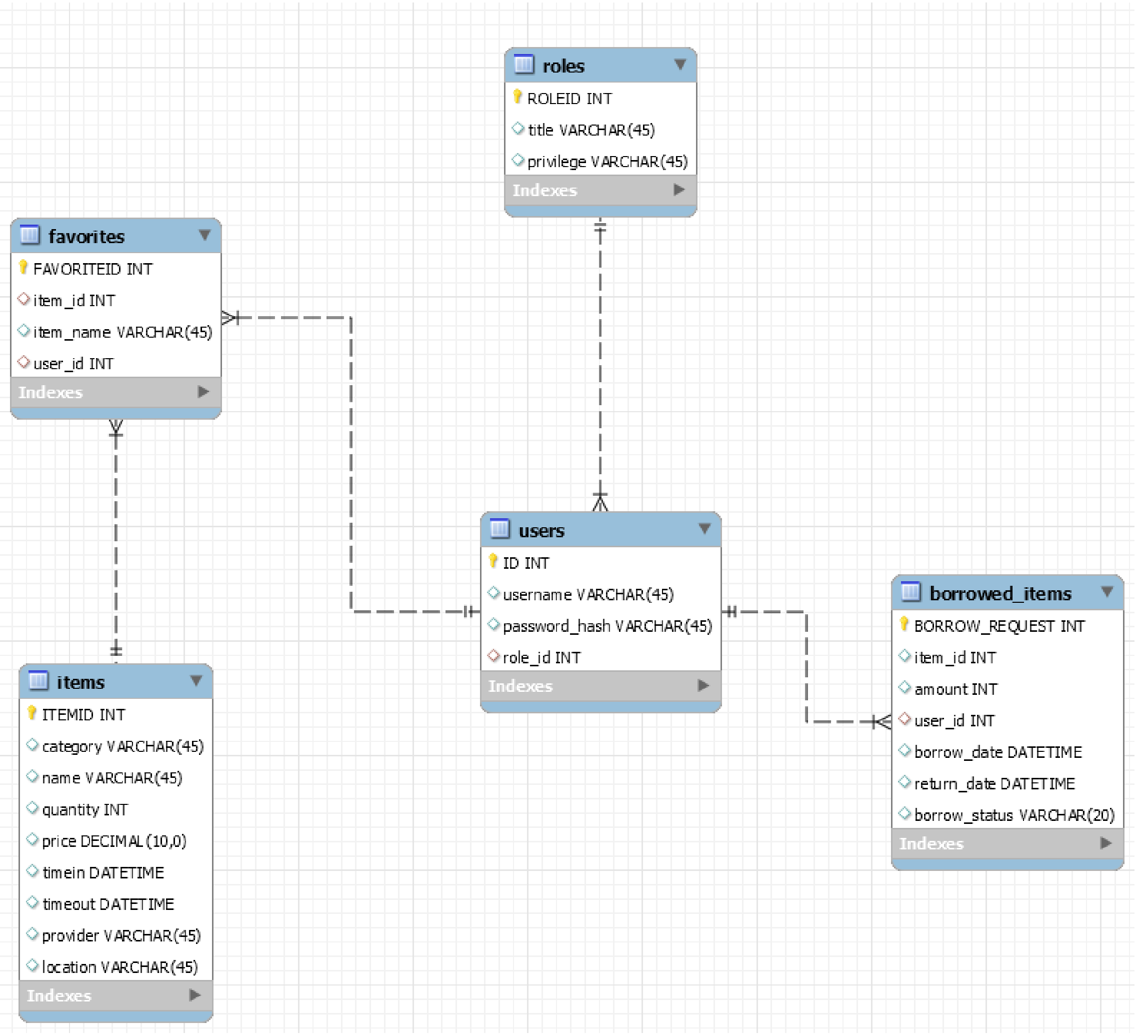


Figure 1: Our Database Structure

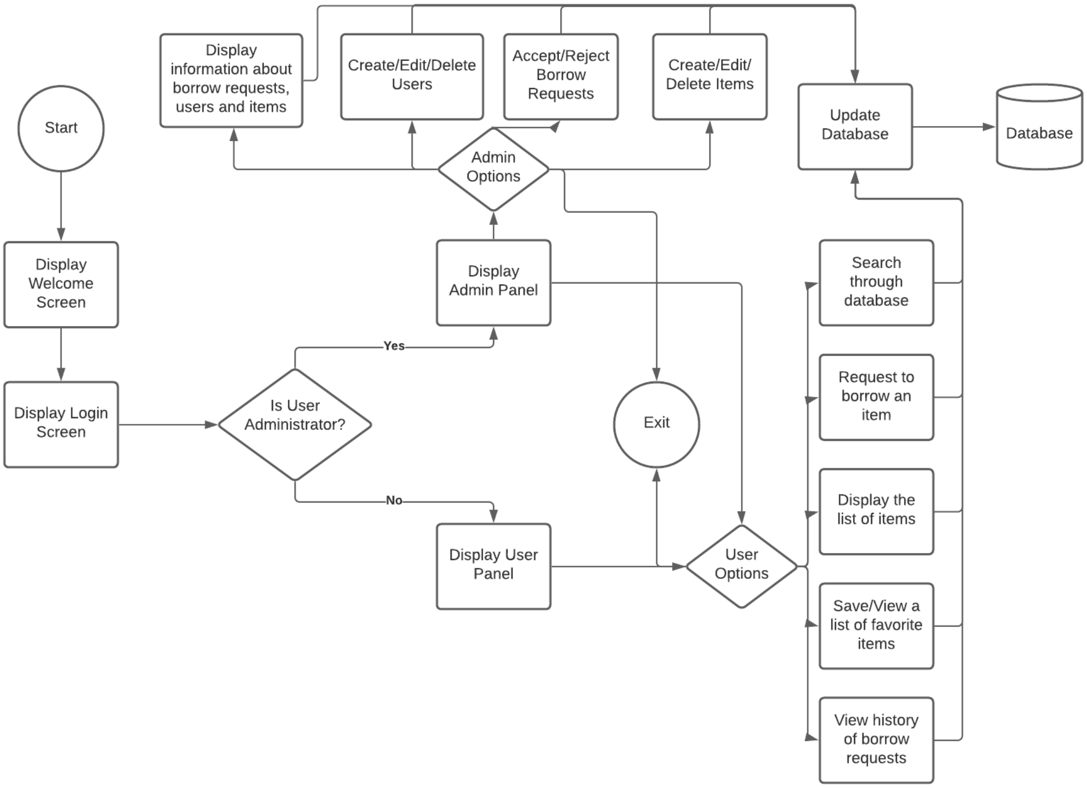

Start
Display Welcome Screen
Display Login Screen
Is User Administrator?
Yes
No
Display Admin Panel
Display User Panel
Admin Options
Display information about borrow requests, users and items
Create/Edit/Delete Users
Accept/Reject Borrow Requests
Create/Edit/ Delete Items
Update Database
Database
Exit
User Options
Search through database
Request to borrow an item
Display the list of items
Save/View a list of favorite items
View history of borrow requests

Figure 2: Process of our Warehouse Management System

2.1.2 Foreign Keys

- **(Users)** role_id - **role_id** is a foreign key connecting to the roles table, providing us information about the users privileges (0 = user, 1 = admin)
- **(Borrowed_Items)** - **user_id** is a foreign key connecting to the users table, allowing specific users to create their own borrow requests
- **(Favorites)** - **item_id** is a foreign key connecting to the items table allowing users to select a specific item they would like to add to their favorite items list for ease of access, **user_id** connects to the user table and is used to identify the user that added the item to their favorites

2.2 The Process of our Warehouse Management System

When the program is launched, the user is brought to a login screen. If the user already has an account, they can log in to the Warehouse Manager by entering their username and password. If the user is an administrator, they will be directed to the administrator dashboard. If the user is not an administrator, they will be taken to the user dashboard. Any user has the ability to log out at anytime.

The user has several options from which they can choose:

- Search the database for item IDs
- Fill out a request to borrow an item
- Display a list of items
- Add an item to their favorites list, which is fully accessible
- View the history of their borrow requests

The administrator user has all of the same options as the standard user, with several additions:

- The ability to add, delete and edit users within the database
- The ability to add, edit, and delete items.
- The ability to accept or reject borrow requests.

When any of these options are chosen, the function updates the database to reflect the user's choice.

2.3 Graphical User Interface Design

For the Graphical User Interface, we decided to go with something that is simplistic yet very intuitive to use. The GUI of both the Login and Registration page is a modern design that was laid out with the intent to prevent confusing and/or overwhelming the User. There are four simple interactive widgets on the Login screen which are clearly labeled for the ease of the user's experience (UX). We made the buttons a dark green to have them stand out more against the dark background of the screen, which makes them clearly visible to the user. As for the Registration page, we decided to keep the same theme as the Login page and we just had to add in a fifth widget to the screen which confirms the password that the user chooses. That is the only new feature added to the screen; otherwise it is exactly the same with minor scaling and text differences to fit the new widget added to the screen. One thing we also added on both of these screens is the widgets that the user has to interact with in order to either log in or register. They are contained inside of a rounded box with a different color styling than the rest of the screen, showing the user that those are the main interactive widgets for the respective screen.

As for the User Interface design for the home page for the application which we refer to as the Dashboard, we wanted to keep with a similar color styling to show continuity between the screens for the application. For the Dashboard, we utilized the JavaFX frame known as a border pane, which allows us to just keep our screens more uniformed and clean. On the left-hand side of the screen, we have the list of functions that a typical user may utilize such as a button to add items to the database and a button to delete items from the database. In the right-hand portion of the screen we have a table that is where items are displayed when queries are run. This is naturally a different color than the rest of the screen for the user to easily differentiate between the left-hand side menu and for the user to be drawn to the table where all of the important information will be displayed in.

2.3.1 Logo Design

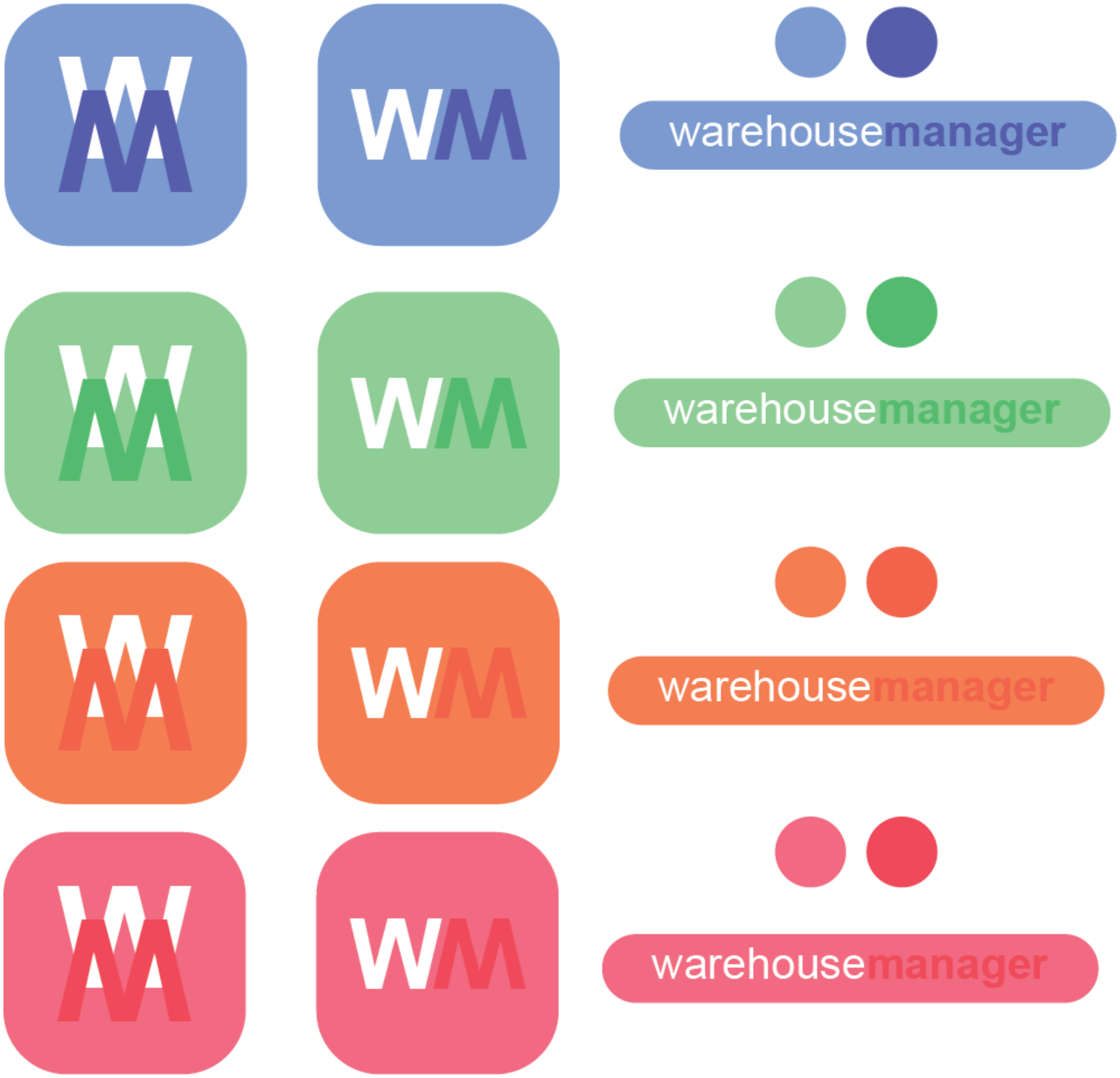
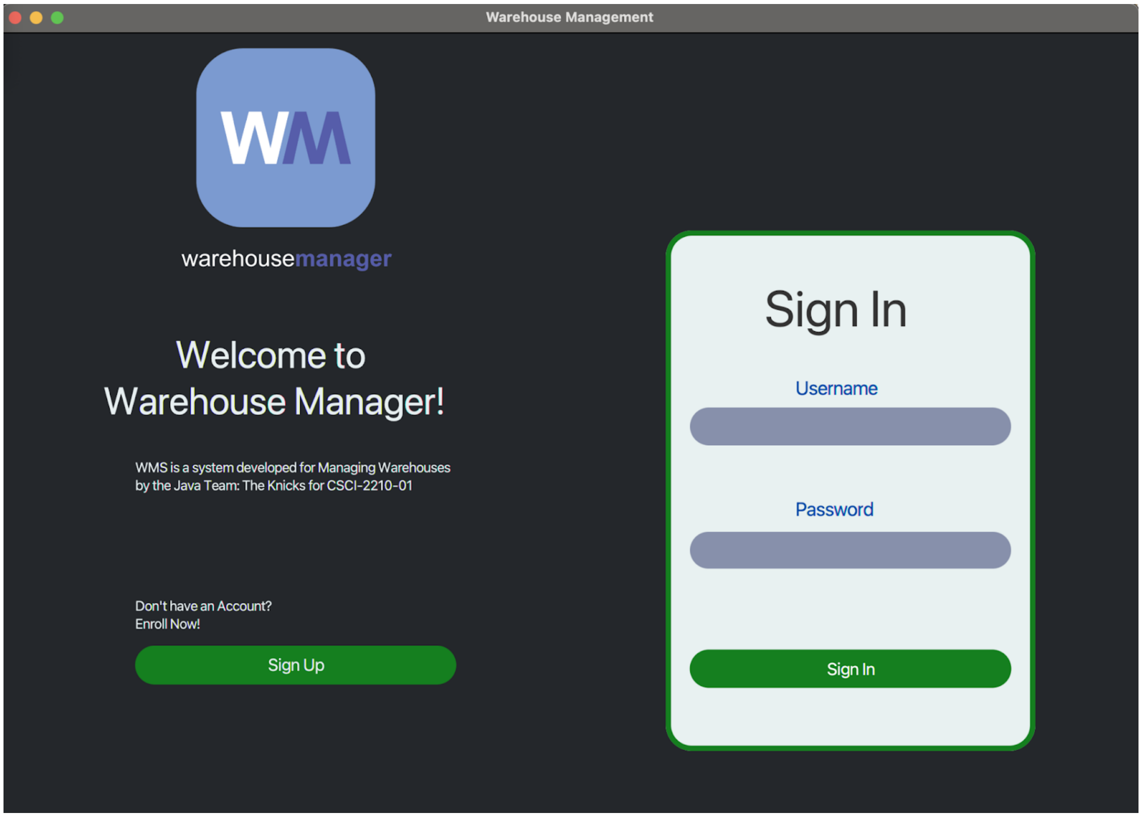


Figure 3: Our Logo Designs

These logo concepts were designed in *Adobe Illustrator 2021* Adobe [2021a]. We ultimately decided on the royal/navy blue combination for our logo design.



Welcome to Warehouse Manager!

WMS is a system developed for Managing Warehouses by the Java Team: The Knicks for CSCI-2210-01

Don't have an Account?
Enroll Now!

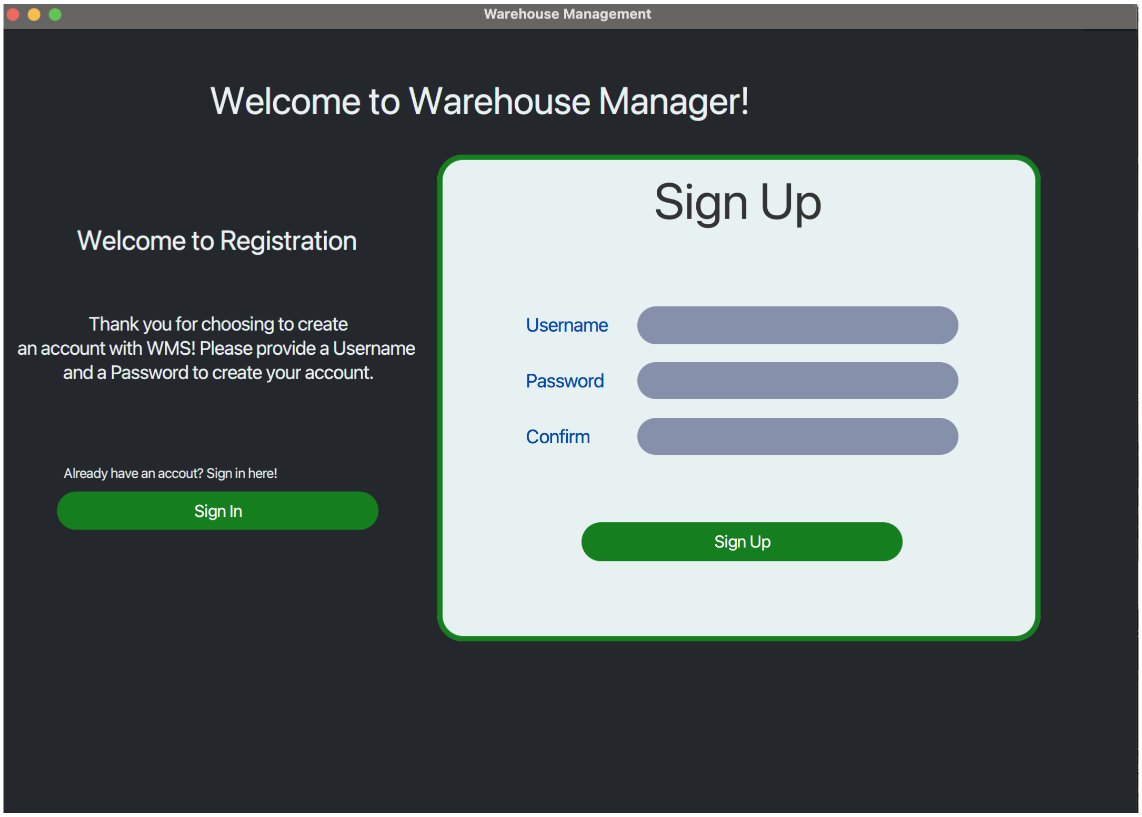
Figure 4: Our Login Screen

2.3.2 Login Page

The login page (Figure 4) welcomes the user to the application and gives them the option to either sign in or sign up. The login page was designed using SceneBuilder Gluon [2021], a program whose purpose is to enhance the workflow of creating JavaFX-based applications. This page was designed using the drag-and-drop functionality. Once the user enters their username and password, our code verifies that the user exists within the database before transitioning to our loading screen. The verification process involves checking if the username and hashed passwords match within the database.

2.3.3 Sign-up Page

The Sign Up page (Figure 5) provides users with the opportunity to create a user account. The user is asked to enter a username, and a password and when the Sign Up button is clicked, the code takes the input fields and verifies if the username already exists. If the username already exists, the user is notified and if prompted to change the username. If everything is successful, the user is notified of the account creation and then they are able to sign in on the Sign In page with no problem. It is important to note that the inputted passwords are put into an MD5 hashing function and are not stored in plaintext. This ensures account security as there are no functions in the code that are able to decode MD5.



Welcome to Warehouse Manager!

Welcome to Registration

Thank you for choosing to create an account with WMS! Please provide a Username and a Password to create your account.

Already have an accout? Sign in here!

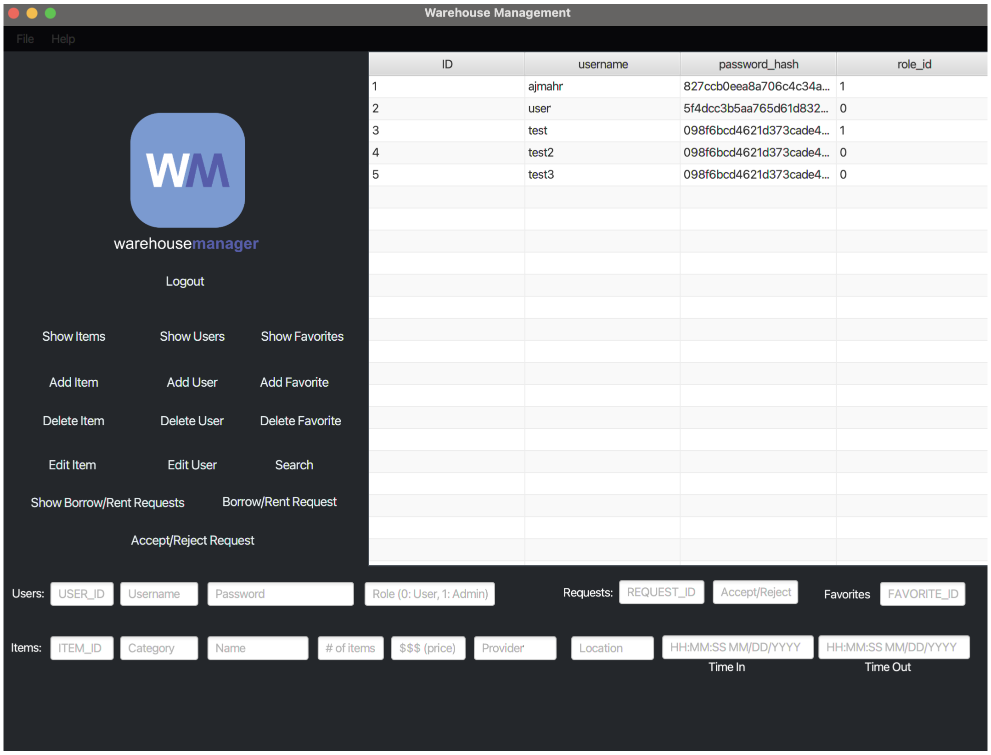
Figure 5: Our Sign Up Page

2.3.4 Admin Dashboard

This interface (Figure 6) contains the buttons and functions needed for Admin level users to create, edit, and delete items and users. It also allows them to accept or reject borrow requests. Admin users have the highest authority within the program and thus we have checks to make sure that all information properly matches before the deletion of an item. The Admin Dashboard differs from the main User Dashboard by adding the extra fields needed to create items and users.

2.3.5 User Dashboard

This interface (Figure 7) contains the buttons and functions needed for main users. If the user signed in is not at the admin level, they will be able to see lists of the available items, their favorite items, and is also provided the options to request to borrow items, check borrowed history, and to add or delete items to their favorites.



ID	username	password_hash	role_id
1	ajmahr	827ccb0eea8a706c4c34a...	1
2	user	5f4dcc3b5aa765d61d832...	0
3	test	098f6bcd4621d373cade4...	1
4	test2	098f6bcd4621d373cade4...	0
5	test3	098f6bcd4621d373cade4...	0

Figure 6: Our Admin Dashboard

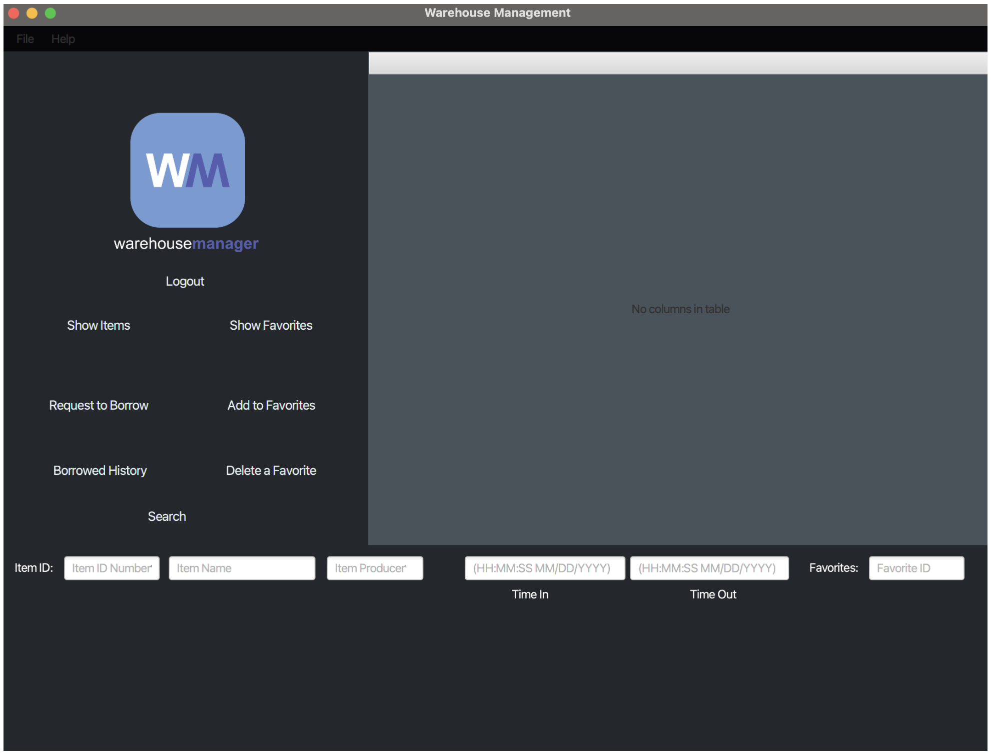


Figure 7: Our User Dashboard

3 Discussion of New Features

Throughout the course of our project, we implemented many different features. Below, each feature will be discussed.

1. The Sign up and Sign in Option
   (a) When entering the application, the user will be given the option to sign up or sign into an existing account. The sign up feature requires the user to create a user name and a password. The user is then notified if the username already exists or if the account was created successfully. The default access level for users that use the Registration page is "User". The sign in feature subsequently requires the user to enter a username and a password that they have previously created.
2. MD5 Hashing of User Passwords
   (a) To provide enhanced user security, the team members decided to hash user passwords using MD5 (Hungund [2021]) and store these values in the database instead of plain-text passwords. In order to verify that passwords match, we take user input, hash it, and compare it to the stored data.
3. Loading/Exit GIF for starting and closing the application.
   (a) As an added feature to enhance the user's experience, (UX) the team will attempt to implement loading and closing GIFs as the system starts and closes.
   (b) The main loading GIF with the box was designed, animated and rendered in Blender (Foundation [2021]), an open source 3D modeling and animation software
   (c) The text loading GIF was designed, animated and rendered in Adobe After Effects 2021 (Adobe [2021b]), a video editing software designed for special effects and motion graphics.

4 Collaboration Method

4.1 IDE

During the course of our project, we used the IntelliJ IDE (IntelliJ [2021]) to generate, build and execute our code.

4.2 Pushing to GitHub

During the course of our project, we used GitHub to collaborate on the code and divide the work. Upon completion of the project, the final code will be pushed to GitHub and will be available for anyone to clone and commit any necessary changes. (GitHub [2021])

References

Mysql workbenchdownload now ". URL https://www.mysql.com/products/workbench/.

Adobe. Adobe illustrator, vector based graphic design software. https://www.adobe.com/products/illustrator.html, 4 2021a.

Gluon. Drag and drop development suite designed for intuitive creation of javafx .fxml files. https://gluonhq.com/products/scene-builder/, 4 2021.

Bilal Hungund. Md5 hashing function made for java. https://www.geeksforgeeks.org/md5-hash-in-java/, 4 2021.

The Blender Foundation. Blender. https://www.blender.org/about/, 4 2021.

Adobe. Adobe after effects, video editing and motion graphic animation design software. https://www.adobe.com/products/aftereffects.html, 4 2021b.

IntelliJ. Indie development environment for java. https://www.jetbrains.com/idea/, 4 2021.

GitHub. Github is a development platform inspired by the way you work. https://github.com/, 4 2021.