



CSC490

Software engineering

Spring 2020

Instructor: Karim Youssef

George Younes
Youssef Abboud
Christina Aouad

Submission 3

A. Detailed Design:

I-Stock Management System:

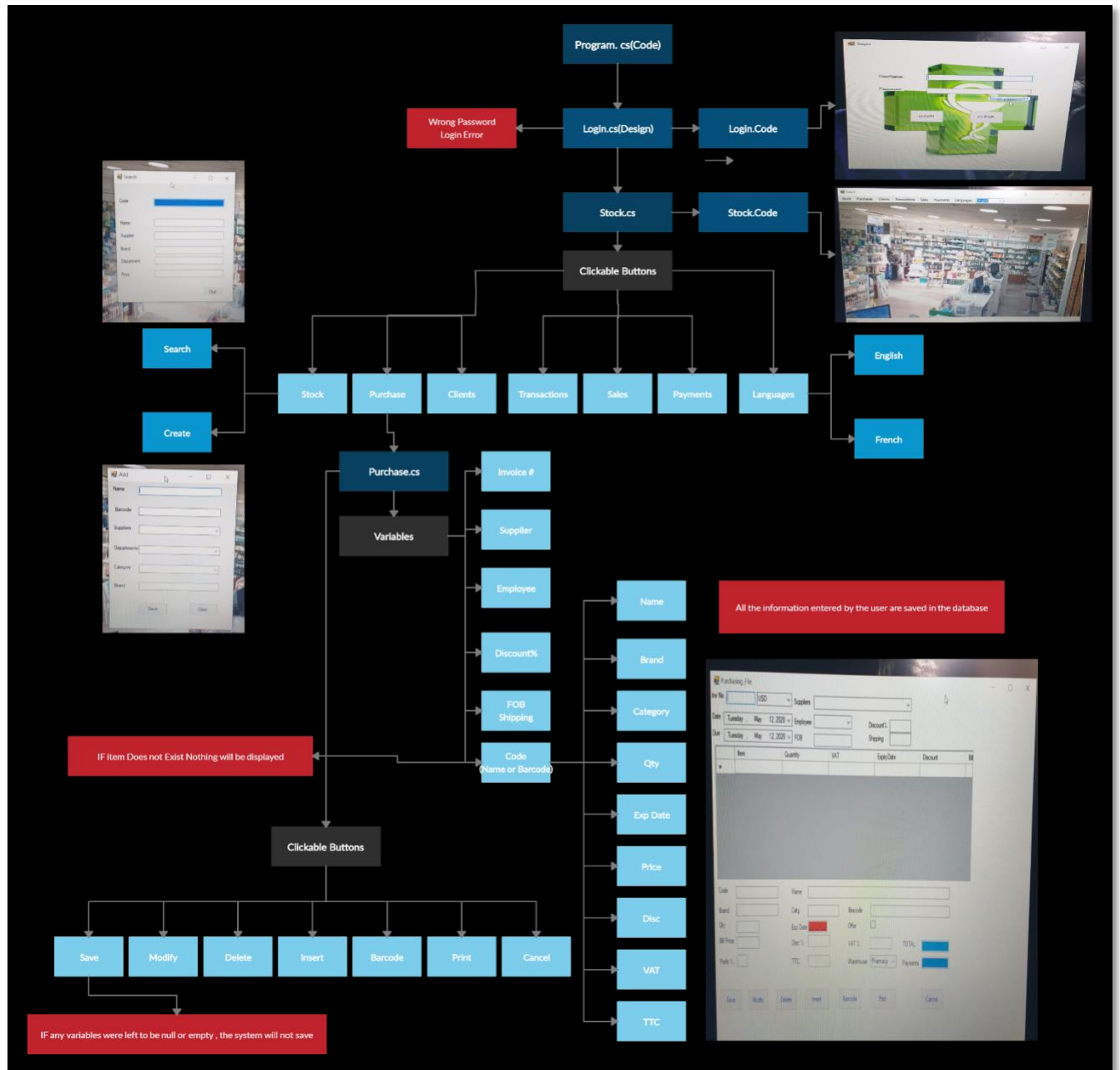
In this system, I wanted to use JDBC as a language and NetBeans as coding platform, but after a meeting with the clients and a lot of researches, I found out that this application will be installed on several windows computers in the pharmacy, this why I found out that using C# as a language would be the most efficient way to develop this application. C# application can be easily converted to .NET application and can be installed on all windows platform and be connected all together through a main server in the pharmacy.

Concerning the database, I originally intended to use Xampp since we are familiar with it and it provide a large amount of data using MySQL Workbench, but the clients wanted Excel as a database since the old system relied heavily on Excel and the manager did not want to introduce a new aspect of database to the staff.

In this UML diagram, I will present how this program is divided to different classes:

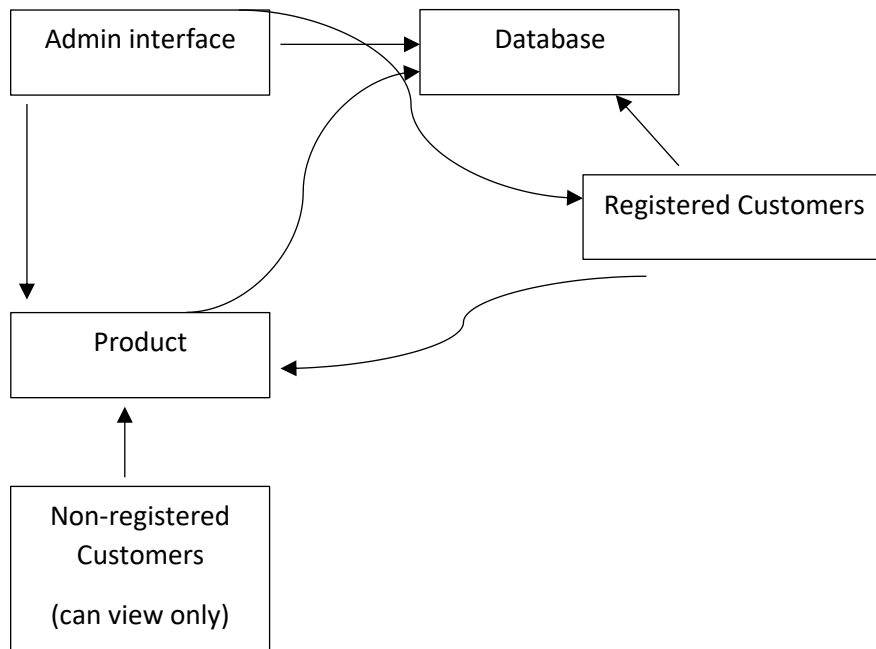
.cs (Design) indicate the interface and the form the user sees. (Next to each .cs (Design) I added a figure of the program in this particular stage)

.cs (Code) indicate the actual code and the function alongside with the variables for each .cs (Design) class

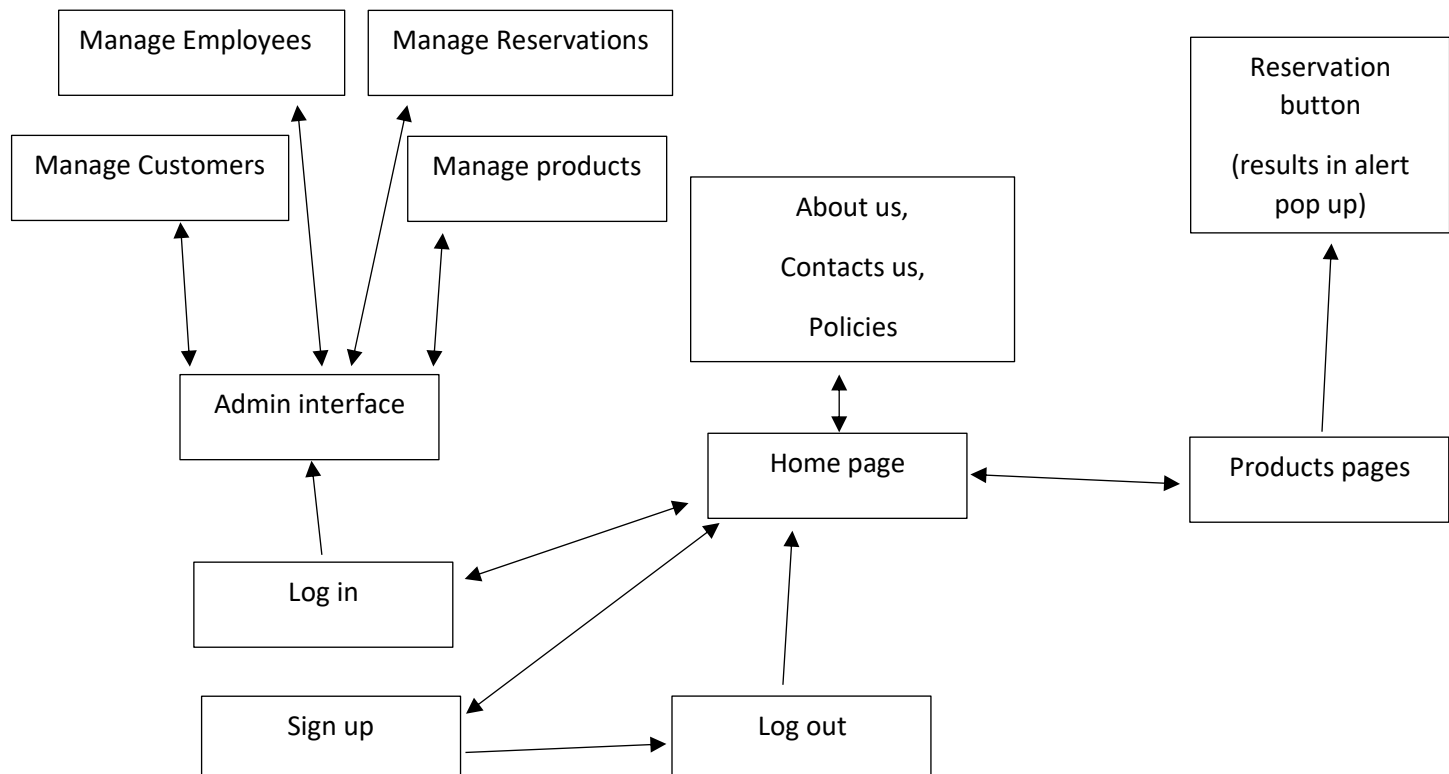


II-Pharmacy's website:

a) Sub-systems module:



b) Modules module:



c) Sub-systems description:

<u>Sub-system</u>	<u>Description</u>
Admin interface	-Has access to all customers, employees, products, reservations -Can add, delete, and update from the above
Product	-Products can be seen under in a page with similar products -Products can be seen individually and from there can be reserved (we do not sell since that is against the Lebanese law)
Registered customers	-Are customers with accounts and can log in -Can view and reserve products
Non-registered customers	-Are customers that do not have accounts or simply have not logged in -Can only view the products
Database	-Everything is stored in the database -All updates, deletions, additions and so on are directly made in the database via ajax

B. Method/Function Specifications:

I-Stock Management System:

Functions:	Login to the system.
Description:	Write username and password and press login to access the software.
Inputs:	String: Username String: Password.
Source:	User Interface
Outputs:	Wrong Password (If username or password are not found in the database) Access to the system if right username and password.
Destination:	Database.
Action:	If Wrong password or username: Access denied If right password and username: Access Granted.
Requires:	Input: Username and password.
Pre-condition:	Application Connected to database.
Post-condition:	-

Functions:	Create an Item
Description:	Add a new item to the pharmacy database.
Inputs:	Name, Barcode, Suppliers, Departments, Category and Brand.
Source:	User Interface
Outputs:	Message: -Saved -Please Insert all Values
Destination:	Database.
Action:	If all inputs are inserted: - “New Item created”. If User press SAVE button without inserting all inputs:” Please Insert All Values”.
Requires:	All inputs.
Pre-condition:	-Database Connection.
Post-condition:	-Database Connection.

Functions:	Search an Item.
Description:	Search for an item.
Inputs:	Full Item’s Name or Barcode.
Source:	User Interface.
Outputs:	Name, Supplier, Brand, Department, Price.
Destination:	Database.
Action:	If item exist in the database, output will be displayed. If item does not exist, nothing will be displayed.
Requires:	Name or Barcode.
Pre-condition:	-Database Connection.
Post-condition:	-Database Connection.

Functions:	Save Invoice
Description:	Save all information of a particular invoice in the database.
Inputs:	Invoice #, USD or LBP, Suppliers, Date, Employee, Code, Qty, Price, Discount, VAT, TTC.
Source:	User Interface.
Outputs:	-SAVED -Please Enter all information.
Destination:	Database.
Action:	If all information is inserted, the save button will save the invoice information, Else it will prompt the user to enter all required information.
Requires:	All inputs.
Pre-condition:	-Database Connection.
Post-condition:	-Database Connection.

Functions:	Barcode Printing.
Description:	Printing Barcode for item without Barcode.
Inputs:	-
Source:	User Interface.
Outputs:	Paper with all required Barcodes.
Destination:	Printer.
Action:	Printing Barcodes
Requires:	Printer connection.
Pre-condition:	Invoice must be saved.
Post-condition:	-

Functions:	Print.
Description:	Printing the invoice with all required information.
Inputs:	-
Source:	User Interface.
Outputs:	Invoice printed with all previous inputs chosen by user.
Destination:	Printer.
Action:	Printing the invoice.
Requires:	Printer Connection.
Pre-condition:	Invoice must be saved.
Post-condition:	-

Functions:	Creating a New Supplier.
Description:	Creating a non-existing Supplier.
Inputs:	Supplier Name, Phone Number, Email Address, Address and Info(optional).
Source:	User Interface.
Outputs:	"Created". "Please Enter all fields".
Destination:	Database.
Action:	If all inputs are inserted: "Created". Else "Please Enter all fields".
Requires:	All inputs.
Pre-condition:	-Database Connection.
Post-condition:	-Database Connection.

II- Pharmacy's website:

Functions:	Login
Description:	Enter email and password to login to account
Inputs:	String: Email String: Password.
Source:	Website Interface
Outputs:	-Welcome Admin -Wrong password -Wrong Email
Destination:	Database.
Action:	If Wrong password or username: Wrong Password, Wrong Email. If right password and username: Welcome Admin.
Requires:	Input: Email and password.
Pre-condition:	Internet Connection
Post-condition:	Internet Connection

Functions:	Register
Description:	Register in a new Account
Inputs:	String: Email String: Password String: First name String: Password.
Source:	Website Interface
Outputs:	-Already exist -Field Missing, Try Again. -Registered Successfully.
Destination:	Database.
Action:	If Email already exist: Already exist If filed missing: Field Missing, Try Again. If user entered non existing email and password, first name and last name: Registered Successfully.
Requires:	Input: Email, Password, First name, Last name.
Pre-condition:	Internet Connection
Post-condition:	Internet Connection

Functions:	Logout.
Description:	Logout from current page
Inputs:	-
Source:	Website Interface
Outputs:	-
Destination:	Previous page.
Action:	Logout form THIS page
Requires:	-
Pre-condition:	Internet Connection
Post-condition:	Internet Connection

Functions:	Reservation
Description:	User is able to reserve an online item to buy it later
Inputs:	Click On reservation button
Source:	Website Interface
Outputs:	-Item reserved. -Not available. -Server Error.
Destination:	Database.
Action:	If item is available: Item reserved. If item is not available to be purchased: Not available. If internet Error: Server Error.
Requires:	Input: Email, Password, First name, Last name.
Pre-condition:	Internet Connection
Post-condition:	Internet Connection

Functions:	Reservation
Description:	User is able to reserve an online item to buy it later
Inputs:	Click On reservation button
Source:	Website Interface
Outputs:	-Item reserved. -Not available. -Server Error.
Destination:	Database.
Action:	If item is available: Item reserved. If item is not available to be purchased: Not available. If internet Error: Server Error.
Requires:	Input: Email, Password, First name, Last name.
Pre-condition:	Internet Connection
Post-condition:	Internet Connection

Functions:	AddBtt
Description:	Add a new employee to pharmacy website database.
Inputs:	Email, Password, FirstName, Last Name, Number.
Source:	Website Interface
Outputs:	-Error. -Employee Already Exist. -Added Successfully.
Destination:	Database.
Action:	Creating all new information for a new employee, if new employee and all input are inserted: Added Successfully. If Employee already exist: Employee Already Exist. If internet Connection: Error.
Requires:	Input: Email, Password, FirstName, Last Name, Number.
Pre-condition:	Internet Connection
Post-condition:	Internet Connection

Functions:	RemoveNbr ()
Description:	Remove an employee based on a phone number
Inputs:	Phone number (int)
Source:	Admin interface, manage employees
Outputs:	"Error in fields" if there is an error in the input "Removed Successfully!" if all worked out and the employee was removed "No employee" if there is no employee that exists with this number
Destination:	Database.
Action:	If wrong: No action If removed: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Functions:	Removeemail ()
Description:	Remove an employee based on an email
Inputs:	Email(string)
Source:	Admin interface, manage employees
Outputs:	“Error in fields” if there is an error in the input “Removed Successfully!” if all worked out and the employee was removed “No employee” if there is no employee that exists with this email
Destination:	Database.
Action:	If wrong: No action If removed: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Functions:	Addbtt ()
Description:	Add a customer
Inputs:	Email (string), password(string), first name (string), last name (string)
Source:	Admin interface, manage customers
Outputs:	“customer already exists” if there is a customer with the same email “Added Successfully!” if all worked out and the customer was added “Error in fields” if there is any error in the input
Destination:	Database.
Action:	If wrong: No action If added: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Functions:	removecemail ()
Description:	Removes customer(s) based on email
Inputs:	email (string)
Source:	Admin interface, manage customers
Outputs:	“Error in fields” if there is an error in the input “Removed Successfully!” if all worked out and the customer was removed “No customer found” if there is no customer that exists with this email
Destination:	Database.
Action:	If wrong: No action If removed: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Functions:	Additem ()
Description:	Add a product
Inputs:	name (string), brand(string), description (string), quantity (int), price (float)
Source:	Admin interface, manage products
Outputs:	“customer already exists” if there is an item that exists with these parameters “Added Successfully!” if all worked out and the item was added “Error in fields” if there is any error in the input
Destination:	Database.
Action:	If wrong: No action If added: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Functions:	removeitem ()
Description:	Removes an item
Inputs:	name (string), brand(string), description (string), price (float)
Source:	Admin interface, manage products
Outputs:	“Removed Successfully!” if all worked out and the item was added “Error in fields” if there is any error in the input
Destination:	Database.
Action:	If wrong: No action If removed: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Functions:	updtbtt ()
Description:	Updates an employee’s email and phone number
Inputs:	first name (string), last name (string), new Email (string), new phone nb(int)
Source:	Admin interface, manage products
Outputs:	“Employee updated” if all worked out and the item was added “couldn’t update” if there is any error in the input
Destination:	Database.
Action:	If wrong: No action If updated: Updates the database via ajax
Requires:	Being on the admin interface
Pre-condition:	Logged in as admin
Post-condition:	-

Website Pages Definition:

In this part we will specify the name of every page or group of pages included in our website and define what it is and what it does briefly.

Name of the page	Definition
Privacy	This is a HTML page that state the privacy policy concerning everything related to the clients and the pharmacy (data, personal information...)
Shop	This is a PHP page that imports all the information concerning the pharmacy products (data about products+ images) from the database
About	This is a PHP page that gives the user some information and background about RAHBANI pharmacy.
Connection	This is PHP file that make sure the website is connected to the SQL database
Contact	This is a PHP page that contains all information if a user wants to communicate with the pharmacy (email, phone number, Major Staff number and email...)
CSS	This is CSS page it includes the designs for the whole website.
Fonts	These are different fonts used throughout the website to make it look better
Img	These are images used in the website
JS	JavaScript file with all the functions (all explained individually above)
Verification	Includes many php pages. These pages include Explanation below.
a_item.php	Loads all products from the database to the admin page
addcustomer.php	From the admin interface. Adds a customer. Function from JS file explained above
addemployee.php	From the admin interface. Adds an employee. Function from JS file explained above
Logout.php	Logs out. Function from JS file explained above
R_item.php	From the admin interface. Removes a product. Functions from JS file explained above
Remove.php	From the admin interface. Removes an employee. 2 functions from JS file explained above
Removecustomer.php	From the admin interface. Removes a customer. Function from JS file explained above

Reservationbp.php Reservationbt.php Reservationc.php Reservationd.php Reservationp.php Reservations.php Reservationsn.php	Reserves an item. Each file corresponds to the reservation of an item under a certain category (baby products, baby toiletries, cosmetics, dental, para-medicals, supplements, sport-nutrition)
Signup.php	Signs up. Function from JS explained above
Update.php	From the admin interface. Updates an employee. Function from JS file explained above
Verification.php	PHP file that verifies if it is Sonia, the founder and CEO of the pharmacy, then it logs her in to the admin interface. If it is someone else, it verifies if the email and password are in the database or not and logs in the person or not. (login ())
404page.html	404 page.
Admin.php	The first page the admin sees after logging in
Index.php	Pages you see before logging in
Index2.php	Pages you see when you are logged in
Login.php	Log in page. Function from JS explained above.
Pharmacy.sql	Database used for the website