Politecnico di Torino

Department of Electronics and Telecommunications

Project and Laboratories on Communication Systems

Technical Report

Mattia Mirigaldi: s288224@studenti.polito.it
Reza Khoshzaban: s288966@studenti.polito.it
Pouyan Asgari: s289607@studenti.polito.it

September 27, 2022

Contents

1 Introduction

This project aims to provide a dynamic inventory management system able to control variety of items and users, the functionalities are user dependent and for example customers cannot modify items or other users while administrators have full control of users and items and can assign multiple operators to their branches who would be in charge of the branch and the customers and items associated to that branch. This service is provided through 3 different interfaces that are totems, mobile application and a web page (used for management purposes).

This project provides the following services to users who are registered as administrators :

- User Management including removal,addition and modification of customers and operators registered through the respective administrator
- Item Management including removal,addition and modification of items registered through the respective administrator and renting/returning an item for a customer.
- Totem Management including removal,addition and modification of totems registered through the respective administrator
- Account management including modification of data such as password and username

The services provided to the operators of the branches are considered as following:

- User Management including removal,addition and modification of customers registered through the respective branch
- Item Management including removal,addition and modification of items registered through the respective branch and renting/returning items for a customer of the branch
- Account management including detail modification to information such as password and username

And the customers can access the services indicated below:

- Rent and return items from the respective branch
- List the items available in the branch
- Account management including detail modification to information such as password and username

2 Database

In this section are described the tables in the database created using microsoftSQL servers. The database consists of 6 tables for Users,Items and the totems.

2.1 Users

The users section of the database includes 3 tables for 3 different user types in "Admins","Operators",and "Customers". The hierarchy of privileges is as following: Admins -> Operators -> Customers.
The tables of the different user types are described below:

id	firstname	lastname	username	email	password	rfid
1	John	Doe	admin1	ad1@mail.com	ad1	1

admin_id	id	firstname	lastname	username	email	password	rfid	branch
1	11	John	Doe	operator1	op1@mail.com	op1	11	branch1

admin_id	opr_id	id	firstname	lastname	username	email	password	rfid	branch
1	11	111	John	Doe	customer1	cs1@mail.com	cs1	111	branch1

As seen in the tables, each administrator is capable of connecting to the operators and customers of its through the admin_id column in their respective tables, and each Operator have access to the customers registered by that operator only. The id of each user is unique and so is the user name of the users which are checked during registration and modification of the users by their respective admin or operator. The branch column is for the purpose of being capable of having more than one operator for a branch and being able to distinguish and manage each of them in case the user of the service considers using the service for a multi-section branch.

2.2 Items

The items section of the database is currently consisted of 2 tables in "Items" and "Books" which is extendable in case that the customer of the service wants to manage different or multiple types of items. The tables are considered as the following:

admin_id	*opr_id*	*cus_id*	*id*	*name*	*category*	*branch*	*rfid*
1	11	111	1111	Gone with the wind	books	branch1	1111

item_id	id	title	author	genre	publisher	date	location	description
1111	2	Gone with the wind	Margaret Mitchell	Novel	Company	1985	sec A1-shelf C2	A book

As shown in the tables, the "Items" table is a generic table in which any type of item can be added and managed with its unique administrator or operator which are connected to their items using the their respective id columns in the table while the "Books" table is an extension to the main table to include further information about the item which is connected to its specific item using the itemid column in the table. With this approach, the details table can be multiplied to be compatible with other types of items such as laboratory instruments and etc just by making new table with the itemid column included and the column names specific to that type of item.The cusid column in the items table is the column used upon "rent" such that the method used for this function replaces the data in the column with the id of the customer and changes it back to 0 in "return" function The management and data extraction of the tables is done using the INNER JOIN method of SQL.

2.3 Totems

This table is specifically created to associate each totem to a specific branch and and administrator to avoid conflicts like a customer of a branch being able to login to another branch's service. The table is as the following:

admin_id	opr_id	totem_id	macAddress	branch
1	11	333	00:00:5e:00:53:af	branch_1

The Mac address of the totem is retrieved upon the login method and checked with this table to confirm that the user trying to access the totem is related to that branch.

3 Back-end

The back-end of the project is consisted of MicrosoftSQL used as database and Flask property of Python as the server alongside HTTP Services from the flutter side plus a python script specified for totem in order to retrieve the RFID information from the RFID reader of the totem to the server which will be described in depth later on in this section.

3.1 General description

The management of the back-end in general is done in a way that each time a function is called from the front-end such as modifying or listing information from different tables of the database, The function awaits a method from the HTTP Service of the related section and forwards information from the front-end if needed. The related method in the HTTP service method retrieves the forwarded information and sends them to the server via a specified link referring to a method in the python code. The server is consisted of methods specific for each type of function and is capable of receiving data as a body from HTTP services and start the method related to the requested function. The methods are consisted of specifications and SQL queries to manage the database or retrieve or add data to its tables according to the requested function using the "pyodbc" library which makes connection to a specific database possible. The response from the python server back to HTTP service is done using JSON properties and the "jsonify" function.The retrieved data is managed and either stored or sent to the front-end with a navigator in order to proceed with the request.

3.2 Multiple user login management

multi-user login in possible in this case thanks to unique HTTP links sent to the server upon function calls. This is done in a way that each user has a unique link consisted of its id and in the case of operators and customers also their respective branches and their administrator id or/and operator id if necessary. This property is also used as a way to send data from HTTP services to the server since it is possible to convert data received from the HTTP link to strings of variables in the python script and use them if needed.

3.3 Login functions

Login functions are divided in 2 parts since this service provides login using credentials for all its interfaces and login using RFID for the totem and the mobile app.

3.3.1 Login using RFID

In this function, the RFID read by either totem or mobile is forwarded from the RFID read method and a "SELECT * FROM [table] where RFID = (?)" query is executed on the table related to the privilege of the user who is trying to login.The user is navigated to the home page in case of a non-null return from the database and the user data is sent to the HTTP service using a dictionary package which includes all the information of the user.A flag is sent in case the RFID doesn't match with any user to show a "User not found" error in the front-end.The table name is a variable which can change according to the user type selected in the login page of the front-end in case of the web application or the page from which the user is trying to login in the totem or the mobile application. The idea of having a direct login for the customers in the Totem instead of a drop-down menu like the web application is to simplify the totem interface as much as possible for customer satisfaction purposes.

3.3.2 Login using credentials

This function is almost identical to the RFID login method with the difference that instead of the RFID code being forwarded, this time the username and password of the user is retrieved form the front-end and the query is run with username and password as its specification. Login with credentials is available in all the interfaces in case of RFID reader malfunctions.

3.4 Settings functions

The settings function for the users consists of two methods. First is to list the current information of the user and the second is to retrieve the inserted new information of the user and replace them in the database. Since the username in each table is considered to be unique, a real-time check is performed each time the user types a letter in the specified box which performs a "SELECT * FROM [] WHERE username = (?)" query each time a letter is inserted to avoid repeated "username already used" errors upon pressing SAVE button and instead warn the user in real-time while typing the new username.This property is also used further on in adding users. Modifying the user data in the settings page does not include ID or RFID of the user for privacy reasons since only the superior of the user should be able modify their IDs if necessary. As mentioned above, The user data is sent to the HTTP services as a dictionary and thus saved in a vector. They are also updated with the new information after applying the settings.

3.5 List functions

Lists are one of the most important parts of this service since they are needed in all the interfaces and both items and users and the most important property of them is for them to be dynamic in size and the information that they present.

3.5.1 Users

The list of the users is available for the admins and operators with this difference that in the operators case, the users listed are the users that are customers of the same branch that the operator is part of ,but the administrator can list the customers and operators of all branches using detailed query checks with dynamic user type and branch selection which are sent to the server and the "SELECT" queries are run with dynamic tables. in a more detailed way, the type of the user that the administrator selects to list is forwarded to the server and dynamic table selection for the query is done in the list method using a variable instead of the name of the table. Since the admin is connected to all its users with its ID as described in the database section, the admin id is also considered as a WHERE statement in the query and the id is forwarded with the link form the user buffer which was filled with the admin's information upon login.In this way, one admin cannot have access to other admins' users.

3.5.2 Items

The listing of items is a bit more complicated comparing to users part since it is consisted of two tables in the database in the items and the details table which in this case is called books table. In order to fetch the needed data from both tables with just one query, a INNER JOIN function is used which connects the tables on the item id column. The list is used in the items menu in mobile app and web application and in the pending list in the totem case which is available for the operators. In the other instances of the lists in web application, the lists are shown according to the selected branch. The name of the branch is taken from the dynamic value of the drop-down menu and is used as a WHERE statement in the query implemented in the python server method.

3.6 Insert functions

Insert functions are used when the user attends to add a user or an item which is available in web app for the admin and all the interfaces for the operator. The filled list in the front-end is sent as a body to the server and using an INSERT query, the new user or item is added to the database. As mentioned before, in case of adding a user or an item without RFID in web app or mobile, the assigned operator will be able to add their RFID from the pending list available in the totem. Real-time username check is also performed in this stage while adding a user as mentioned before.

3.7 Remove functions

Remove function is a simple function using a "DELETE * FROM [] WHERE" query which is used in all the interfaces taking in the RFID of the intended user or item or as an alternative to be used in case of RFID reader malfunction or in the web app, it is possible to remove the user or item using their user name in case of users or title and author in case of items.

3.8 Modify functions

Modification is one of the important parts of this service which is provided for the case of both users and items. This function works similar to the settings function but with the difference that, in this case, a user can modify the information of the other users related to them. In case of operators, an operator is capable of modifying customers and items registered by them to their respective branch and in the case of administrators, they can modify the information regarding to the customers,operators and items related to them in all branches. This section includes modifying all the details of the items and users including their RFID so that the admin/operator can insert the RFID also manually to a user. The modification method uses UPDATE query on the intended table by taking in the new information as a body with a WHERE statement of the user or item's ID to identify the object in the table.

3.9 Pending list functions

The pending list, lists the items which are added using web page without the RFID or with the mobile phone using the "Add without RFID" function. By clicking on the listed item, the user is taken to an ADD RFID page from which the scanned RFID is sent to the considered function to add the scanned RFID to the "rfid" column of the selected item.

3.10 RFID management

The RFID code is scanned by a MFRC522 RFID reader through a python script in the totem which continuously runs in the totem and sends the RFID code as soon as scanned to the server using the HTTP URL with a POST request. The code is received in the server script and assigned to a variable to be used by the other functions when needed.

3.11 NFC interface

The NFC interface is mobile phone UI exclusive. Using a library called "NFC MANAGER" in dart flutter, the access to the NFC interface of the mobile phone is granteda and upon calling the functions in login or item/user management, the RFID is read via the NFC of the phone and the code is sent to the RFID methods in the server and managed identical to the totem backend functions.

3.12 Totem registration

Totems are identified by their branches and their mac address. The mac address of the totem is sent alongside each RFID scan performed and checked by its table to confirm that the user belongs to the branch in which the user is trying to login at. getmac property of python is used for this purpose. The admin is able to manage and modify information related to its totems using the web application in the part section intended for it.

The code management of the functions is as the following:

The HTTP service scripts are managed separately among different user types except for the web application in which the file http_service.dart includes all the functions. The python scripts for the server are divided to 3 files for each type of interface in "mobile_methods.py","webApp_methods.py" and "totem_methods.py" to avoid code complexity and better git management while group mates were working on the project in parallel.

4 Front-end

Front-end has been implemented using flutter that is an open source framework by Google for building natively compiled, multi-platform applications from a single codebase.
It has been used for all the UI's (web - totem - mobile)

4.1 Web application

The web application has been developed with the aim of implementing all the services that would have been inconvenient to implement with totem. In the begging it was thought to be used only by an operator and it was a simple menu that allowed the operator to choose between options like add/remove item and add/remove user.
The web app functionalities have been expanded and now is a multi-functions tool that implement a variaty of services based on user role.

- Customer : helps the customer to navigate between items in a very web-friendly environment. The web app can also be used customer to check state of its loans/favorite items and to modify its personal data.
- Operator : the web app allows to handle with ease customers and items, the operator can search the specific one is looking through the all list or by filtering the list by setting search options.
- Admin : similarly to operator the admin can manage with ease operators like he can add/remove or by modify an existing one

4.1.1 Web welcome page and login

The welcome page of the web application can be seen below in fig 8, where an user can :

- enter as guest, he can have a look inside the website with limited functionalities indeed he can see the items available and the categories
- login by using its own unique username and password and by specifying role
- register as an admin by using the pre-acquired key

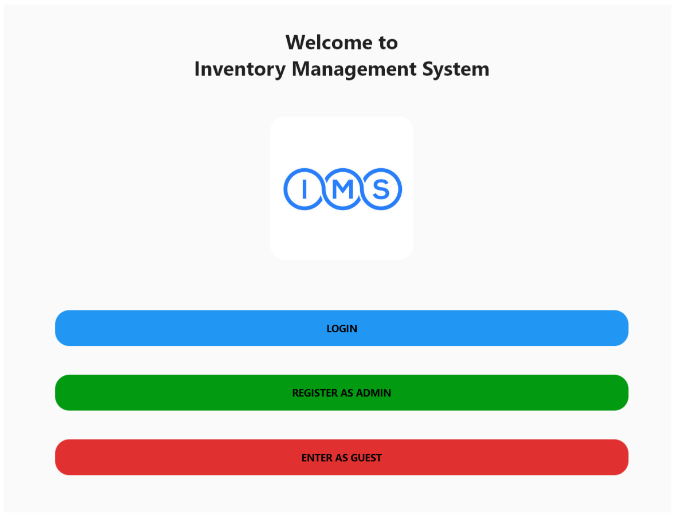
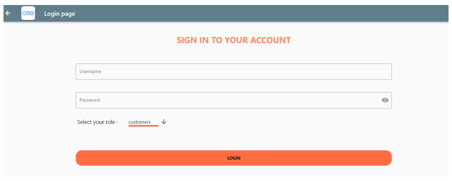
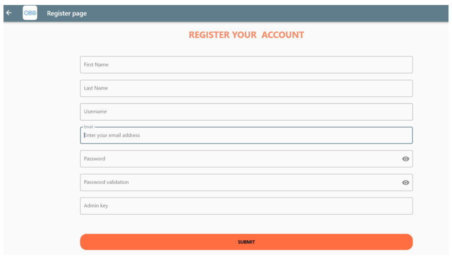


Figure 1: Welcome page [file WelcomePage.dart]



(a) Login page [file Login.dart]

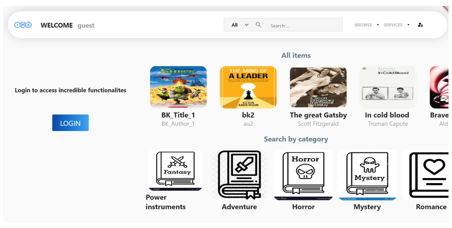


(b) Reg as admin [file Register.dart]

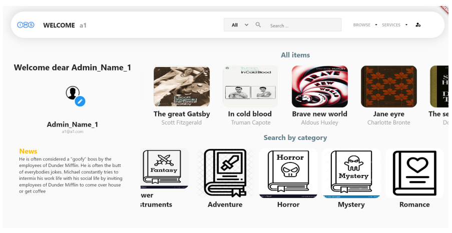
For each field in the login and register page is used a validator, it checks if the value inserted is not null and if it is then it popups an error message. When registering is also check if the username is already used and if the passwords match.

4.1.2 Dashboard page

After the welcome page the user is redirect to the **dashboard page**, here the user can found with ease all the services offered by the web app and has been thought as "safe point" where user can return at anytime while navigating in the web app.



(a) Dashboard entering as guest



(b) Dashboard after login

The dashboard has been developed to be intuitive and user-friendly, it is composed of three parts :

- Appbar : placed in the top, composed of a clickable IMS logo a searchbar and dropdown menus.
- Customer recap : are shown user personal info and it also contains a "news" section where user can found last news about its account or web app updates.
- Feed : where user can found shortcuts to browse between items, like filter them based on category

Appbar :
The appbar is placed at the top, it is kept fix across pages while user navigates since it allows to access all services.
It is composed of

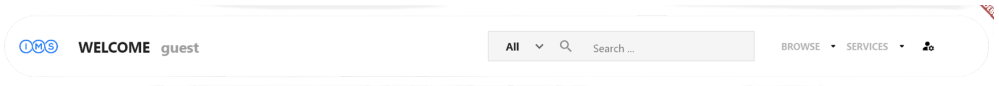


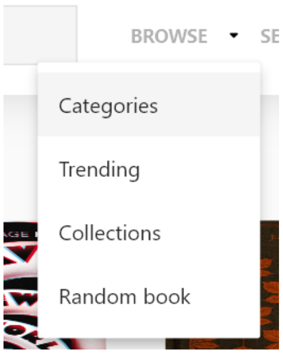
Figure 4: App bar [file App_bar.dart]

- IMS logo, when clicked re-direct the user to dashboard page
- Search bar, it allows to search a specific item or category by selecting the filter and typing in the text field
- Browse drop-down menu, it allows to choose a list of items based on
  - categories
  - trending
  - random
  - collections
- Services drop-down menu, the services shown depends on your role.
  If the user is a guest :
  - Help and support

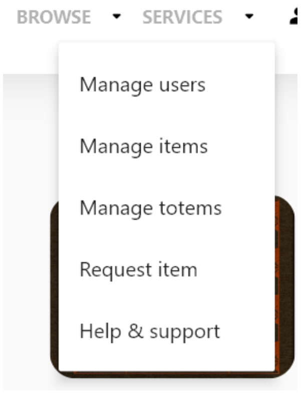
  If the user is a customer then he can also
  - Request an item

  If the user is an operator or admin then he can also
  - Manage users (if operator then only customers)
  - Mange items
- My profile drop-down menu, it shows options like my
  - my loans
  - my favorite

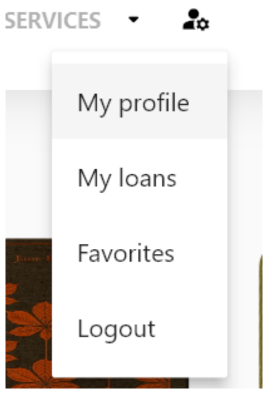
- my profile
- logout



(a) Browse menu

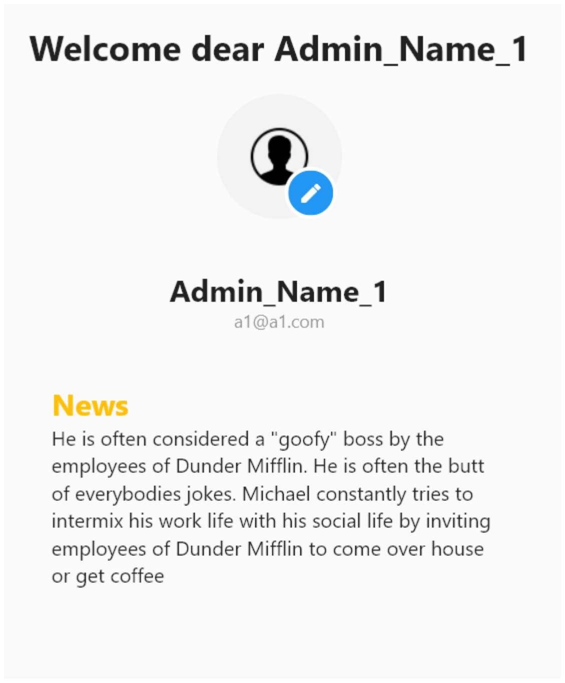


(b) Operator services menu



(c) My profile menu

Customer recap :
It contains a recap of all the customer data and a clickable icon that when clicked allows to change the profile pic of the user.
There's also a "news" section where user can found personal news or updates by IMS



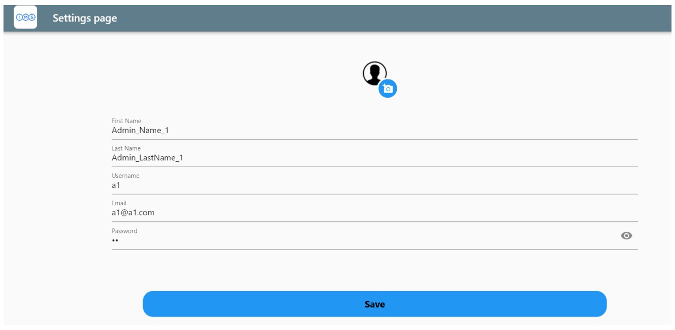
Welcome dear Admin_Name_1

Admin_Name_1
a1@a1.com

News
He is often considered a "goofy" boss by the employees of Dunder Mifflin. He is often the butt of everybodies jokes. Michael constantly tries to intermix his work life with his social life by inviting employees of Dunder Mifflin to come over house or get coffee

Figure 6: Customer recap [file UserDashboard.dart]

If the user wants to modify its data then it must go in the profile menu and select "my profile", a page like the one below will open where user can change its personal data.



First Name
Admin_Name_1
Last Name
Admin_LastName_1
Username
a1
Email
a1@a1.com
Password
••

Figure 7: User settings [file UserSettings.dart]

Feed :
In the feed section the user find a scrollable list of all the items that can expand by clicking the "all items" and also there's a scrollable list of categories that when click open a new page showing all items belonging to that category.
Selecting the item will be open a new page showing all the item details and its availability. The user then can choose if item is available to reserve it or can add it to list of its favorite items.

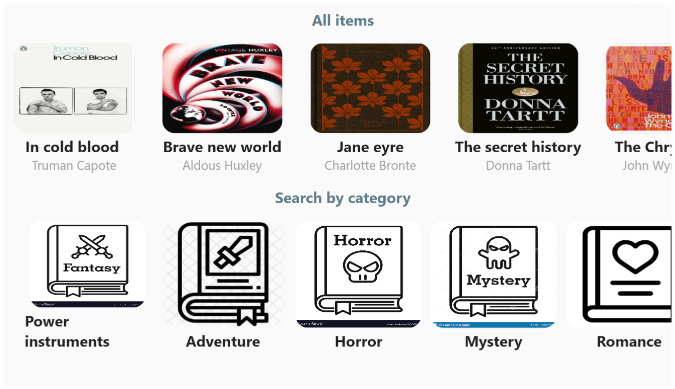
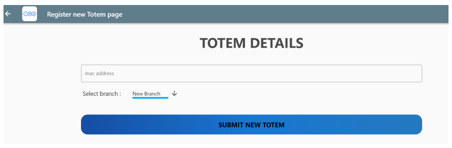
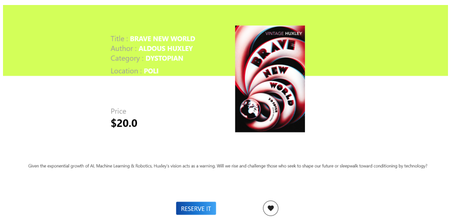


Figure 8: Browse shortcut [file FeedDashboard.dart]



mac address
Select branch : New Branch

(a) Add totem page [file addTotem.dart]



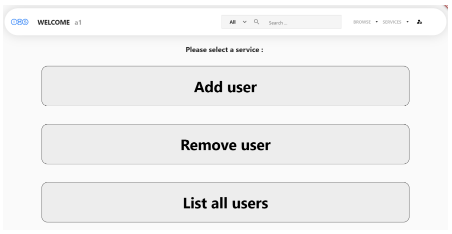
Title : BRAVE NEW WORLD
Author : ALDOUS HUXLEY
Category : DYSTOPIAN
Location : POLI
Price
$20.0

Given the exponential growth of AI, Machine Learning & Robotics, Huxley's vision acts as a warning. Will we rise and challenge those who seek to shape our future or sleepwalk toward conditioning by technology?

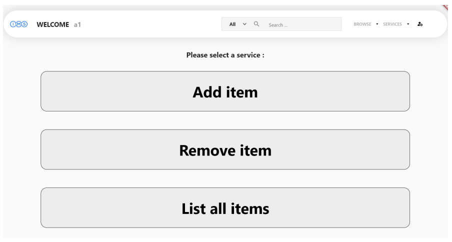
(b) Item page [file ItemPage.dart]

4.1.3 Operator and admin services :

The operator and admin can manage users and items, meaning that they can modify an item/user or add a new one or delete an existing one.
As explained in the backend section the access permission is hierarchical, meaning that an operator can only modify customers and items of its branch while an admin can modify all items and user (both customers and operators)

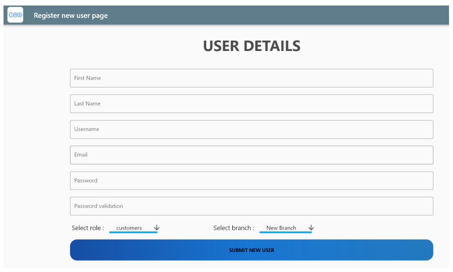


(a) Manage users [ManageUsersPage.dart]

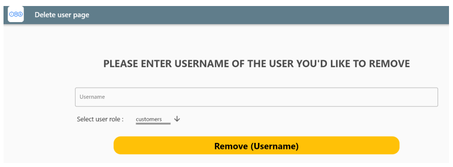


(b) Manage items [ManageItemsPage.dart]

To add a new user the operator/admin fills the forms in fig 11a , as can be seen the RFID field can be inserted manually or can be select to not fill it. If chosen the latter option the user will be add to a pending list of users that can be seen in the totem and the procedure of adding an user is completed by passing the RFID in the totem.
To remove an user it is simply typed its unique username and then complete the procedure by clicking the remove button (fig 11b)
The modify item page is pretty similar to the one of adding a new user, in this case though the fields will be already filled with user data.

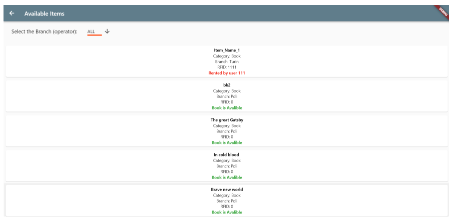


(a) Add user [addUserPage.dart]



(b) Remove user [removeUserPage.dart]

The add/remove items layout is similar to the one of user, in this case tough the item is searched by its unique identifier.
To modify an item it is first searched (fig 12a) then it is chosen by the result list and a new page will appear (fig 12b). Here can be modified item's data and lastly the data are updated by clicking the save button.



Select the Branch (operator): ALL

Item_Name_1
Category: Book
Branch: Turin
RFID: 1111
Rented by user 111

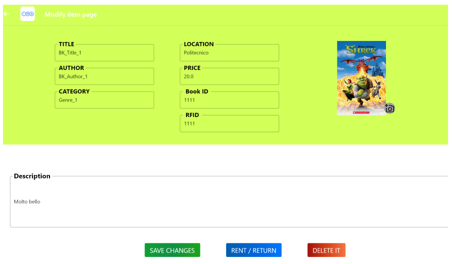
bk2
Category: Book
Branch: Poli
RFID: 0
Book is Avalible

The great Gatsby
Category: Book
Branch: Poli
RFID: 0
Book is Avalible

In cold blood
Category: Book
Branch: Poli
RFID: 0
Book is Avalible

Brave new world
Category: Book
Branch: Poli
RFID: 0
Book is Avalible

(a) Find item [listItemPage.dart]



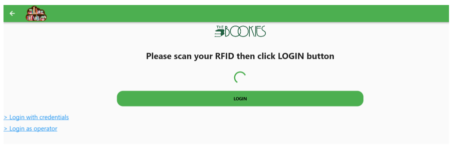
TITLE: BK_Title_1
AUTHOR: BK_Author_1
CATEGORY: Genre_1
LOCATION: Politecnico
PRICE: 20.0
Book ID: 1111
RFID: 1111
Description: Molto bello

(b) Modify item [modifyItemPage.dart]

4.2 Totem interface

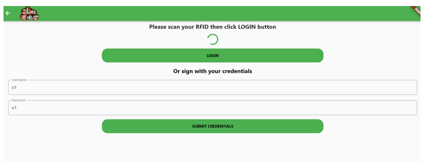
Totem UI has been developed to implement totem's functions to be as simplest as possible since the totem is used by customers and operators in the service point and so should be fast and easy. Login to the totem (fig 13a) can happen with :

- RFID, indeed the totem waits to scan an rfid
- with credentials, using the unique username and pwd
- login as operator, similarly to login as operator can be used credentials (fig 13b)



Please scan your RFID then click LOGIN button

(a) Totem login page [file TLogin.dart]



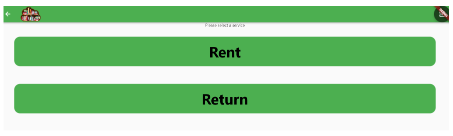
Please scan your RFID then click LOGIN button

Or sign with your credentials

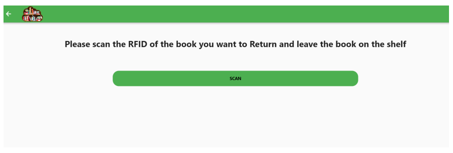
Username: o1
Password: o1

(b) Login as operator [file TLoginOperator.dart]

The customer then can choose between renting an item or return one (fig 14a), to rent/return it then it can be done by simply scanning its RFID.



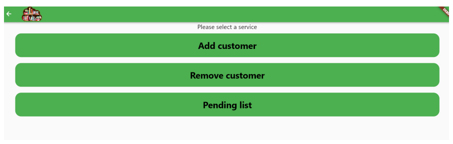
(a) Customer services



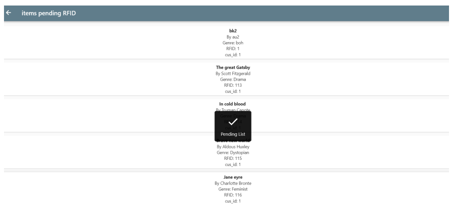
Please scan the RFID of the book you want to Return and leave the book on the shelf

(b) Return an item

The operator with a similar layout can add/remove an item by scanning its RFID or can complete the pending requests by scanning the RFID too.



(a) Operator management of customers


items pending RFID

bk2
By au2
Genre: boh
RFID: 1
cus_id: 1

The great Gatsby
By Scott Fitzgerald
Genre: Drama
RFID: 113
cus_id: 1

In cold blood
By Truman Capote
Genre: Drama
RFID: 114
cus_id: 1

Brave new world
By Aldous Huxley
Genre: Dystopian
RFID: 115
cus_id: 1

Jane eyre
By Charlotte Bronte
Genre: Feminist
RFID: 116
cus_id: 1

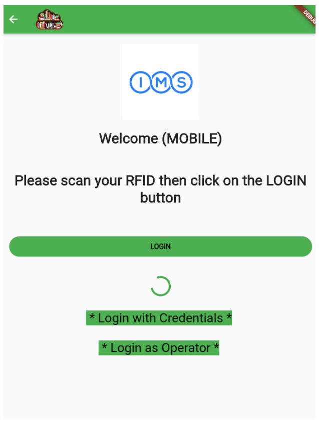
(b) Pending item list

4.3 Mobile interface

Mobile application interface is privileged with the NFC property for the use of customers and operators. This interface generally implements the totem's functionalities with or without NFC. In other words, the approach for this interface was to emulate a portable totem for the operators and an easier access to the rented items' list for the customers.
The properties of the mobile UI are the following:

- Login using credentials and RFID card for customers and operators (fig:16a)
- list of the branch items for the customers and operators of the branch (fig:16d)
- list of the branch customers for the operators of the branch(fig:16e)
- Modification of the items and users of a branch by the operator(fig:16b)
- Settings for the user data modification (fig:16f)
- list of rented items for the customers of the branch (fig:16c)
- Adding users and items with or without RFID for the operators(fig:16g and 16h)
- Removing users and items with or without RFID for the operators (fig:16i)

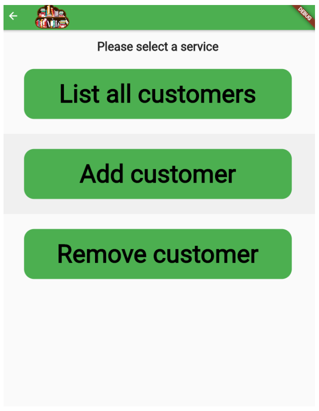
The reason for having an option to add or remove users or items also without RFID code is for the case of the mobile phones in which the NFC interface is not supported. The items and users added without the RFID code, are sent to the pending list of the operator in the Totem interface for the RFID code to be added later.The RFID of the user or item is also manually addable using the modification section of the web interface .



Welcome (MOBILE)

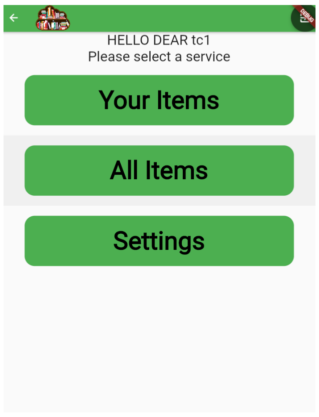
Please scan your RFID then click on the LOGIN button

(a) file MWelcomePage.dart



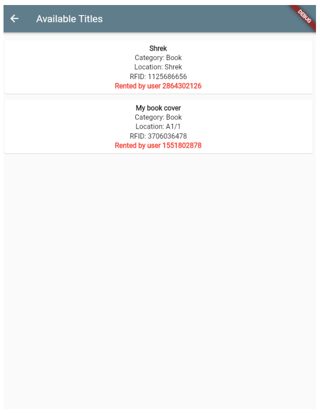
Please select a service

(b) file MModifyCustomers.dart



HELLO DEAR tc1
Please select a service

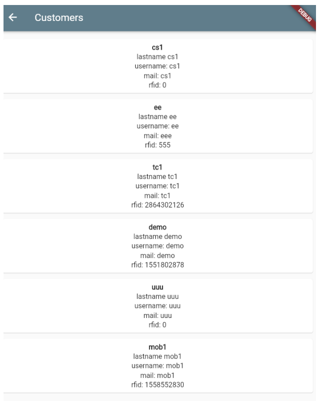
(c) file MHomePage_us.dart



Shrek
Category: Book
Location: Shrek
RFID: 1125686656
Rented by user 2864302126

My book cover
Category: Book
Location: A1/1
RFID: 3706036478
Rented by user 1551802878

(d) file MListItems.dart



cs1
lastname cs1
username: cs1
mail: cs1
rfid: 0

ee
lastname ee
username: ee
mail: eee
rfid: 555

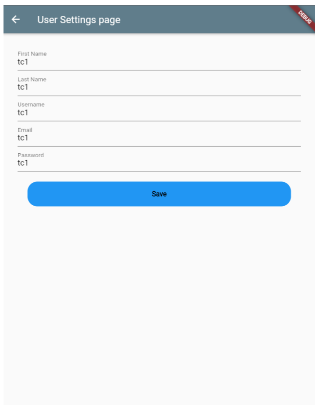
tc1
lastname tc1
username: tc1
mail: tc1
rfid: 2864302126

demo
lastname demo
username: demo
mail: demo
rfid: 1551802878

uuu
lastname uuu
username: uuu
mail: uuu
rfid: 0

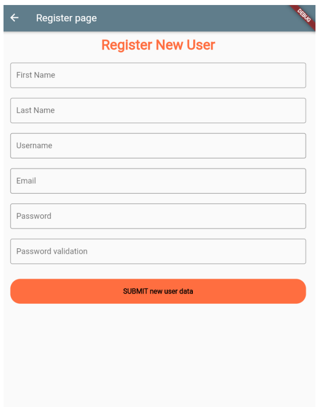
mob1
lastname mob1
username: mob1
mail: mob1
rfid: 1558552830

(e) file MListCustomers.dart



First Name
tc1
Last Name
tc1
Username
tc1
Email
tc1
Password
tc1

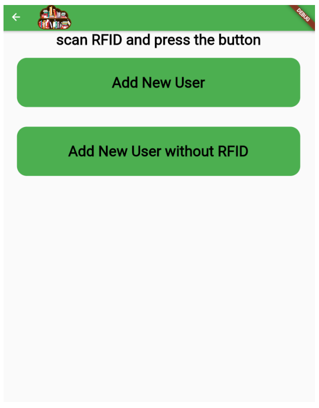
(f) file UserSettings.dart



Register New User

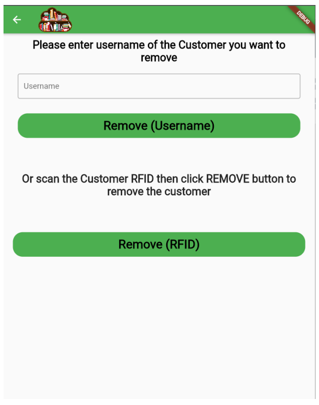
First Name
Last Name
Username
Email
Password
Password validation

(g) file MAddCustomer.dart



scan RFID and press the button

(h) file MAddCustomerRFID.dart



Please enter username of the Customer you want to remove

Username

Or scan the Customer RFID then click REMOVE button to remove the customer

(i) file MRemoveCustomer.dart