Itay Getahun

1070 sela College

Itay Getahun



**Signing in codes**

Verified Employee ID: 000000000

Password = 12345

Verified Costumer: 999984131

Password = 1234ABCD

GitHub link   
<https://github.com/itayG98/Libary_App_Solution>

UWP Libary application Assignment

**Overview**

My application has 2 types of users first Costumer and second is Employee and can handle its tasks using MookData class and Logic class.

Features:

* Signing up Costumer / Employee
* Logging in into appropriate page
* Choose library item to see its Fields
* Borrow / Buy /Return specific item
* Change Password
* Change user’s details

Employee’s extra features:

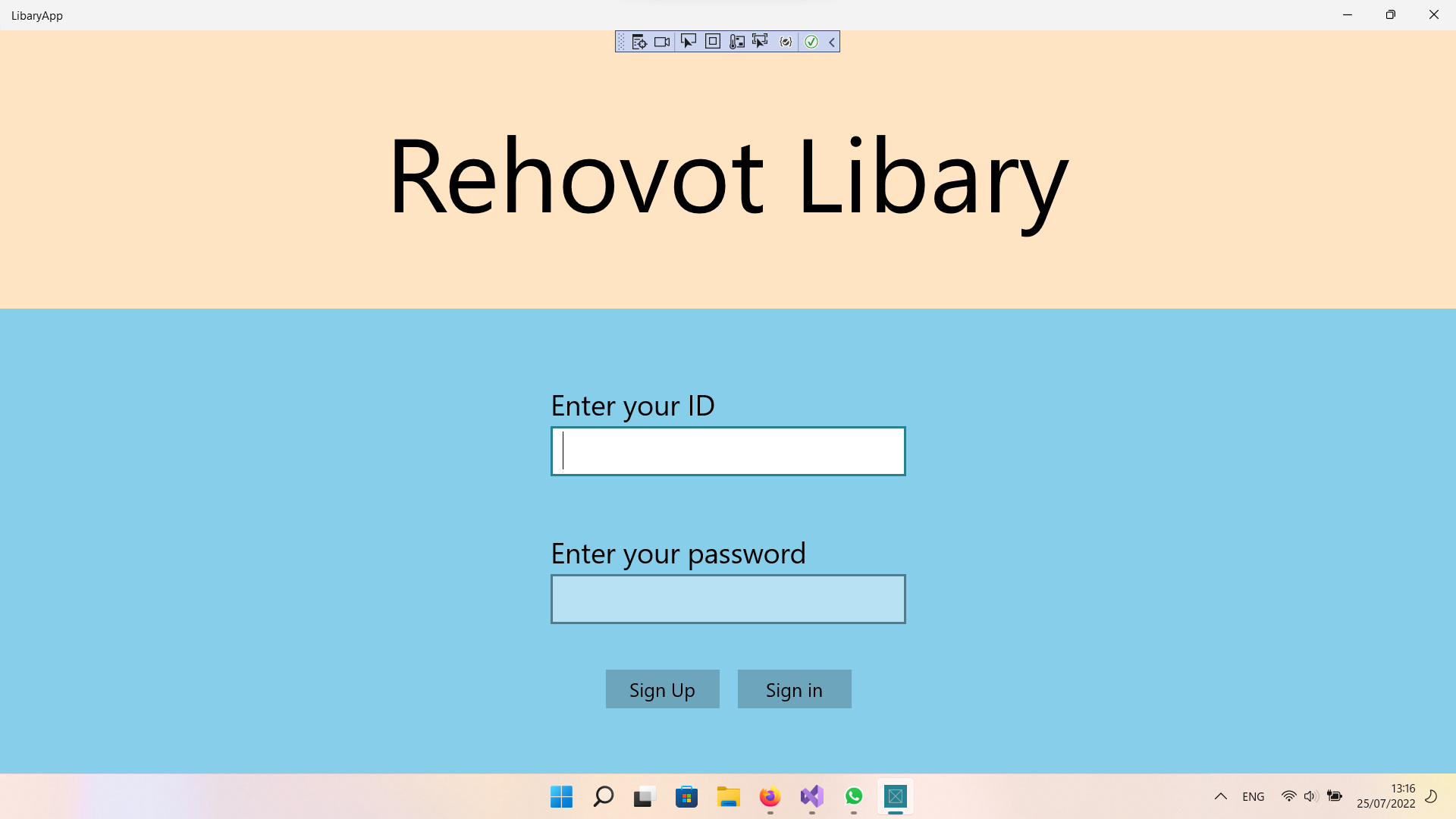
* Add library items to the system
* Change person’s discount or details
* View all Borrowers / costumer /Employees / all Users
* Delete from the system User except itself
* Delete Book from the system

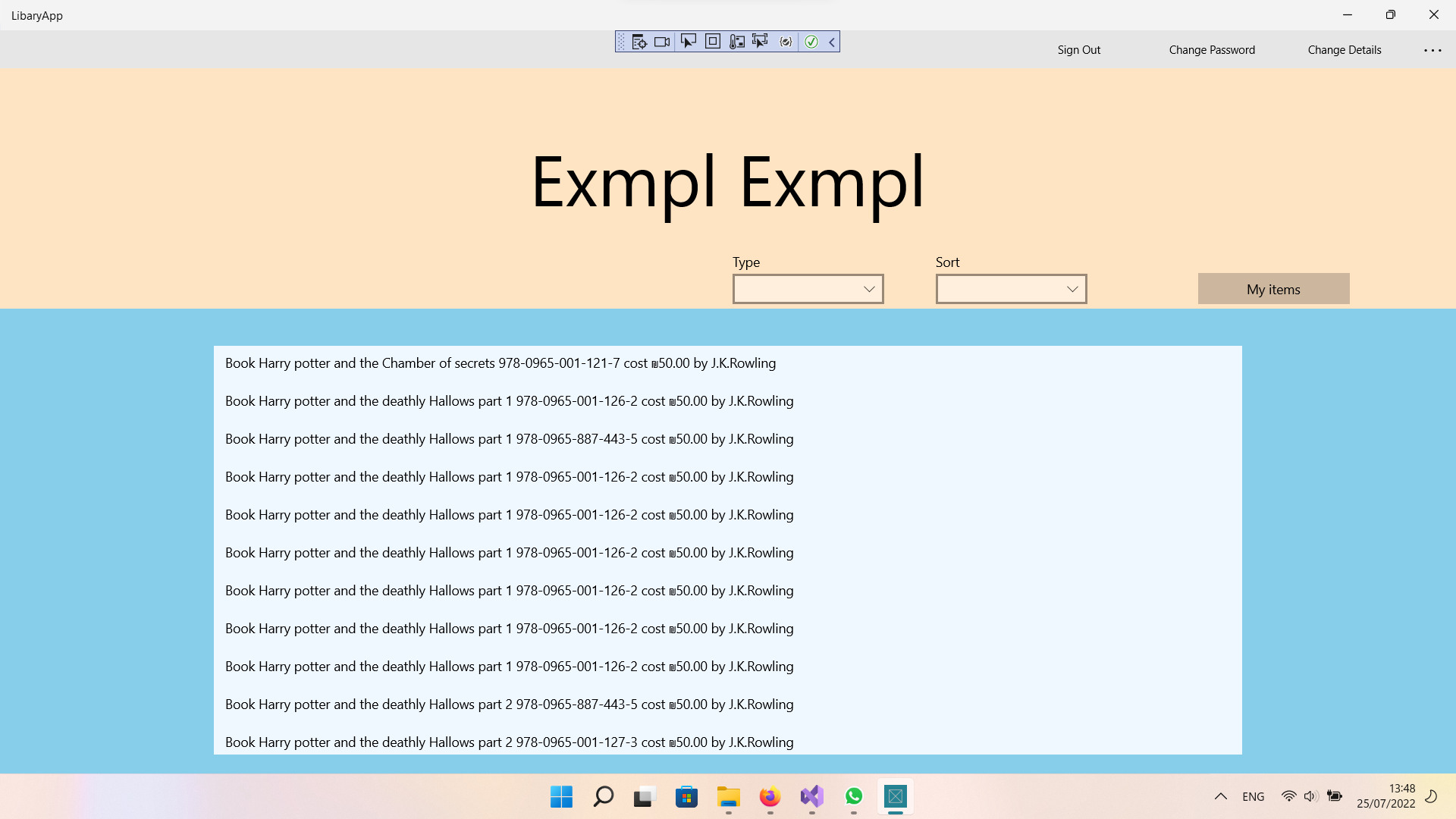
Sign in Page

My main page checking if User’s ID exist and check its password validity.

If so loges him into appropriate page (Employee page or Costumer page).

The user can also sign-up Navigating to another page.





Costumer Page

The Costumer page let the user view only the available book or journals and sort in alphabetically or by price and show only specific type

When user want to see his borrowed items, he has a specific button.

When clicking in an item the user gets into the item order page

Employee page

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

The Employee page has all Costumers functionality and can view all the system’s users include employees.

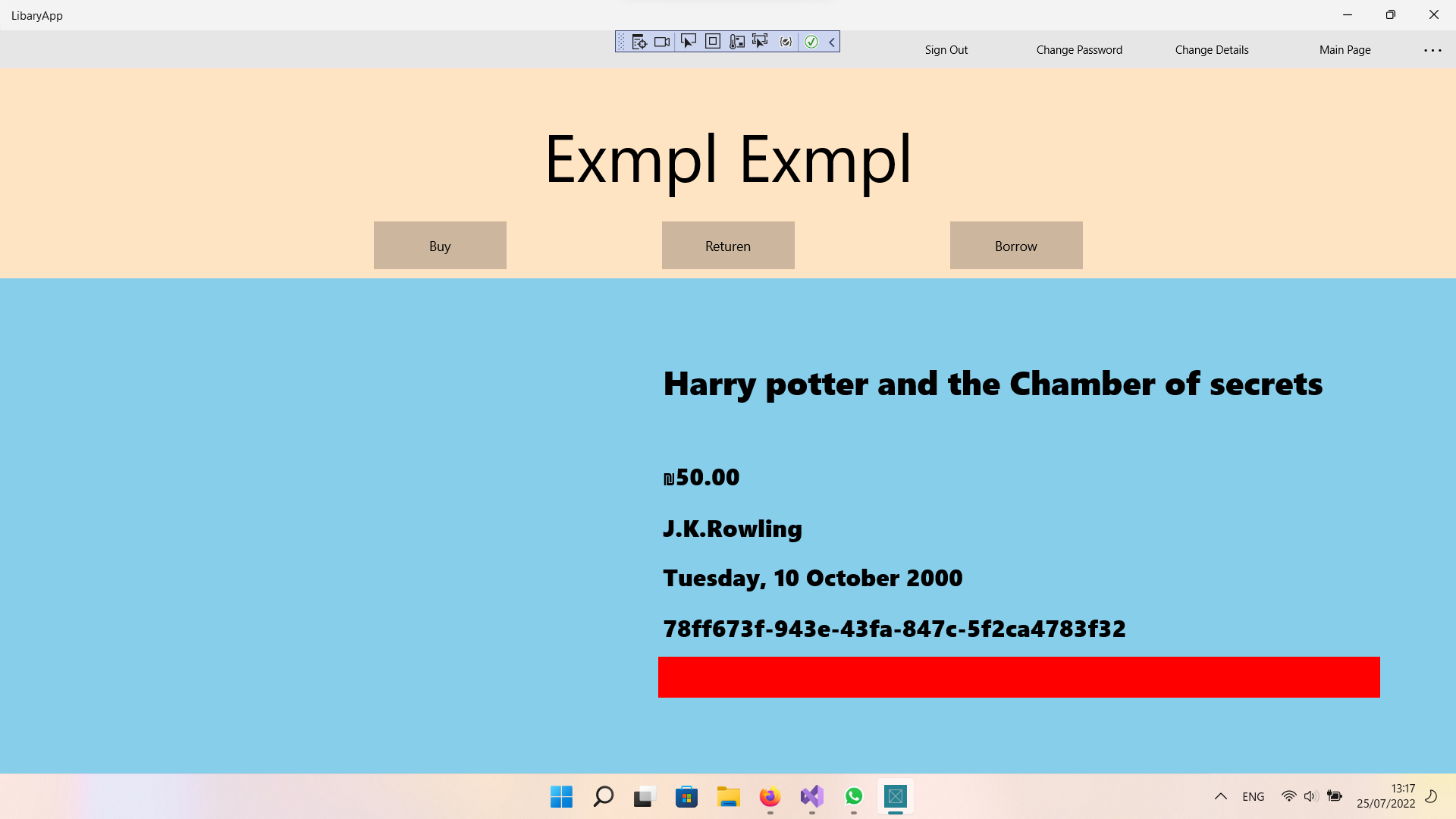
Important to mention that he can also view unavailable books.

When pressing on person the Employee can navigate into his page and change them.

The sort and type choosing is according to the left button

Where you chose whether to see users or library items.

Order Item Page



Any user can order buy and return an available item.  
Red means Unavailable  
Green means Available.

When borrowing or buying a message Dialogue shown.

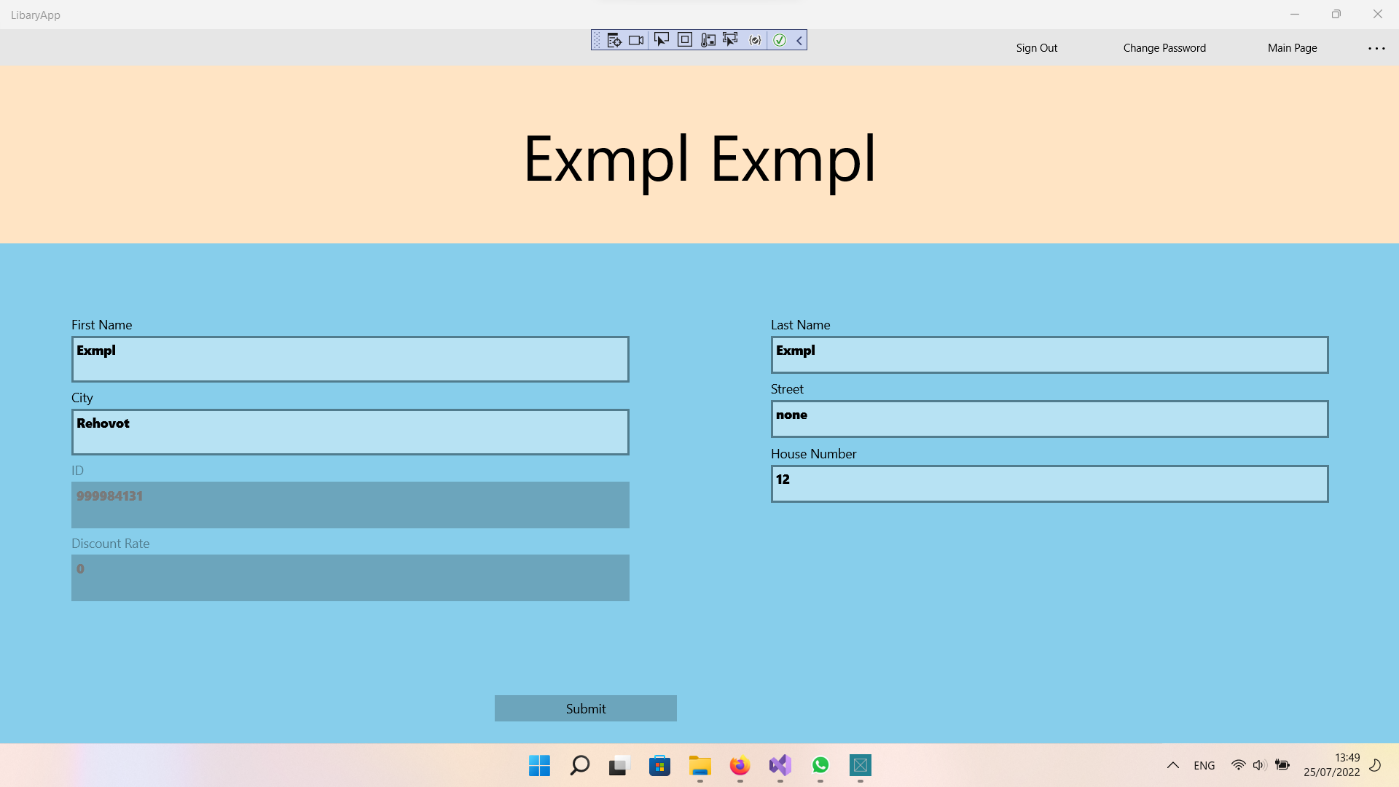
If employee press on change details, it navigates him to the items edit page

Else to the normal person edit details page.

Edit user Page

Any user can change its details and Employee can change any User’s details if navigated from his data page.

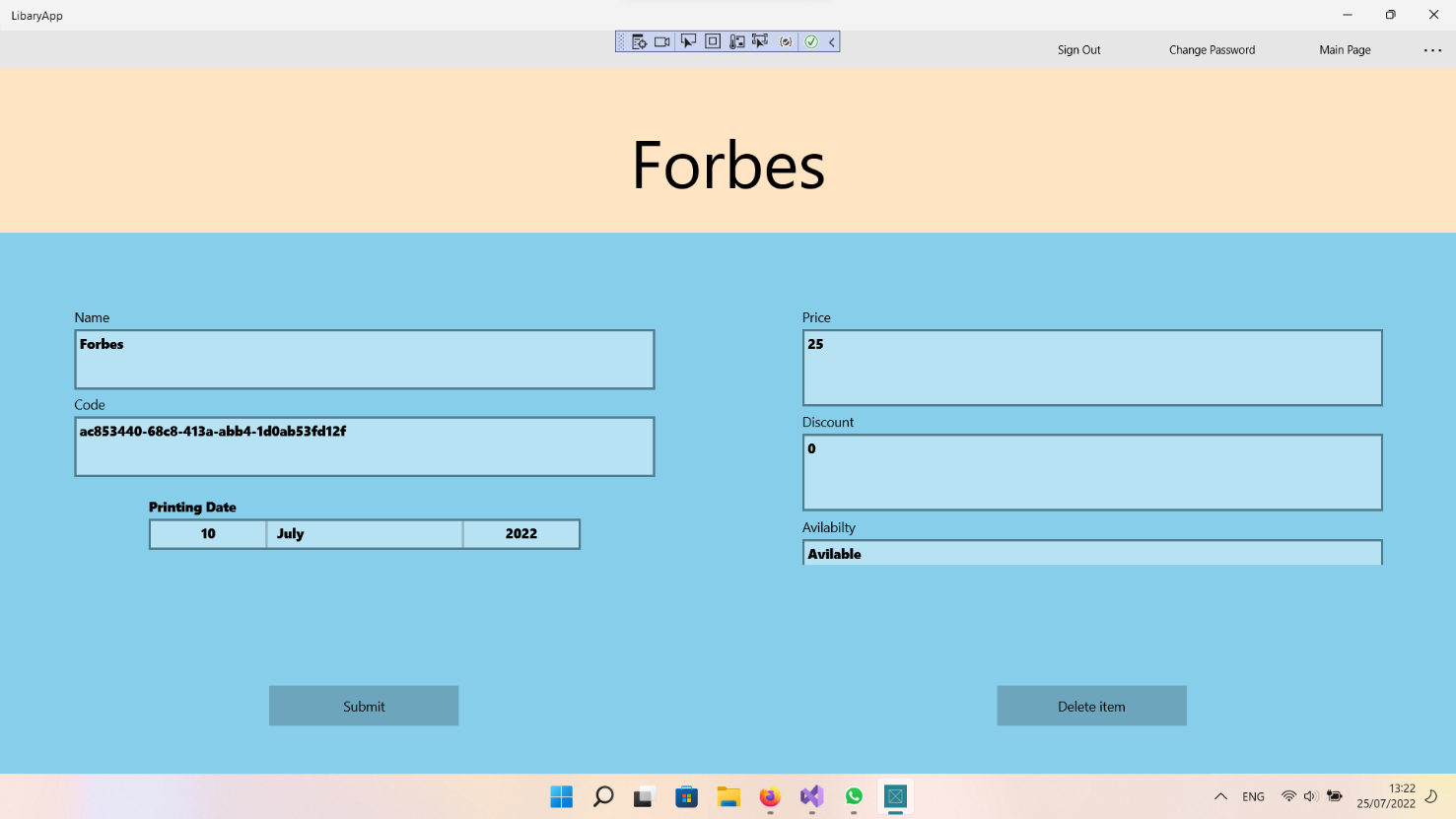
Changing ID is not available and only employees can change discount percentage (Up to 50 due to Person class in the model.dll).



Edit item page

Only Employee can edit Item access this page and change book’s data or delete the item

Navigate to this page from Order Item Page if the signed user is employee.



Sign up Page

User can navigate to Sign Up page through the sign in page.

After filling all the data and validate it the user can sign up as a user or as an Employee if he signs with “12345” password code.

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

App bar buttons

The app bar enable navigation though the application.

Change detail's action defined by the user.

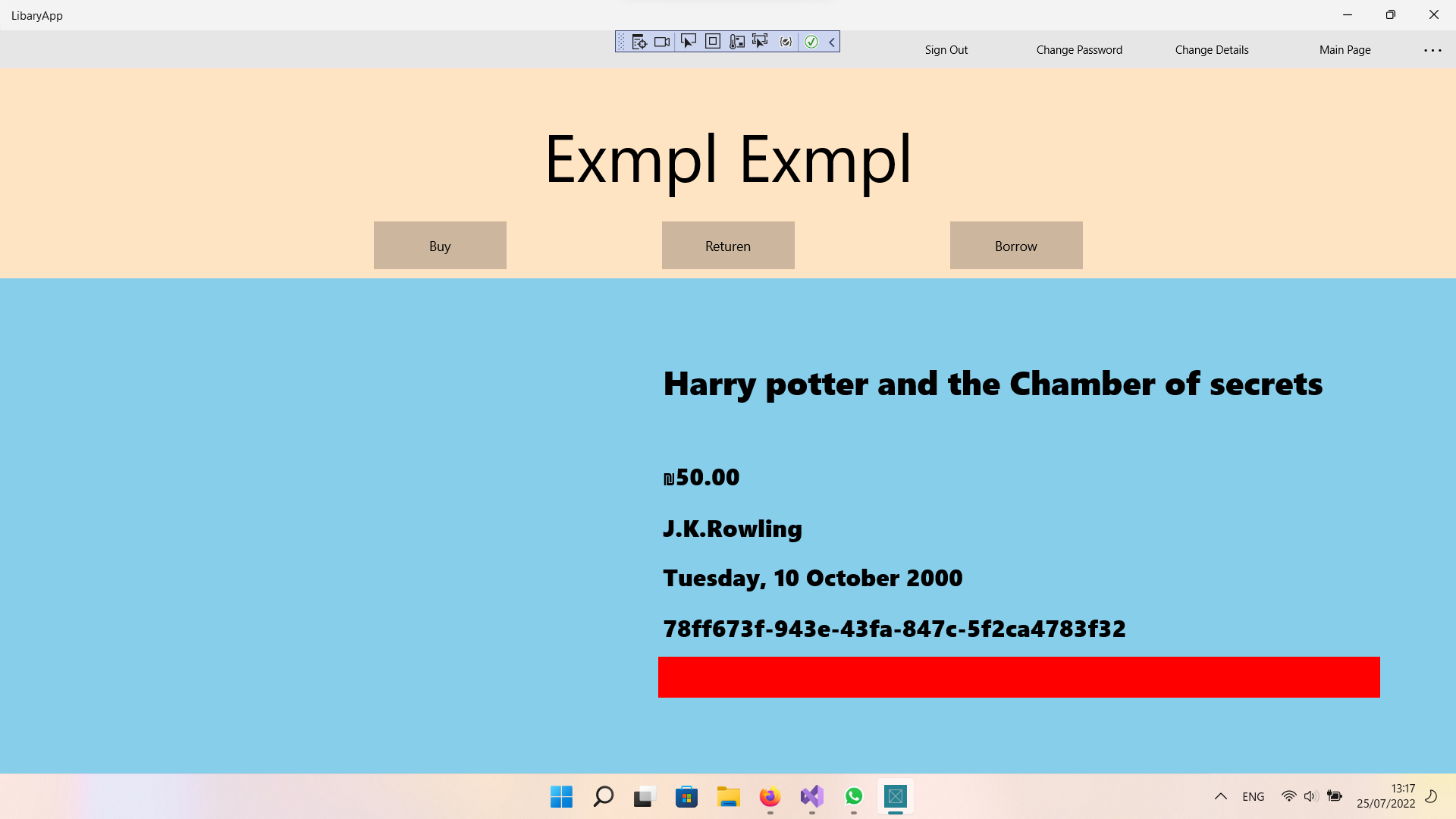
If it is a costumer it always navigate to the edit person page with current user.

Else it navigate to edit items if the current page is order page

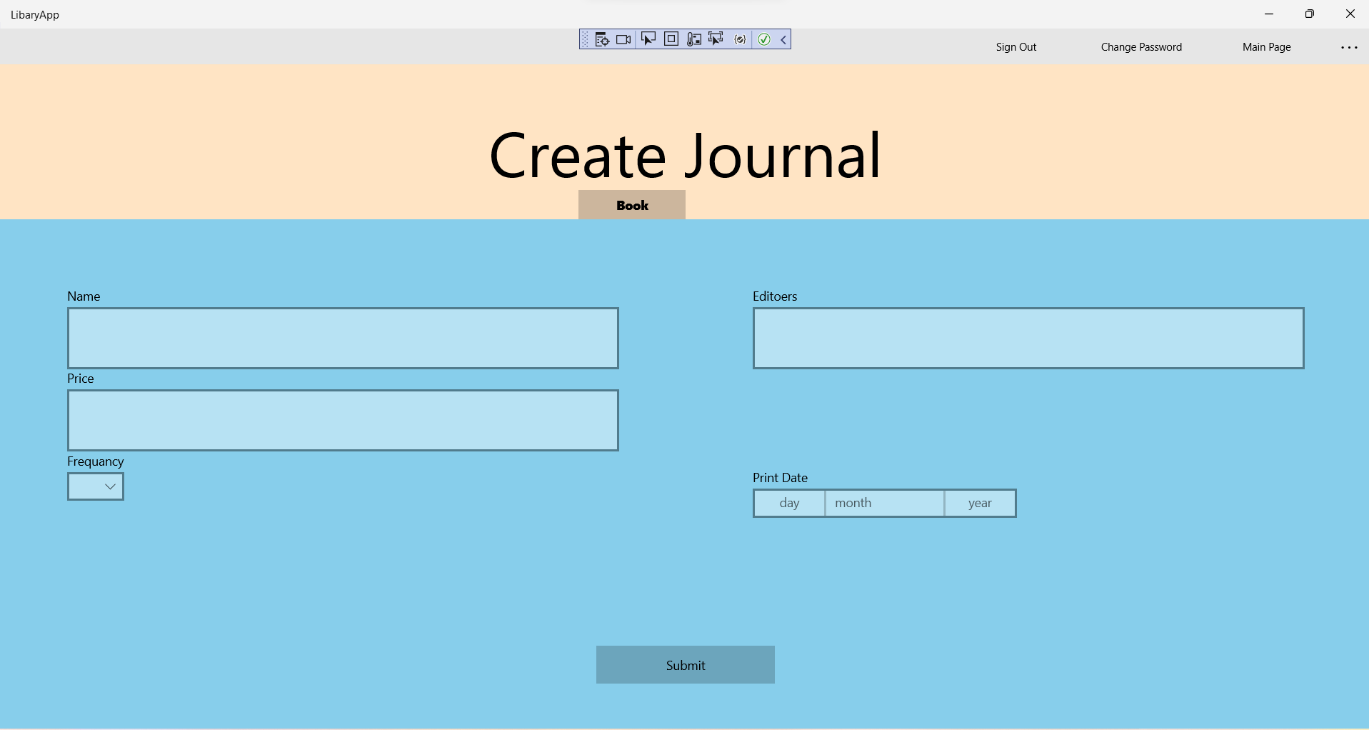
Or to edit other user’s details if the person details page is open

Add item is accesible only throuhg the employee page

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

Create Item Page



Only Employee can access Create item page

Under the headline you can click on a button and choose to create Journal or another library item

When clicking on submit it validate and create the item into the MookData class.

**Optional additions which were not included in current version**

**\*Duo to time shortage**

Features:

* Billing system and implementation of discount
* Adding more verified codes to publisher and countries
* Search by name of users or library items
* Add more profiles of user deriving from costumer’s functionality
* Add a data track of all transaction which made.

**Summery**

I developed 4 projects for my application

* Library model  
  class library for the objects needed in the library and their behavior.  
  This class library is dependent and does not work with any special references DLL’s.
* Logic and mook Data  
  class library for mook data and its Repository interface and logic object for managing the application .  
  This class library work with the library model.
* Library app  
  The UWP project having variety of UWP pages the user can navigate through  
  This project familiar with both Model and Logic.
* Library app Tests  
  I made test methods to the class libraries projects and due to confusion, I made some methods as integration test (Testing how the Logic class and the Irepository integrate with the mook data class)

**Key fixes from zoom talk:**

I may split the library repository class into 2 classes for Person’s repository and Library item’s repository