





# Faculty of Engineering

## Summer Training Report

CCE Department – Third year – English and French Sections

### Agriculture Store System

made by

**Ahmad Salemeh, 65036**

**Ali Kassas, 65650**

Supervised By

**Ali Mokdad**

Presented To

**Dr. Walid Fahs**

October 2022

## TABLE OF CONTENTS

### Contents

<b>LIST OF FIGUERS:</b> .....	<b>4</b>
<b>ABSTRACT .....</b>	<b>5</b>
<b>REQUIREMENTS AND DELIVERABLES .....</b>	<b>6</b>
<b>GENERAL INTRODUCTION.....</b>	<b>7</b>
<b>LITERATURE REVIEW .....</b>	<b>8</b>
<b>DATABASE IMPLEMENTATION.....</b>	<b>9</b>
<b>MOST SOFTWARE IMPLEMENTATIONS.....</b>	<b>15</b>
<b>CUSTOMER ACTIVITY DIAGRAM: .....</b>	<b>28</b>
<b>ADMIN ACTIVITY DIAGRAM:.....</b>	<b>29</b>
<b>CUSTOMER SEQUENCE DIAGRAM .....</b>	<b>30</b>
<b>ADMIN SEQUENCE DIAGRAM .....</b>	<b>32</b>
<b>CONCEPTUAL DIAGRAM .....</b>	<b>33</b>
<b>CUSTOMER USE-CASE DIAGRAM .....</b>	<b>34</b>
<b>ADMIN USE-CASE DIAGRAM.....</b>	<b>35</b>
<b>RELATIONAL DATABASE DIAGRAM.....</b>	<b>36</b>
<b>CONCLUSION AND PERSPECTIVE .....</b>	<b>37</b>
<b>BIBLIOGRAPHY .....</b>	<b>38</b>

---

## LIST OF FIGURES:

**FIGURE 1: HOME PAGE**

**FIGURE 2: SIGN UP PAGE**

**FIGURE 3: SIGN IN PAGE**

**FIGURE 4: SHOPS PAGE**

**FIGURE 5: TREES PAGE**

**FIGURE 6: PLANTS PAGE**

**FIGURE 7: TOOLS/SEEDS/MEDICINES PAGE**

**FIGURE 8: ITEM PAGE**

**FIGURE 9: CART PAGE**

**FIGURE 10: PURCHASE PAGE**

**FIGURE 11: THANKS PAGE**

**FIGURE 12: ABOUT US PAGE**

**FIGURE 13: MEMBERSHIP IN HOME PAGE**

**FIGURE 14: MEMBERSHIP SUBSCRIPTION**

**FIGURE 15: MEMBERSHIP PAYMENT**

**FIGURE 16: MEMBER LOGIN PAGE**

**FIGURE 17: HELP CENTER PAGE**

**FIGURE 18: DISEASES PAGE**

**FIGURE 19: LANDS PAGE**

**FIGURE 20: ADMIN HOME PAGE**

**FIGURE 21: ALTER PAGE**

**FIGURE 22: VIEW PAGE**

**FIGURE 23: SEARCH FOR PAGE**

**FIGURE 24: CUSTOMER ACTIVITY DIAGRAM**

**FIGURE 25: ADMIN ACTIVITY DIAGRAM**

**FIGURE 26: CUSTOMER SEQUENCE DIAGRAM (1)**

**FIGURE 27: CUSTOMER SEQUENCE DIAGRAM (2)**

**FIGURE 28: ADMIN SEQUENCE DIAGRAM**

**FIGURE 29: CONCEPTUAL DIAGRAM**

**FIGURE 30: CUSTOMER USE-CASE DIAGRAM**

**FIGURE 31: RELATIONAL DATABASE**

## ABSTRACT

Agriculture Store System is an application based on the idea of online agriculture store that fills the needs of your backyard garden, in this application the customer can create his own account to use it everytime he wants to access our store and buy a variaty of different interesting stuff and items including a various collection of trees, seeds, plant medecines, farmer and planting tools, ect..., furthermore the customer can have access to a very special collection of items and other great features by subscribing to our membership moreover customers can check our Help center to find out some ideas about different types of lands and what are the plants that are suitable to plant in this land as well as tips on how you can improve it's productivity. On the other hand, customers can also check for most common tree diseases discriptions and ways to treat each disease.

As for the admin side of this project, they have full access to the system after the permission given by the main admin (store owner), and some of the things that admins can do are: adding or deleting any items existing in the store/ adding new items to the storehouse/ adding and deleting admins and workers/ view total invoices purchased by customers as well as viewing the items remaining in the storehouse/ search for admins/ members/ items/ ect....

## REQUIREMENTS AND DELIVERABLES

### 1) This java application has two users:

- **Admin (the shop owner):** the admin logs in to the application and have full access to the system where he can:

- Add and delete items.
- View total invoices.
- Search for item, member, worker and view info about each.
- View items remaining in the Storehouse.
- Manage the items: update (change) their prices and quantity.

- **Customer (End user):** the customer logs in to the application and have access to several things like:

- Adding trees, seeds, planting tools, medicines and pesticides, etc... To the cart.
- View, change, and delete items from cart.
- Go to purchase and pay for items existing in the cart.
- Try our prediction system that allows customer to check for land characteristics and ways to improve it as well as check for tree diseases and ways of treatment.
- Have access to extra interesting, useful features through subscribing to the membership.

## General Introduction

Greenland is an application based on the idea of online agriculture shop. Basically this project allows end users (customers) to access various multiple shops that concerns everything about agriculture such as buying trees, seeds, planting tools, medicines and pesticides, etc. , it also shows information about each item in any shop. Moreover this application gives access to various interesting features for customers who subscribes to membership, also it allows customers to view some of several lands characteristics and ways to improve it, as well as some of the most common diseases that plants are infected by. As for the admin side of this application, they are applicable to do several actions concerning the store and how to modify it as well as other features.

The aim of this project is to allow people to buy products related to plants and planting, as well as to view some informations about important stuff related to plants such as the lands and plant diseases.

**This application was implemented using **java** language for the design and functionality of the app.**

## Literature Review

A similar project can be found on the internet through the following url: <https://www.trees.com/>

Source:

 trees.com was first indexed by Google more than 10 years ago, if you visit this site you can notice a different type of trees for the customer to check and buy, which gives our application a step forward since **1)** not only we sell different types of trees as this site does, but also tools that ease the planting process, as well as different type of plants like seeds, **2)** moreover you can find in our application some medicines for these plants in case they were infected in any disease, **3)** you can check your land characteristics and types of trees you can plant in as well as how to improve the productivity of this land, **4)** you can check some common tree diseases descriptions and ways to treat these diseases, **5)** what also makes our application better is the pricing , where we chose carefully the best acceptable price for each product, **6)** the ability to select different items at the same time and then purchasing them all at once, in addition that you can view and alter on your cart, etc.

## DataBase Implementation

This project database was created and built using **MYSQL**.

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
-- Host: 127.0.0.1:3306
-- Generation Time: Oct 31, 2022 at 08:02 PM
-- Server version: 5.7.36
-- PHP Version: 7.4.26

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

-- Database: `summerproject`


-- Table structure for table `accounts`
DROP TABLE IF EXISTS `accounts`;
CREATE TABLE IF NOT EXISTS `accounts` (
    `userID` int(11) NOT NULL AUTO_INCREMENT,
    `username` varchar(50) NOT NULL,
    `password` varchar(50) NOT NULL,
    `securityQuestion` varchar(50) NOT NULL,
    PRIMARY KEY (`userID`),
    UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `invoice`
DROP TABLE IF EXISTS `invoice`;
CREATE TABLE IF NOT EXISTS `invoice` (
    `userID` int(11) NOT NULL,
    `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
`fname` varchar(50) NOT NULL,  
`lname` varchar(50) NOT NULL,  
`userPh` int(50) NOT NULL,  
`purchase_date` datetime NOT NULL,  
`total_price` double NOT NULL,  
PRIMARY KEY (`id`),  
KEY `invoice_ibfk_2`(`userPh`),  
KEY `userID`(`userID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

**-- Table structure for table `invoicedetails`**

```
DROP TABLE IF EXISTS `invoicedetails`;  
CREATE TABLE IF NOT EXISTS `invoicedetails` (  
`id` int(10) NOT NULL AUTO_INCREMENT,  
`Item id` int(10) NOT NULL,  
`ItemType` varchar(50) NOT NULL,  
`ItemName` varchar(50) NOT NULL,  
`Quantity` int(10) NOT NULL,  
`Price` double NOT NULL,  
`Invoice_id` int(11) NOT NULL,  
PRIMARY KEY (`id`),  
KEY `Invoice_id`(`Invoice_id`),  
KEY `Item id`(`Item id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

**-- Table structure for table `items`**

```
DROP TABLE IF EXISTS `items`;  
CREATE TABLE IF NOT EXISTS `items` (  
`id` int(10) NOT NULL AUTO_INCREMENT,  
`ItemType` varchar(50) NOT NULL,  
`Price` double NOT NULL,  
`quantity` int(10) NOT NULL,
```

```
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `medecine`

DROP TABLE IF EXISTS `medecine`;

CREATE TABLE IF NOT EXISTS `medecine` (
  `id` int(10) NOT NULL,
  `medecine` varchar(50) NOT NULL,
  `used for` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

```
-- Table structure for table `membership`

DROP TABLE IF EXISTS `membership`;

CREATE TABLE IF NOT EXISTS `membership` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `type` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

```
-- Table structure for table `seeds`

DROP TABLE IF EXISTS `seeds`;

CREATE TABLE IF NOT EXISTS `seeds` (
  `id` int(10) NOT NULL,
  `seedName` varchar(50) NOT NULL,
  `plantingTime` varchar(50) NOT NULL,
  `harvestTime` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

```
-- Table structure for table `tools`

DROP TABLE IF EXISTS `tools`;

CREATE TABLE IF NOT EXISTS `tools` (
```

```
 `id` int(10) NOT NULL,  
 `ToolName` varchar(50) NOT NULL,  
 PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

**-- Table structure for table `trees`**

```
DROP TABLE IF EXISTS `trees`;  
CREATE TABLE IF NOT EXISTS `trees` (  
 `id` int(10) NOT NULL,  
 `TreeType` int(10) NOT NULL,  
 `TreeName` varchar(50) NOT NULL,  
 `TreeSize` double NOT NULL,  
 `waterAmountPerWeek` int(10) NOT NULL,  
 `weatherType` varchar(50) NOT NULL,  
 `HarvestTime` varchar(50) NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY `TreeType` (`TreeType`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

**-- Table structure for table `tree\_type`**

```
DROP TABLE IF EXISTS `tree_type`;  
CREATE TABLE IF NOT EXISTS `tree_type` (  
 `id` int(11) NOT NULL AUTO_INCREMENT,  
 `TreeType` varchar(50) NOT NULL,  
 PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

**-- Table structure for table `users`**

```
DROP TABLE IF EXISTS `users`;  
CREATE TABLE IF NOT EXISTS `users` (  
 `id` int(10) NOT NULL AUTO_INCREMENT,  
 `firstname` varchar(50) NOT NULL,
```

```
`lastname` varchar(50) NOT NULL,  
 `phoneNumber` varchar(50) NOT NULL,  
 `address` varchar(50) NOT NULL,  
 `role` varchar(50) NOT NULL,  
 `date` datetime NOT NULL,  
 `membershipType` int(10) NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY `membershipType` (`membershipType`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

---

#### -- Table structure for table `workers`

```
DROP TABLE IF EXISTS `workers`;  
CREATE TABLE IF NOT EXISTS `workers` (  
 `id` int(10) NOT NULL AUTO_INCREMENT,  
 `first name` varchar(50) NOT NULL,  
 `last name` varchar(50) NOT NULL,  
 `position` varchar(50) NOT NULL,  
 `salary` double NOT NULL,  
 `phoneNumber` int(10) NOT NULL,  
 PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Constraints for dumped tables

#### -- Constraints for table `accounts`

```
ALTER TABLE `accounts`  
 ADD CONSTRAINT `accounts_ibfk_1` FOREIGN KEY (`userID`) REFERENCES `users` (`id`) ON DELETE  
 CASCADE ON UPDATE CASCADE;
```

#### -- Constraints for table `invoice`

```
ALTER TABLE `invoice`  
 ADD CONSTRAINT `invoice_ibfk_1` FOREIGN KEY (`userID`) REFERENCES `users` (`id`) ON DELETE  
 CASCADE ON UPDATE CASCADE;
```

#### -- Constraints for table `invoicedetails`

```
ALTER TABLE `invoicedetails`  
ADD CONSTRAINT `invoicedetails_ibfk_2` FOREIGN KEY (`Invoice_id`) REFERENCES `invoice`(`id`)  
ON DELETE CASCADE ON UPDATE CASCADE,  
ADD CONSTRAINT `invoicedetails_ibfk_3` FOREIGN KEY (`Item id`) REFERENCES `items`(`id`) ON  
DELETE CASCADE ON UPDATE CASCADE;  
-- Constraints for table `medecine`  
ALTER TABLE `medecine`  
ADD CONSTRAINT `medecine_ibfk_1` FOREIGN KEY (`id`) REFERENCES `items`(`id`) ON DELETE  
CASCADE ON UPDATE CASCADE;  
-- Constraints for table `seeds`  
ALTER TABLE `seeds`  
ADD CONSTRAINT `seeds_ibfk_1` FOREIGN KEY (`id`) REFERENCES `items`(`id`) ON DELETE  
CASCADE ON UPDATE CASCADE;  
-- Constraints for table `tools`  
ALTER TABLE `tools`  
ADD CONSTRAINT `tools_ibfk_1` FOREIGN KEY (`id`) REFERENCES `items`(`id`) ON DELETE  
CASCADE ON UPDATE CASCADE;  
-- Constraints for table `trees`  
ALTER TABLE `trees`  
ADD CONSTRAINT `trees_ibfk_1` FOREIGN KEY (`id`) REFERENCES `items`(`id`) ON DELETE  
CASCADE ON UPDATE CASCADE,  
ADD CONSTRAINT `trees_ibfk_2` FOREIGN KEY (`TreeType`) REFERENCES `tree_type`(`id`) ON  
DELETE CASCADE ON UPDATE CASCADE;  
-- Constraints for table `users`  
ALTER TABLE `users`  
ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`membershipType`) REFERENCES `membership`(`id`)  
ON DELETE CASCADE ON UPDATE CASCADE;  
COMMIT;
```

## MOST SOFTWARE IMPLEMENTATIONS

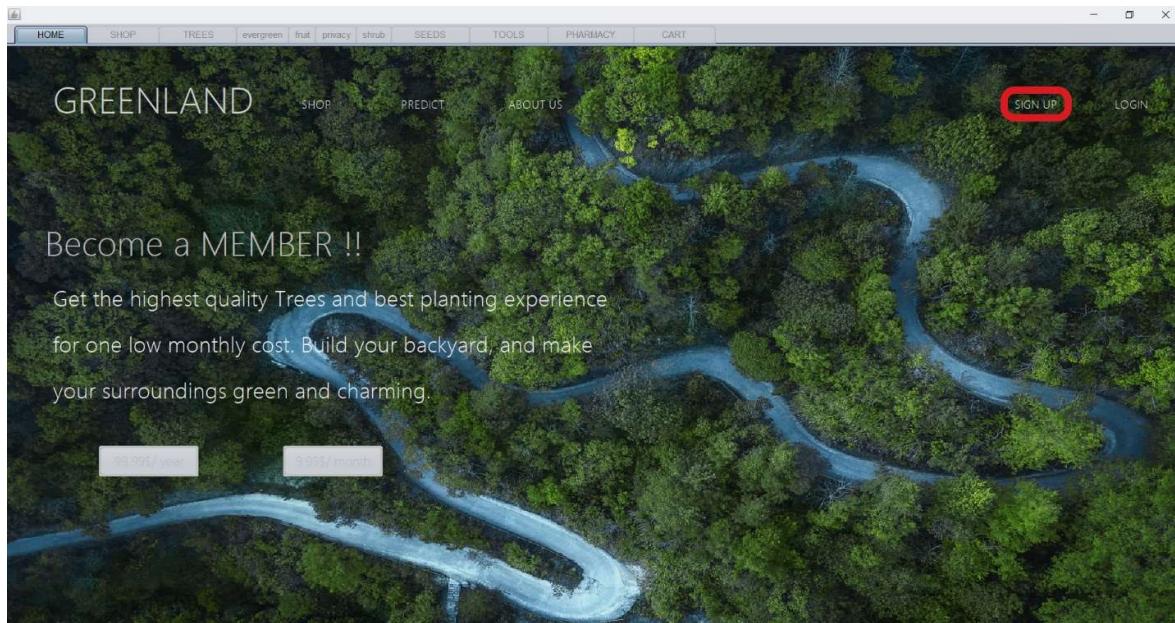


Figure 1: home page

This is the **HOME** page where the program starts, the first thing the customer must do is to create a new account in order to use it each time he wants to visit our site, and this is done by clicking the **signup** button at the top right of the screen.

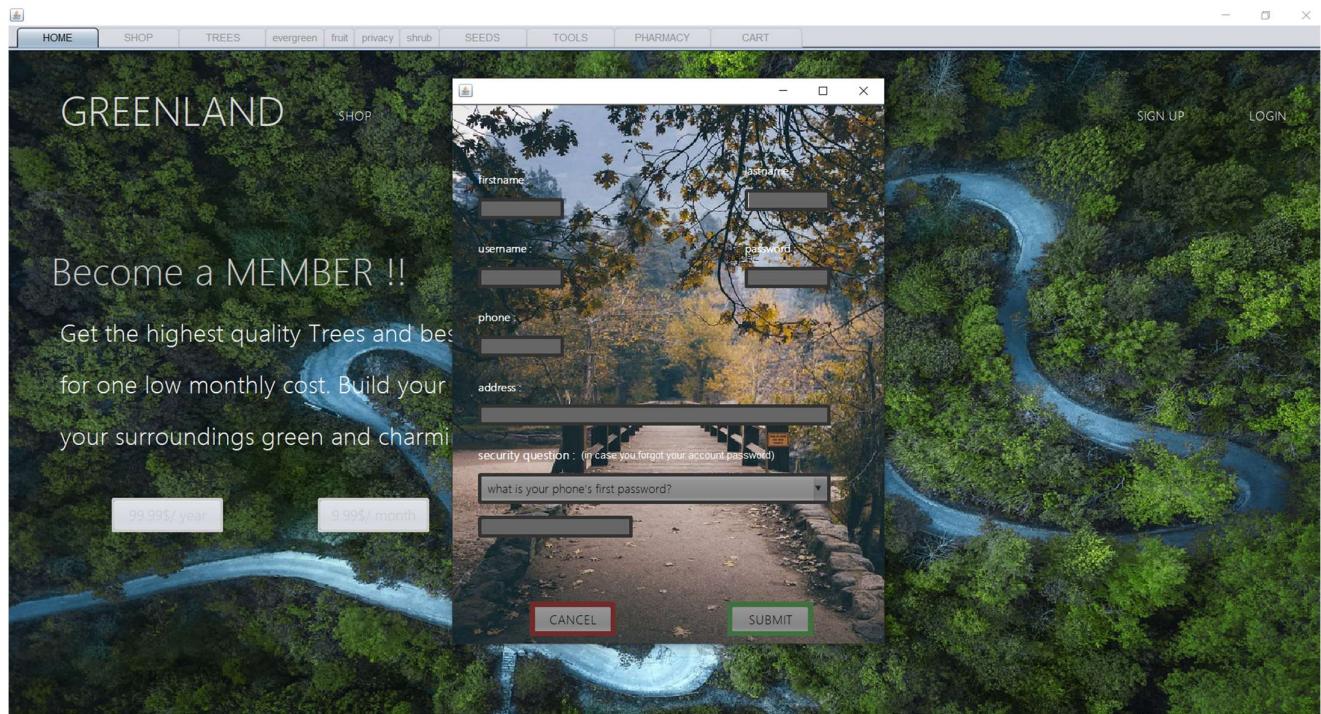
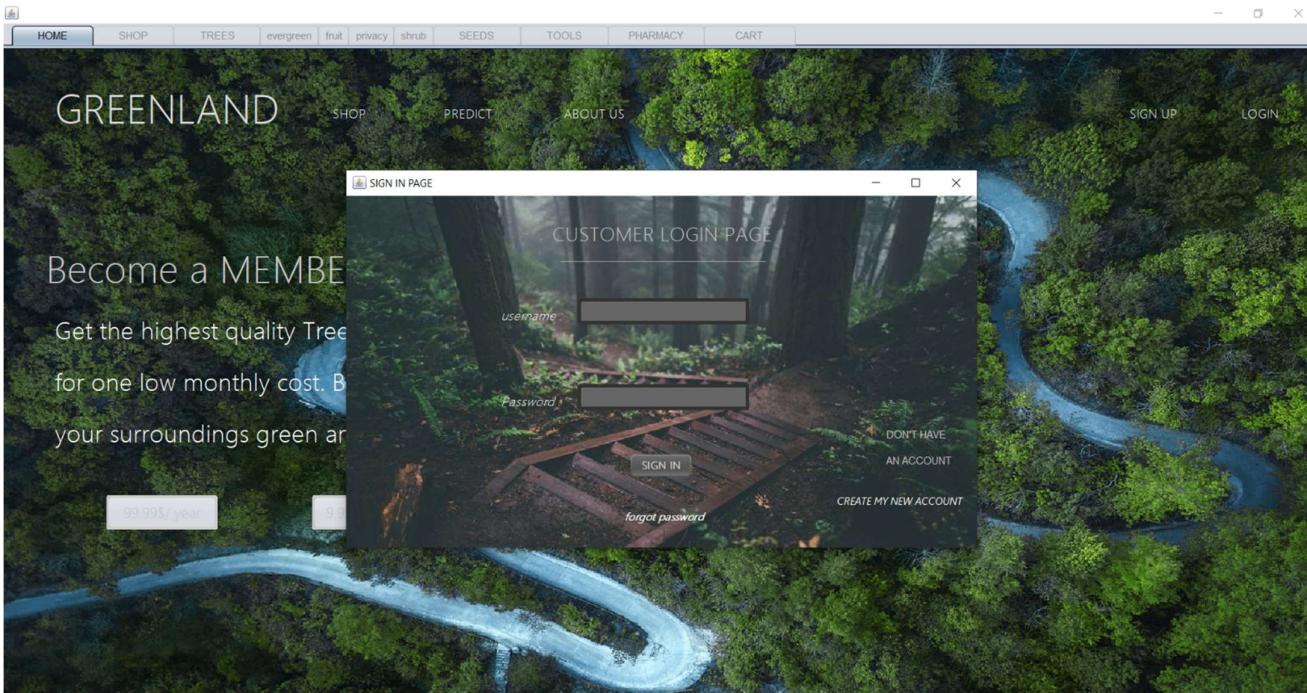


Figure 2: Sign up page

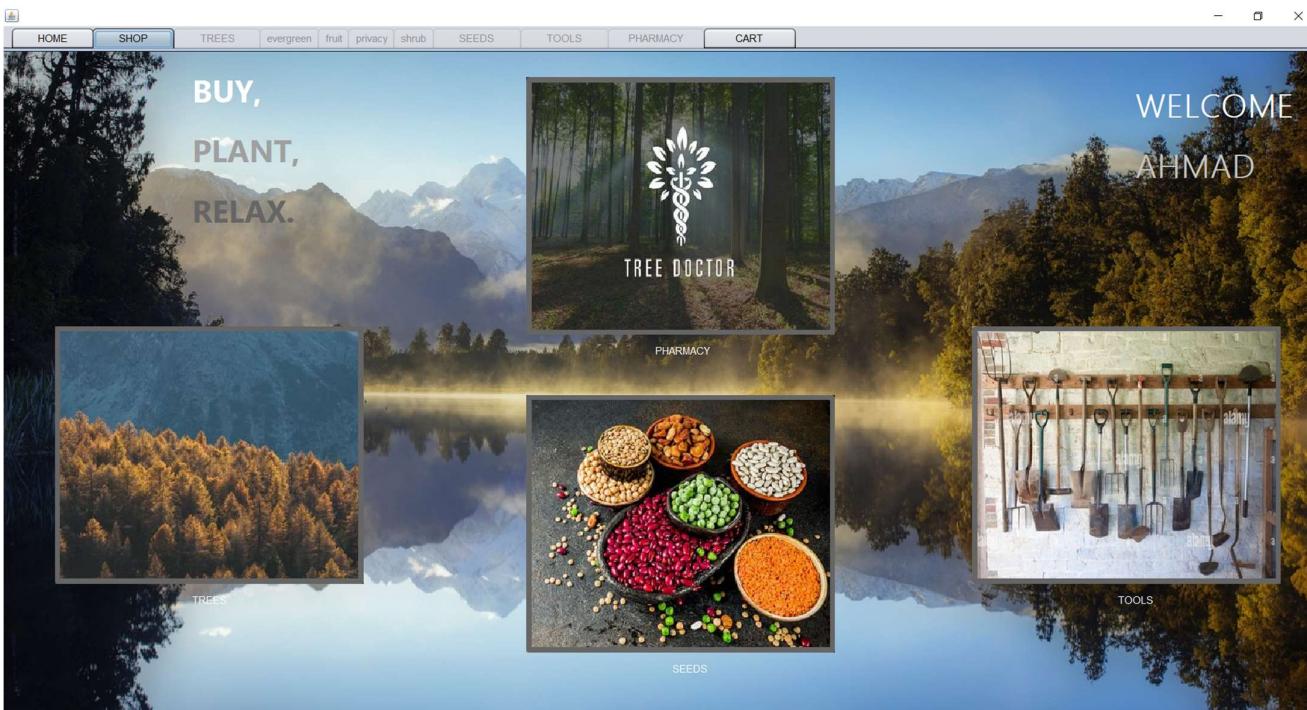
This is the **SIGNUP** page where the customer fills the required data in order for his account to be created. As we can see the customer must fill his first and last name, username and password to use them for logging in to the application, phone number in case we wanted to contact him, address for delivering his orders, as well as a security question to answer in case the customer forgot his account password. **Notes about this page:** (1) the password must be at least 8 characters, (2) all required data must be filled, otherwise the user will not be able to continue.



*Figure 3: Sign in page*

This is the **LOGIN** page where the customer logs in using the username and password that he previously created in the **SIGNUP** page.

Notes about this page: (1) the **forgot password** button is a feature that allows the user to update his password when correctly answering the security questions, (2) the **create account** button opens the **SIGNUP** page to create a new account in case the user pressed the **sign in** button without recognizing the **signup** button.



*Figure 4: Shops page*

After entering the correct username and password, the site directly takes the customer to the **SHOPS** page where he can choose the type of shop he wants to buy from.

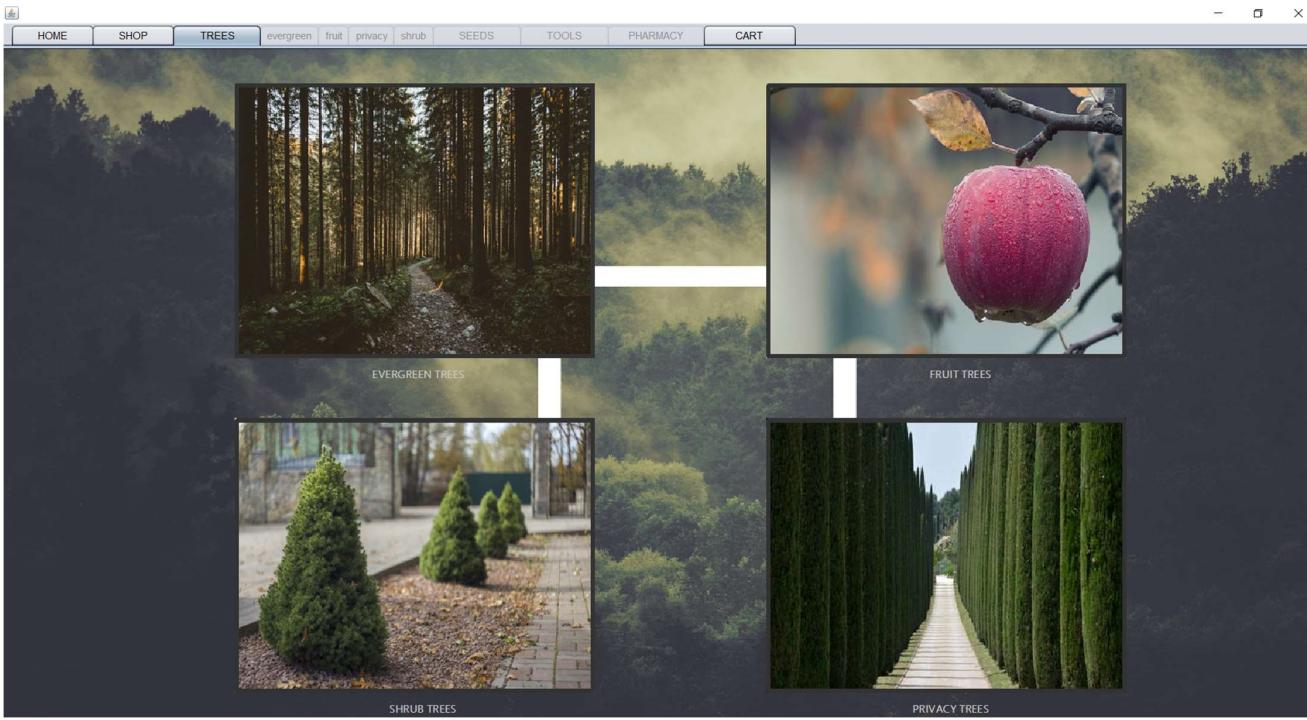


Figure 5: trees page

This **TREES** page opens when the user hits the **trees** button in the **SHOP** page.

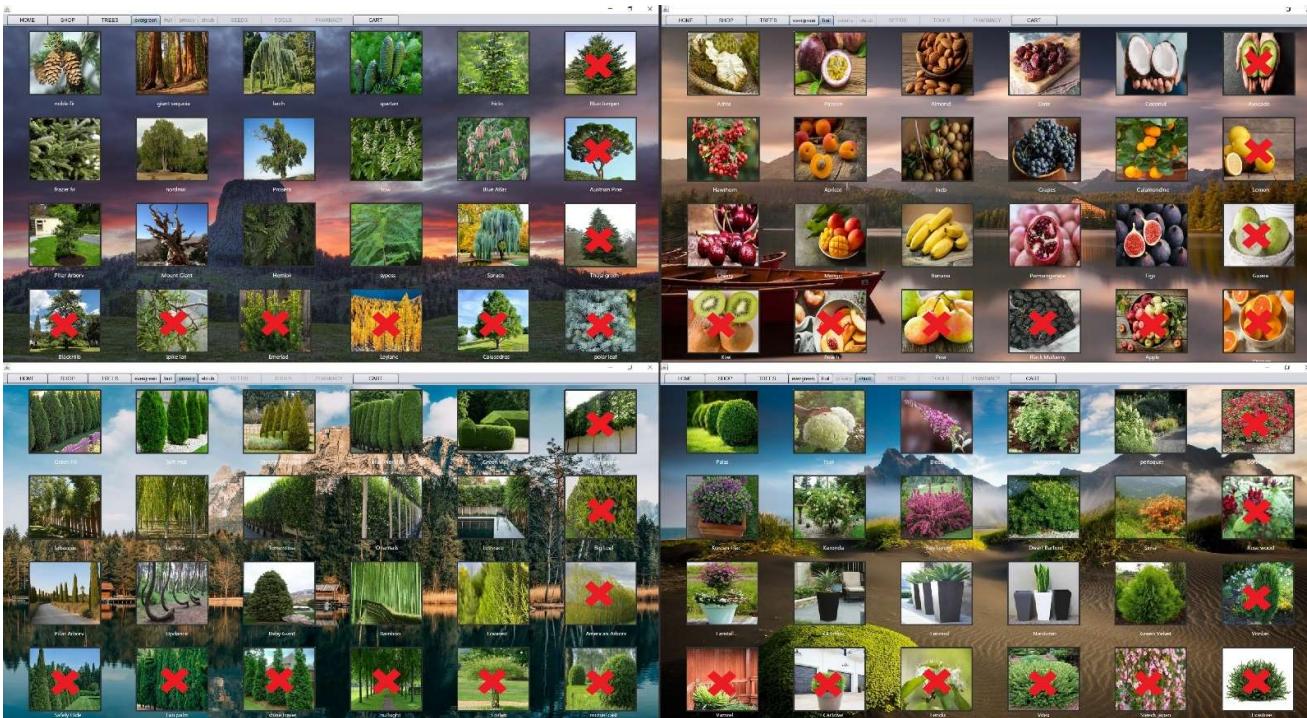


Figure 6: plants page

These are the **collection of trees** the customer can buy after pressing the type of trees he chose in the **TREES** page.

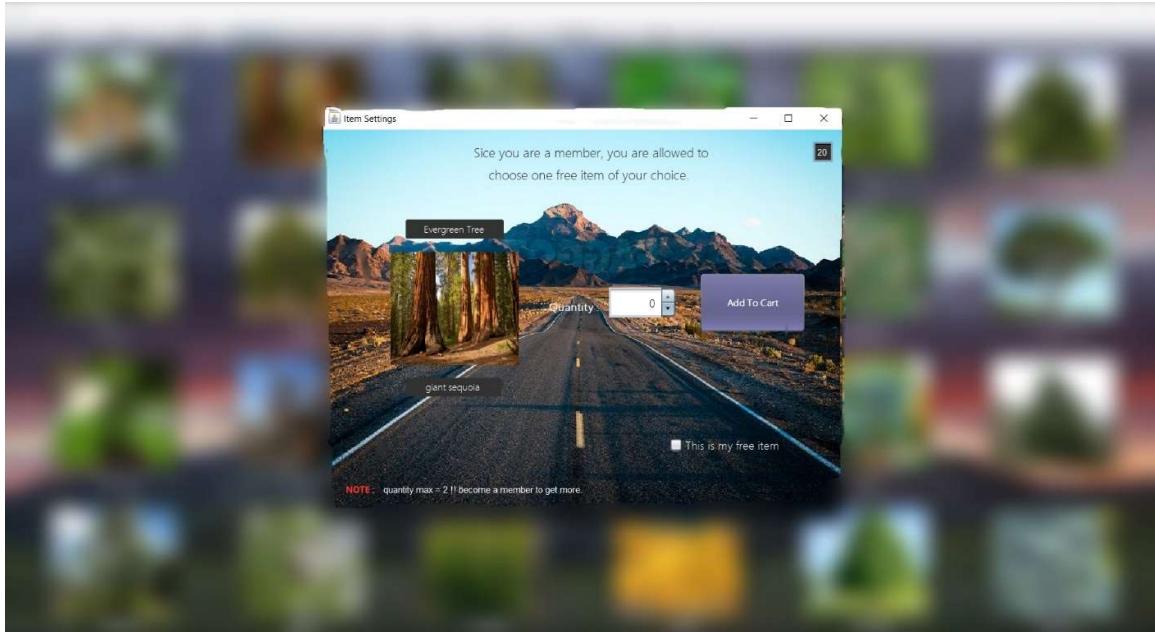
Notes about these pages: (1) the customer can only buy the items that are not labeled in red cross, whereas customers who subscribed to the membership can buy any item they want.



Figure 7: tools/seeds/medicines page

These are the pages corresponding to the Medicines, Tools, and Seeds Shops respectively.

Notes about these pages: (1) the customer can only buy the items that are not labeled in red cross, whereas customers who subscribed to the membership can buy any item they want.



*Figure 8: Item page*

This **ITEM** page opens every time the customer selects an item in order to 'select quantity' and 'add the item to the cart'.

**Notes about this page:** (1) the maximum quantity the customer can choose for any item is two, whereas the member can choose the quantity freely, (2) a special feature for our members is that they can choose 1 free item of their choice every month during the subscription interval.

The screenshot shows a 'Cart' page with a header menu: HOME, SHOP, TREES, evergreen, fruit, privacy, shrub, SEEDS, TOOLS, PHARMACY, and CART. The main content area displays a table of selected items with columns: ID, Item Type, Item Name, Quantity, Total Price, and Date. An item for 'Pharmacy' (ID 123) is highlighted. A tooltip 'Info about selected Item' points to this row. At the bottom of the table is a red 'DELETE SELECTED ITEM' button. The background features a scenic image of a forest and mountains. A green 'Purchase' button is located in the bottom right corner.

ID	Item Type	Item Name	Quantity	Total Price	Date
24	Evergreen Tree	normal	4	140.0	2022-10-26 22:59:17.994
4	Evergreen Tree	Yew	2	50.0	2022-10-26 22:59:22.143
5	Evergreen Tree	Hemlok	2	60.0	2022-10-26 22:59:22.153
123	Pharmacy	bilenthin	2	62.0	2022-10-26 22:59:36.463
128	Pharmacy	Etheophon	4	148.0	2022-10-26 22:59:39.047
108	Farmer Tools	Rake	2	8.0	2022-10-26 22:59:44.775
101	Farmer Tools	Saw	1	8.0	2022-10-26 22:59:47.955
189	Seeds	Corander	3	19.5	2022-10-26 22:59:54.415

*Figure 9: Cart page*

Here is the **Cart** where the customer added every item in it after pressing the **add to cart** button in the **ITEM** page, when everything is set, the customer presses the **purchase** button to complete the purchase of the left items in the cart.

**Notes about this page:** (1) the user can delete any item from the cart by selecting it and hitting the **delete selected item** button, (2) the user can view information about any item existing in his cart by

selecting it and pressing the **info about selected item** button, Moreover he can update the quantity of any item he wants buy simply typing the new item quantity he wants in the Quantity column.

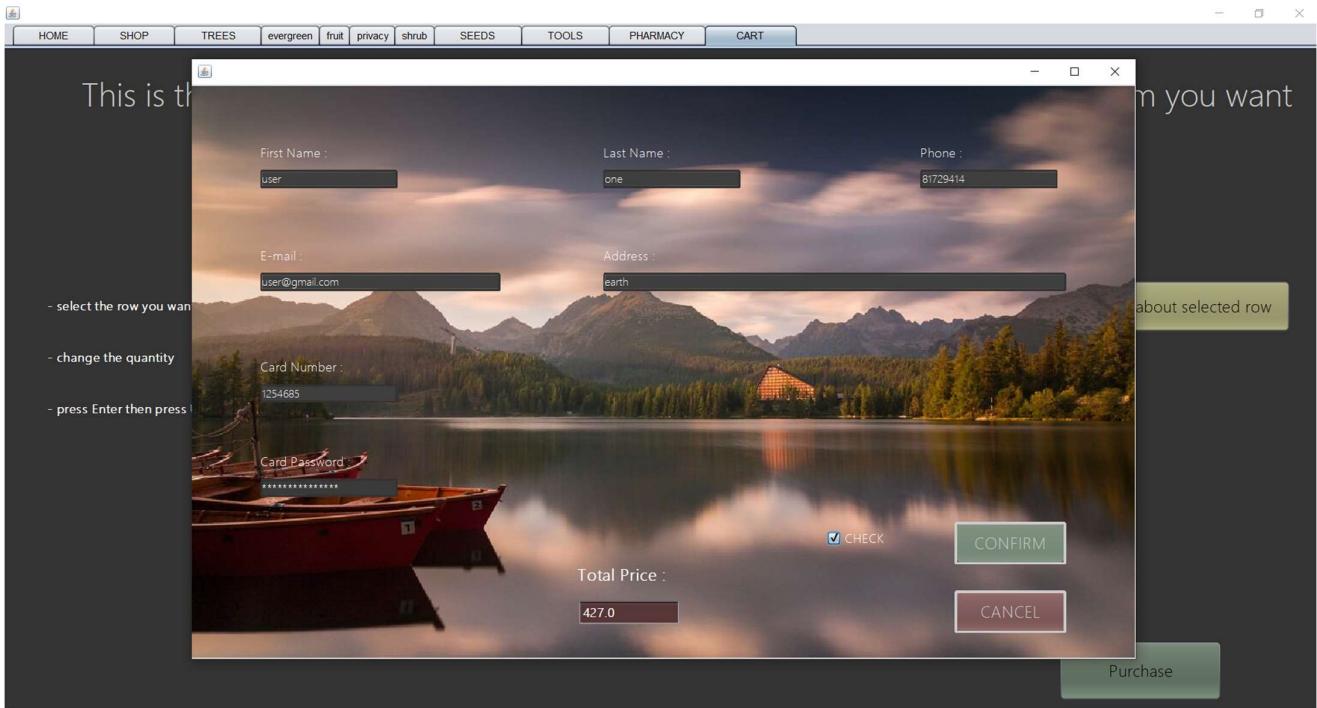


Figure 10: Purchase page

This **PURCHASE** page opens when pressing the **purchase** button in the **CART** page.

Notes about this page:(1) all required data must be filled + checking the small 'check' box in order to complete the purchase, otherwise the user will not be able to continue.

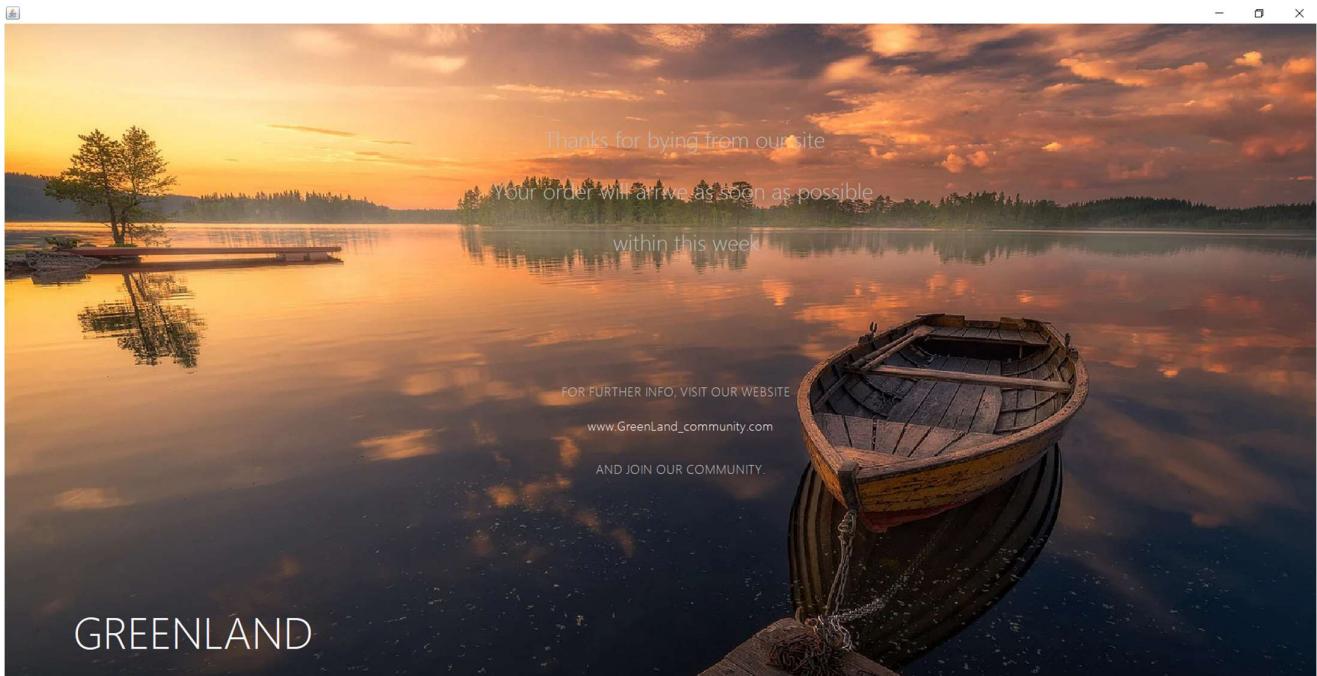
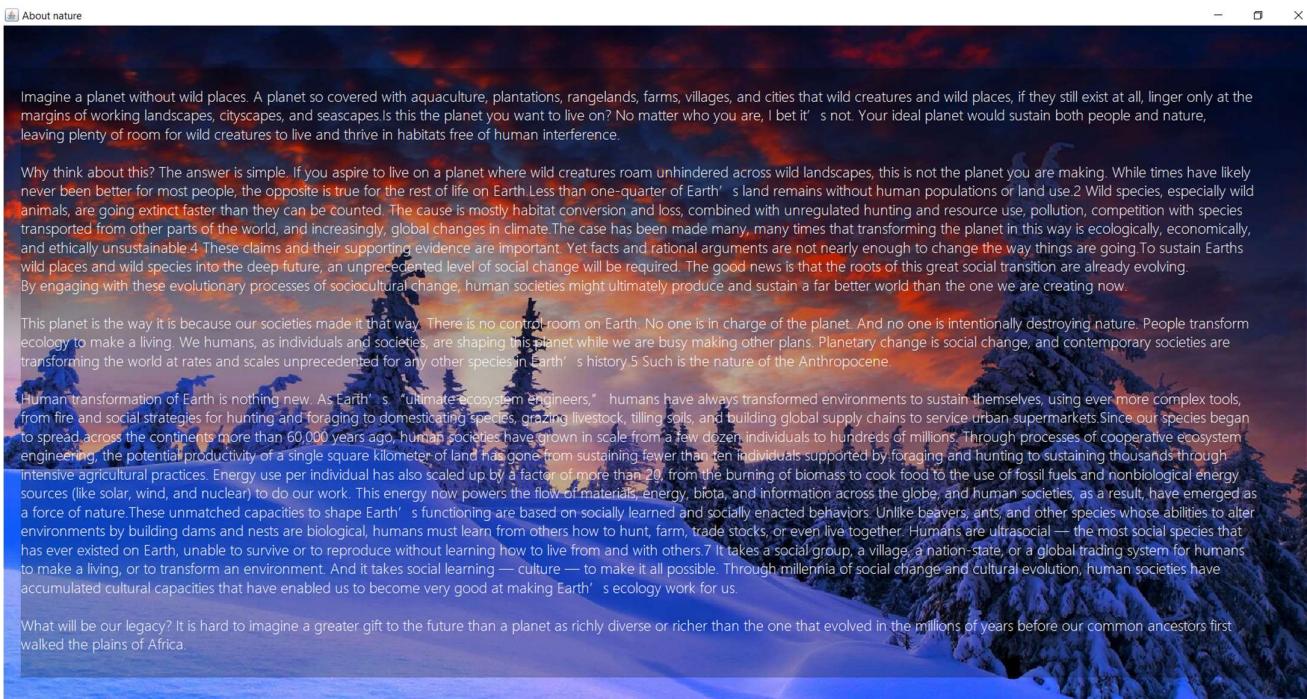


Figure 11: Thanks page

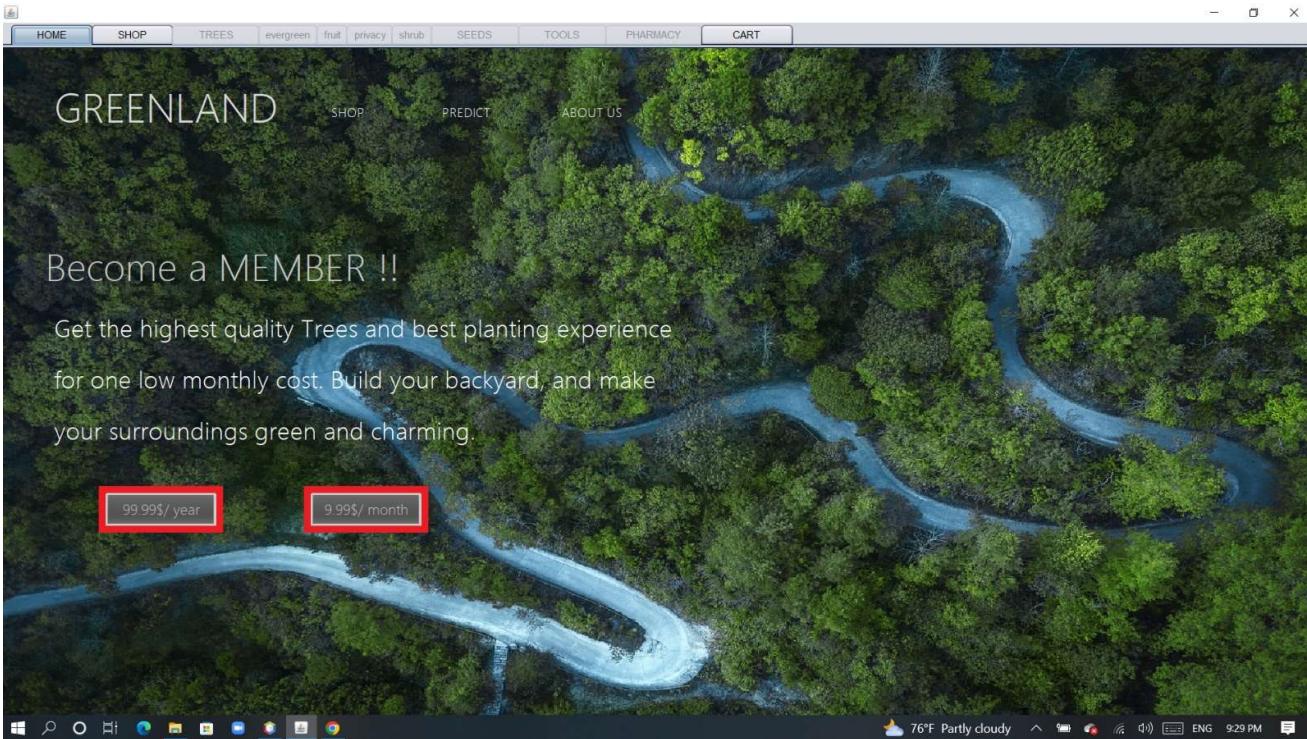
The **END** page appears after confirmation is done in the **PURCHASE** page.

Notes about this page:(1) customer can go back to **HOME** page by pressing the **GREENLAND** label at the button left of the screen, (2) customer can visit our website through the **link**.



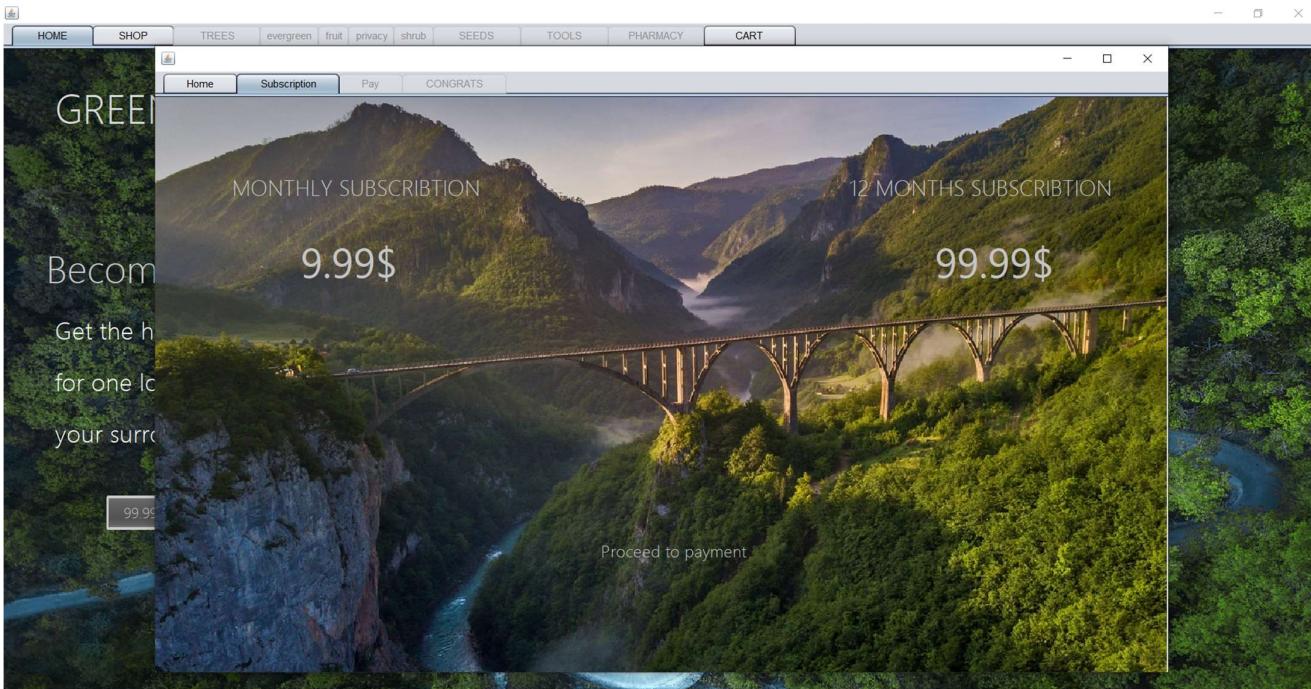
**Figure 12: About us page**

This **ABOUT** page shows an interesting article about nature and related info.



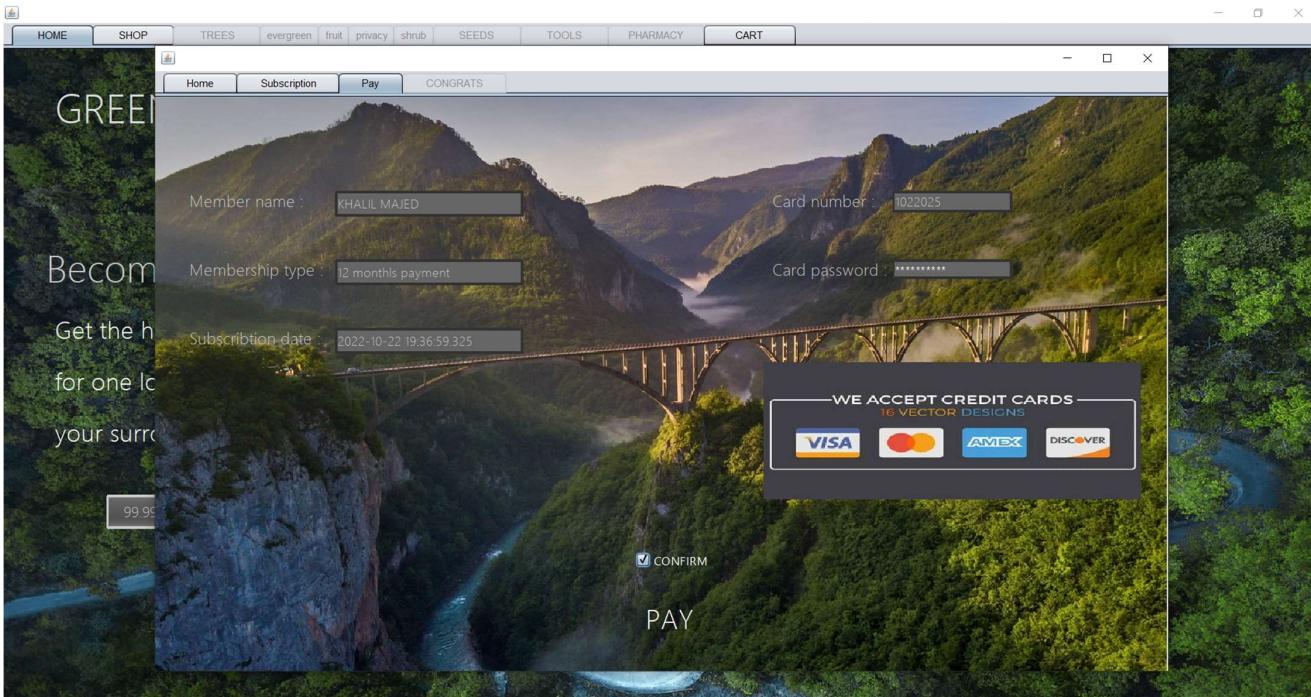
**Figure 13: membership in home page**

After logging in, customer can sign up to our membership by pressing one of the two buttons labeled above, and by becoming a member customer can have access to more items and variety of special features.



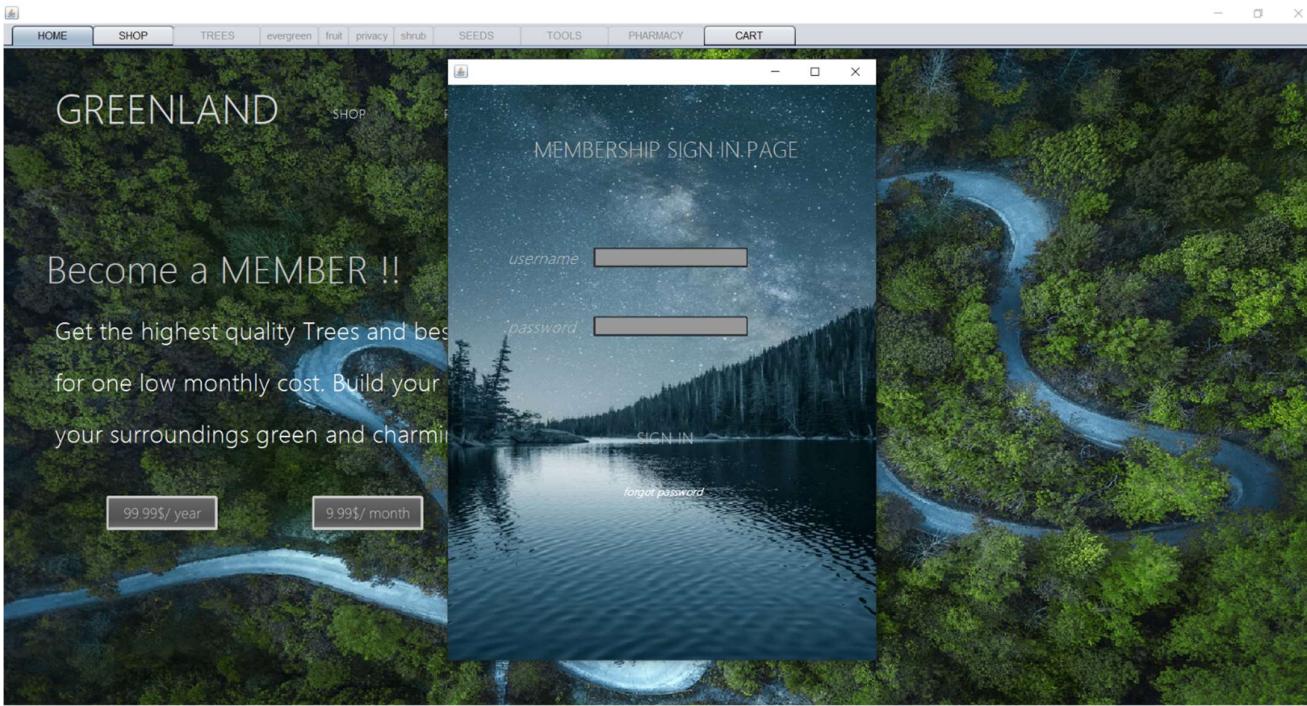
*Figure 14: membership subscription*

This is the **MEMBERSHIP SUBSCRIPTION** page that appears when pressing any button of the two membership subscription buttons in the **HOME** page, where customer can reselect the subscription type if he decided to change his mind about choosing the subscription type in the **HOME** page.



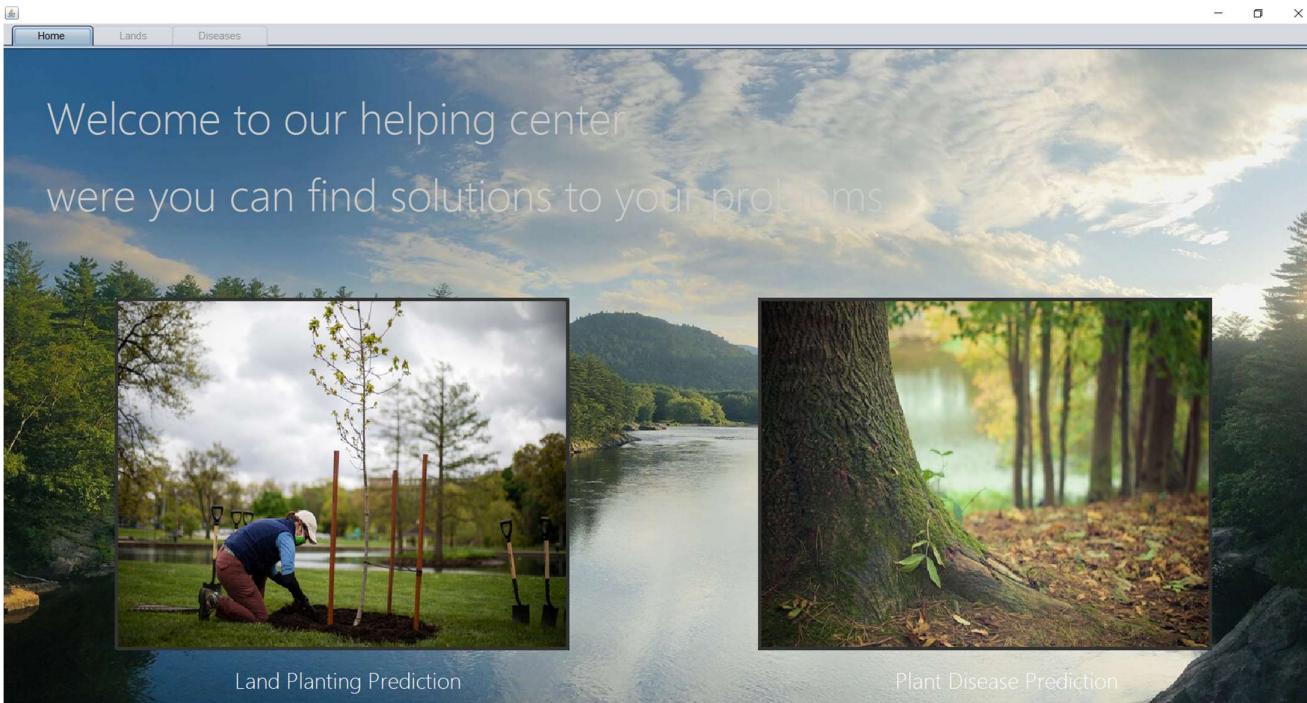
*Figure 15: membership payment*

This is the page that appears after hitting the **proceed to payment** button, the customer must fill all required data + check the small 'confirm' box in order to complete the purchase, otherwise the user will not be able to continue.



*Figure 16: member login page*

This is the **MEMBER LOGIN** page where the customer relogs in as a new member by entering his correct username and password.



*Figure 17: Help center page*

This is the **HELP** system where customer can ask and gets answers about **lands** and **diseases**, this system basically tells you what types of agricultures you can plant depending on your land type and also gives some tips and ways on how to enhance the productivity of your land, furthermore it tells you the type of disease your tree is suffering from and how to treat it through following basic instructions.

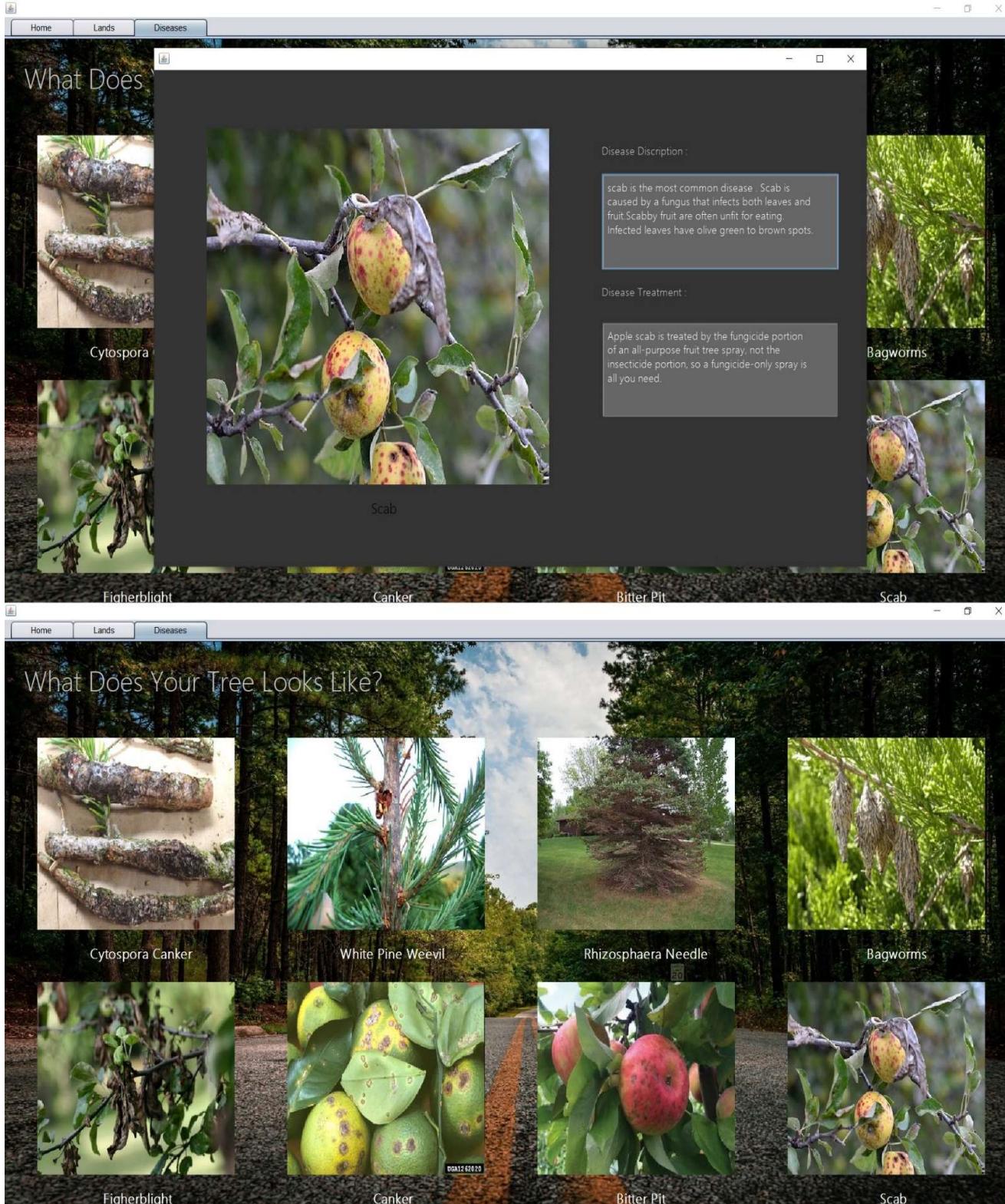


Figure 18: diseases page

An **INFORMATION** page opens showing information needed to know about the selected choice when pressing the **Plant Disease Solutions** button, revealing the disease description and ways to of treatment.

Home   Lands   Diseases

### Sand Soil Land



Characteristics	What To Plant	How To Improve
<p>Known for its light, warm, dry, easy digging and sometimes tends to be acidic and low in nutrients. Sandy soils are often known as light soils due to their high proportion of sand and little clay. These soils have quick water drainage and are easy to work with. Sandy soils are less fertile than other soil types, and more prone to drying out, because they're made up of relatively large particles. This means there are cavernous gaps between the particles, making it easy for water (and water-soluble nutrients) to filter down through the soil, out of the reach of plant roots.</p>	<ol style="list-style-type: none"> <li>1. Sedum</li> <li>2. Lavender</li> <li>3. Artemisia</li> <li>4. Carrots</li> <li>5. Potatoes</li> <li>6. Radishes</li> <li>7. Cucumbers</li> <li>8. Red chokeberry</li> <li>9. Butterfly bush</li> <li>10. Daylilies</li> <li>11. Giant allium</li> <li>12. Herbs</li> <li>12. Sweet alyssum, And other Root Vegetables....</li> </ol>	<p>There are a number of soil amendments that can help improve sandy soils. The number one thing to do to improve this kind of soil is to increase nutrient and moisture retention by improving the soil condition. The best way to do this is to frequently and regularly add organic matter to your soil. Layer in compost, wood chips, composted manure, straw, shredded leaves, grass clippings, and other mulches to help improve the organic matter content of your soil. You can also add peat moss, coco coir, or vermiculite to your soil to help improve moisture retention. However, these will not add nutrients to your soil.</p>

Home   Lands   Diseases

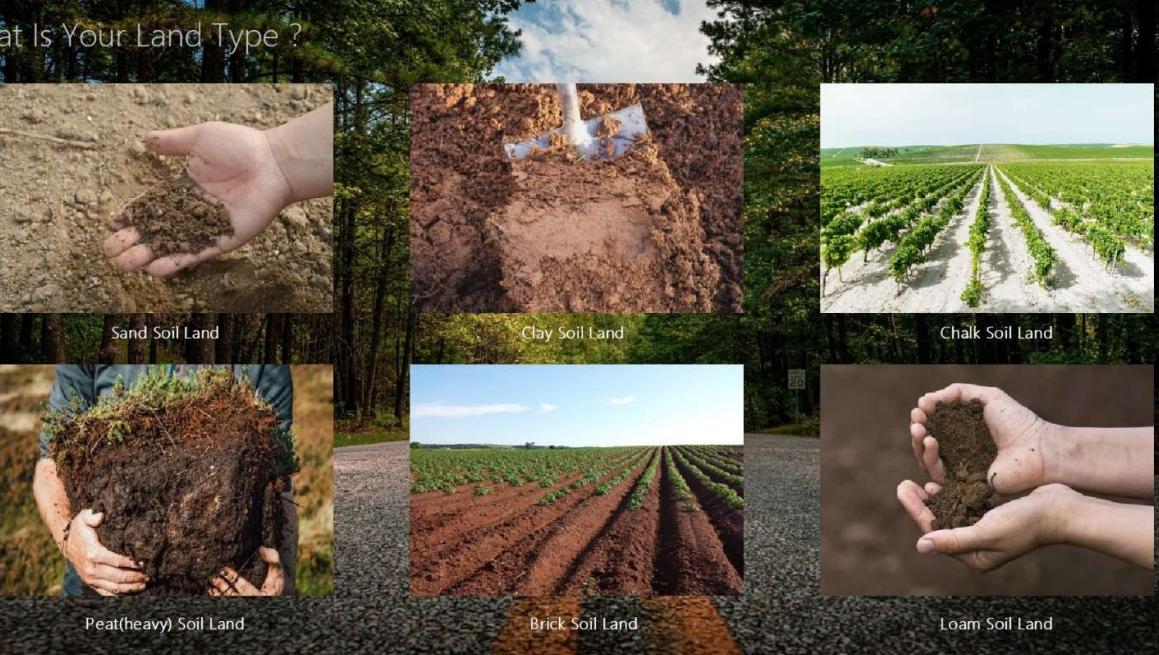
### Peat(heavy) Soil Land

### Brick Soil Land

### Loam Soil Land

76°F Sunny ENG 10:09 AM

### What Is Your Land Type ?



**Figure 19: lands page**

Another **INFORMATION** page opens showing information needed to know about the selected choice when pressing the **Land Type** button, revealing the Land characteristics and ways to of Improvement.

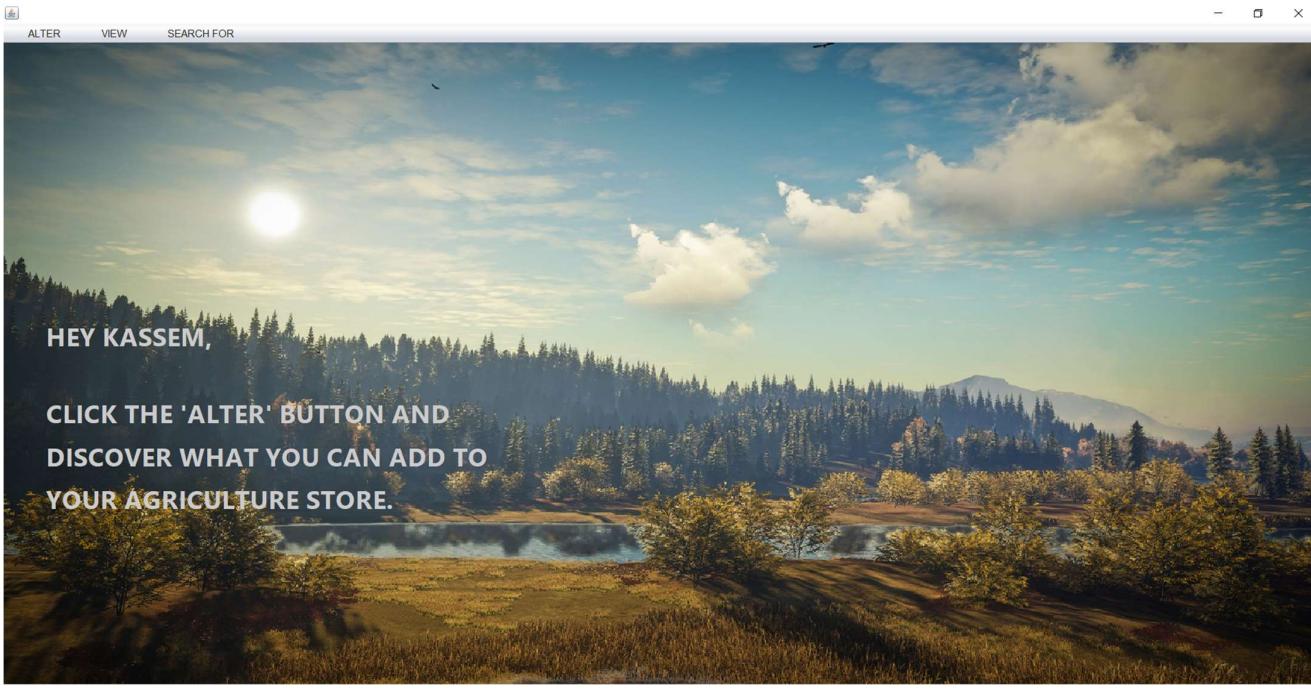


Figure 20: Admin home page

This is the **ADMIN HOME** page, this page allows the admin to take several actions concerning his program such as adding, deleting, and altering new Items and adding new admins by the owner only through the **alter** button, furthermore admins can view the invoice purchased by his customers, check the store house through the **view** button, as well as search for any item through the **search for** button.

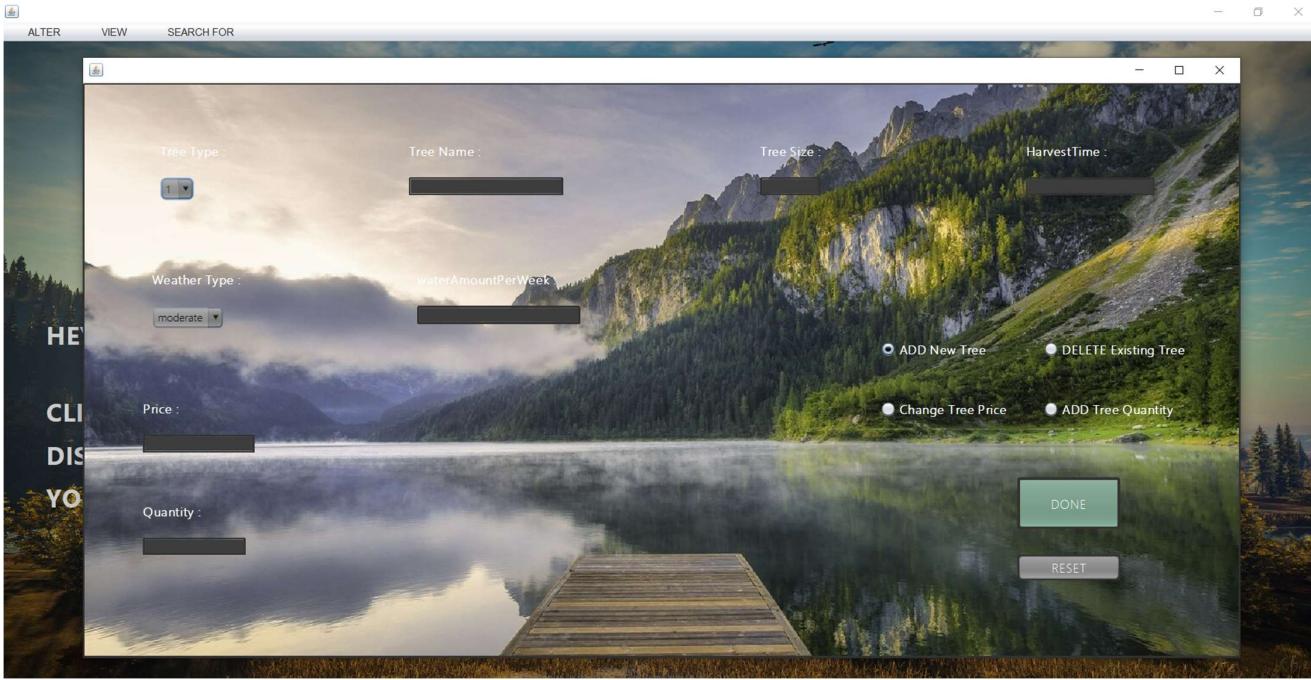
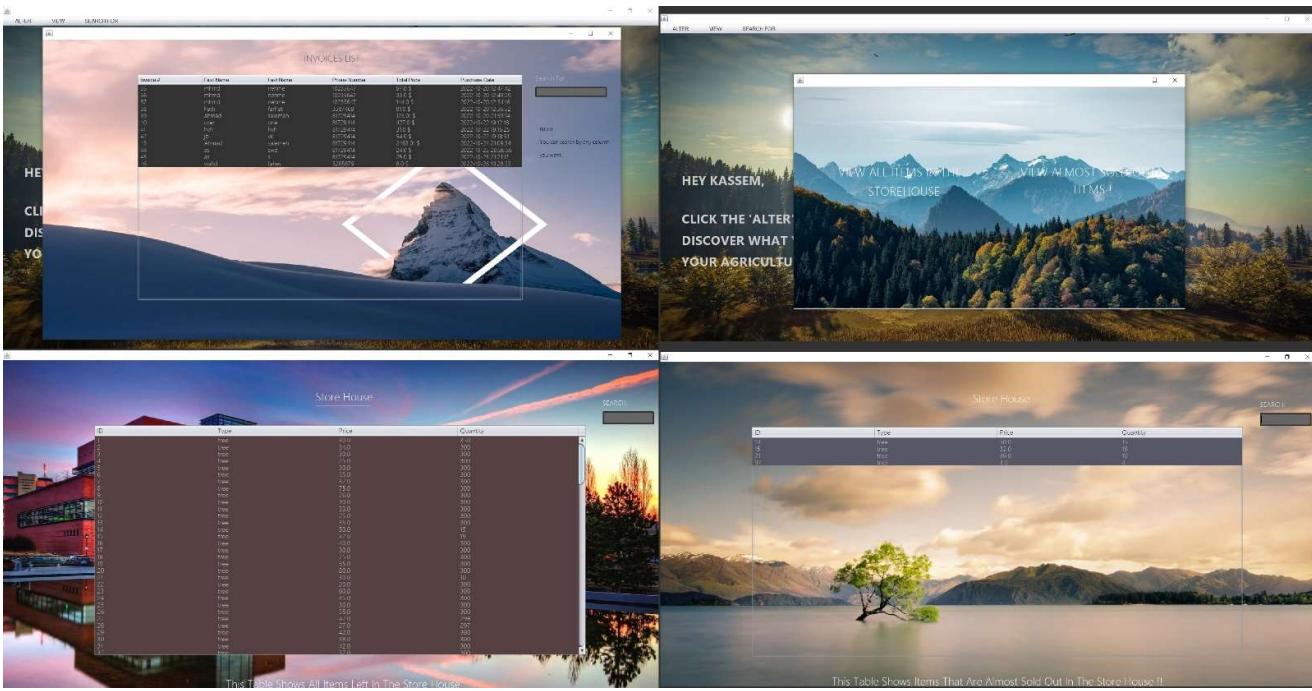


Figure 21: Alter page

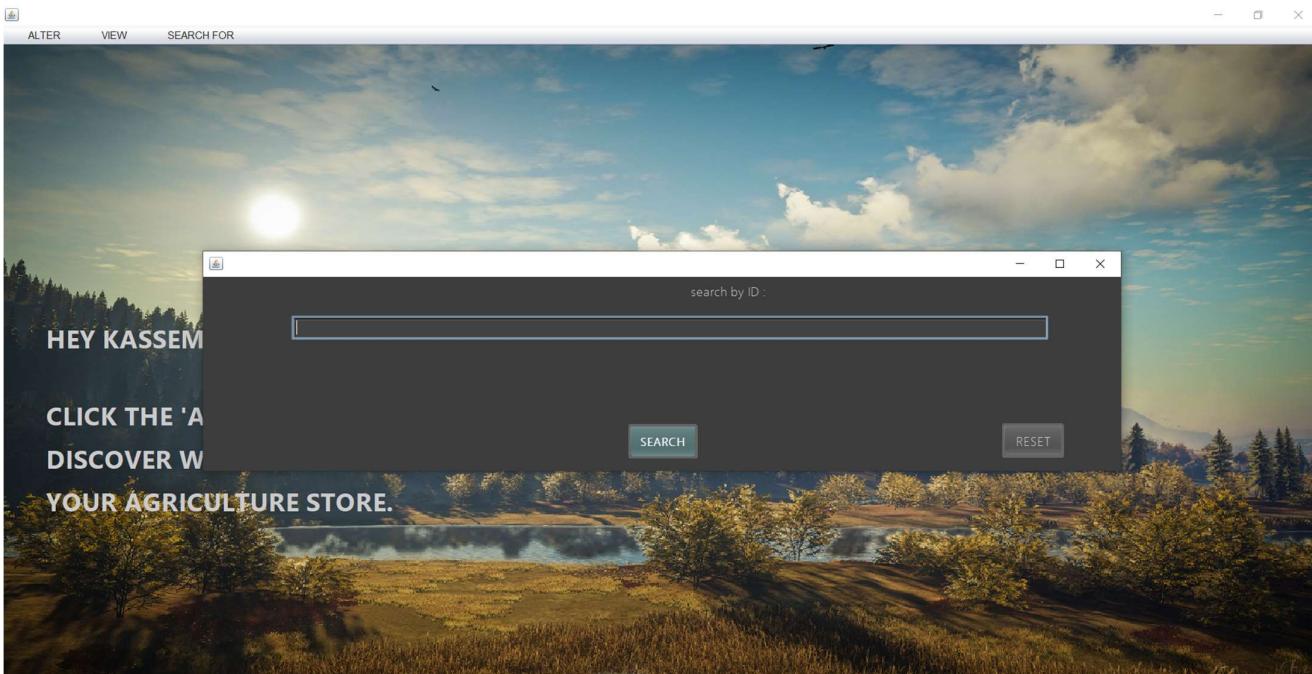
This is a brief example on how the **ALTER** page looks like when the admin hits the **alter** button in the **HOME** page, as we can notice, the admin enters all the info needed in order to add or delete any item.

Notes about this page: (1) all required data must be filled, otherwise the user will not be able to continue.



**Figure 22:** View page

This page shows what the **VIEW** button in the **HOME** page can do when choosing view **invoices** or view **storehouse** buttons respectively, in either case the admin can search for any item he wants using the search field.



**Figure 23:** 'search for' page

This is the **SEARCH** page where the admin can search for anything that is related to his shop from items, another admins, his customers, and even his workers.

## CUSTOMER ACTIVITY DIAGRAM:

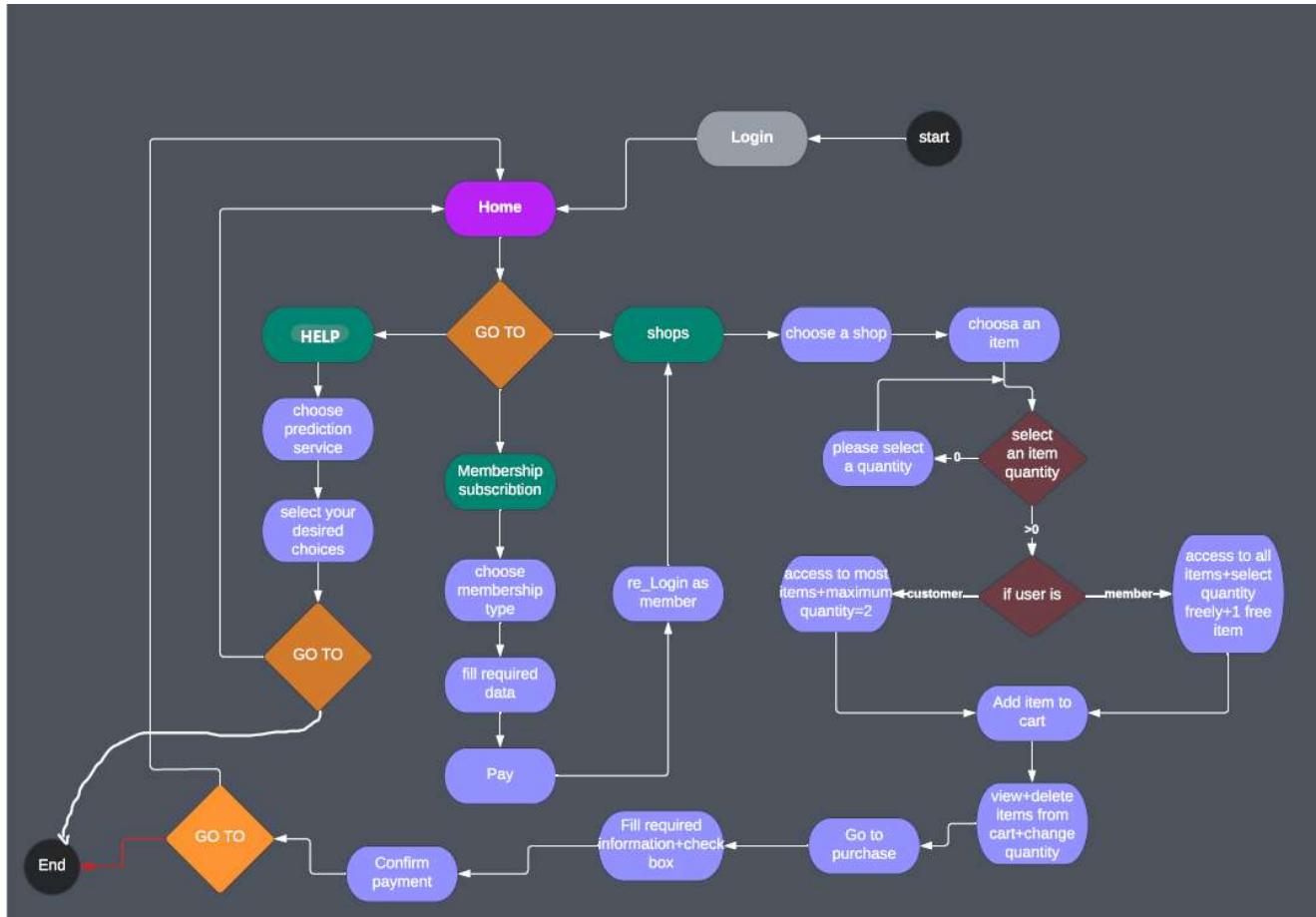


Figure 24: Customer activity diagram

This customer activity diagram shows the **basic** flow of the application when customer logs in, as we noticed the program starts by logging in to the application using the already created account. After logging in, the customer will have access to:

- SHOPS:** the customer after accessing the shops page, he will choose one of the 4 shops (Trees, Tools, Medicines, Seeds) then he will choose an item of his choice from the shop. Customer can't add any item to the cart if the quantity is zero, otherwise if the customer is a **member**, he can select the quantity freely as well as one free item of his choice, else if the customer is **not a member** the maximum quantity the customer can select for any item is '2'. After selecting the quantity, the customer will add the item to the cart, and the same rules are applied for any item he chooses. In the **cart** page the customer can view the items he added to cart in the shops and delete any item of his choice as well as viewing some info about any item in the cart, after he's done from viewing the cart, customer will now go and fill the required info to complete the purchase process.
- Membership subscription:** customer can go and subscribe for the membership through the subscription section in the **home** page where the customer chooses the subscription type (monthly or yearly) and then filling the required info to complete the payment method.
- HELP CENETR:** customer can also go to the help center from the home page to see what the help system can provide info about how to treat some common trees diseases as well as info about lands characteristics and how to improve it.

## ADMIN ACTIVITY DIAGRAM:

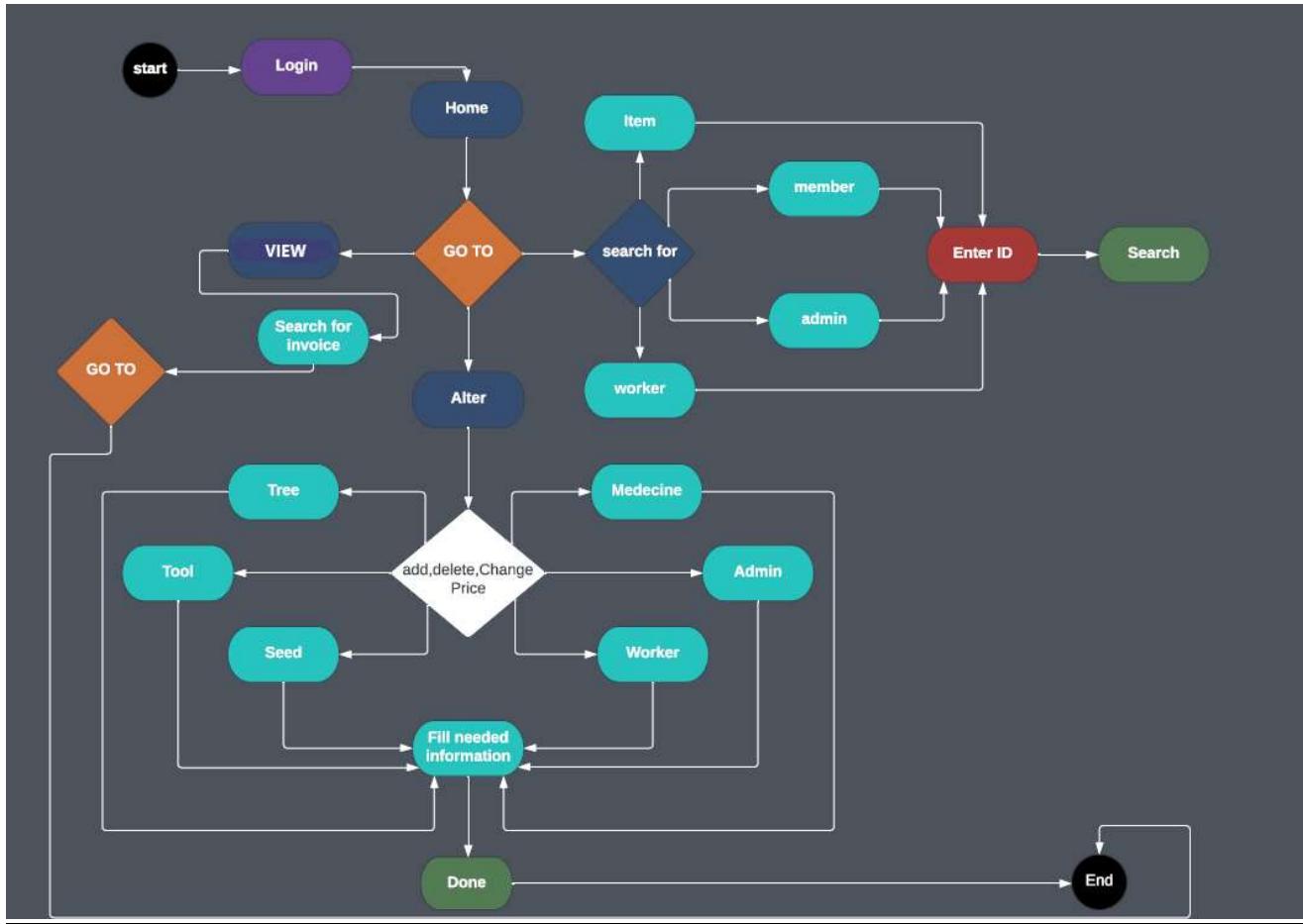


Figure 25: Admin activity diagram

This admin activity diagram shows the **basic** flow of the application when admin logs in, as we noticed the program starts by logging in to the application using an existing admin account created by an admin. After logging in, the admin will be able to do some actions:

1. **ALTER:** here in the altering section, the admin can do several actions such as: add/delete/change price and quantity of any item in the storehouse as well as add/delete workers and co-admins by filling the needed info. NOTE: owners who are also admins can add a new admin.
2. **VIEW:** here the admin can view all the invoices purchased by customers as well as viewing the remaining items in the storehouse.
3. **SEARCH FOR:** here the admin can search for any item/worker/member/co-admin in his system through the ID.

## CUSTOMER SEQUENCE DIAGRAM

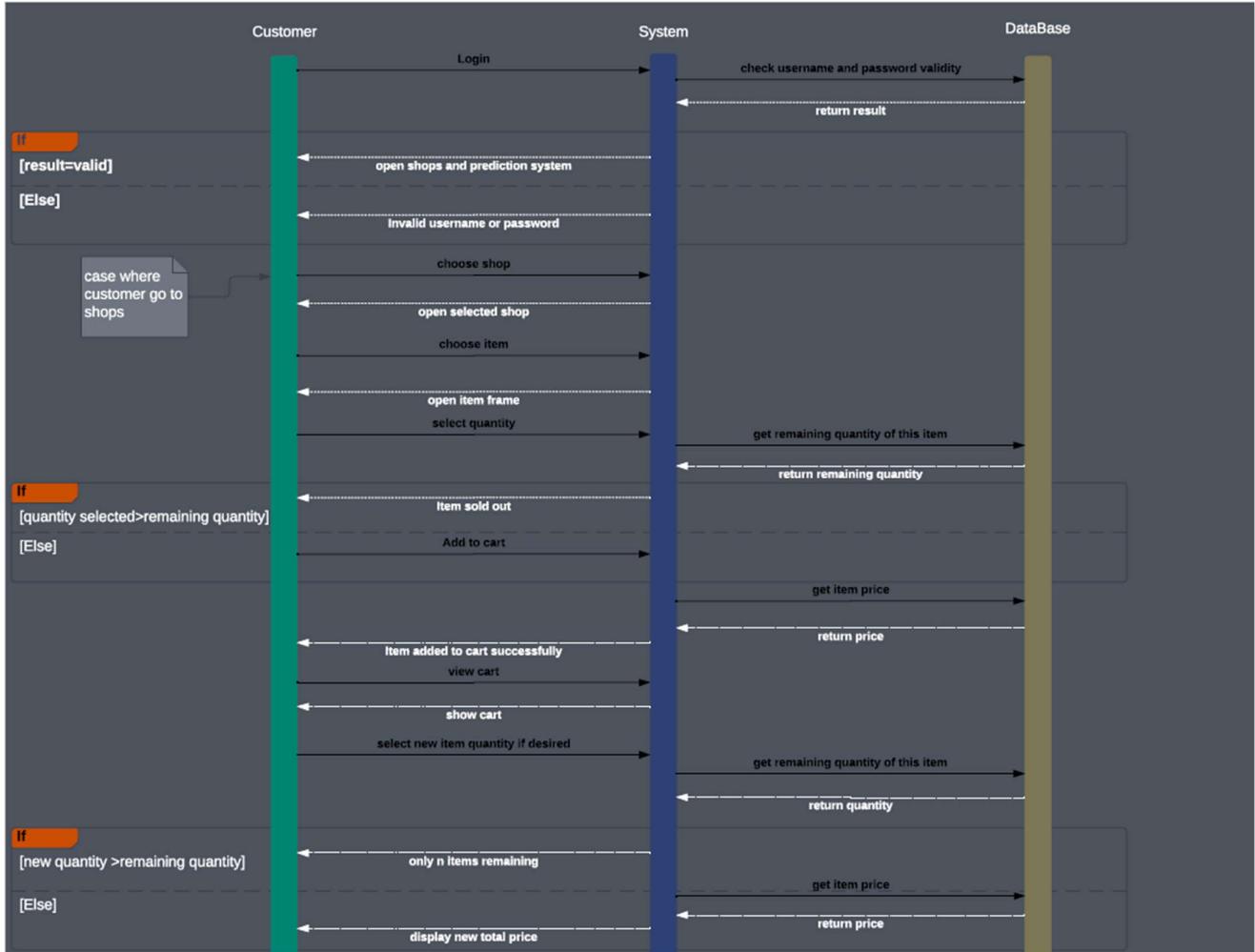


Figure 26: Customer sequence diagram (1)

In this part of the customer sequence diagram, the customer starts by logging in to the application, after hitting the sign in button in the login page, the program checks if the entered username and password exists in database. If not, the system informs an invalid username or password message to the customer, otherwise the database returns valid to the program and the customer directly goes to the **shops page** where he chooses a shop then chooses an item of his choice to select its quantity (maximum quantity for memberless customers is '2'), after selecting quantity the system directly checks the remaining quantity of this item in the storehouse. If the remaining quantity is less than the quantity selected, the system asks the user to select less quantity, otherwise if the customer wants to add the item to the cart the system gets the price of this item from the database and multiplies it by the selected quantity and adds the item with its selected quantity and total price to the cart as well as the item type, and of course the customer can choose to view the cart anytime he wants where he can change the quantity of any item and when doing so the system directly re-calculates the new total price depending on the item price multiplied by the new selected quantity, also customer can delete any existing item he wants from the cart.

## CUSTOMER SEQUENCE DIAGRAM(Continue)

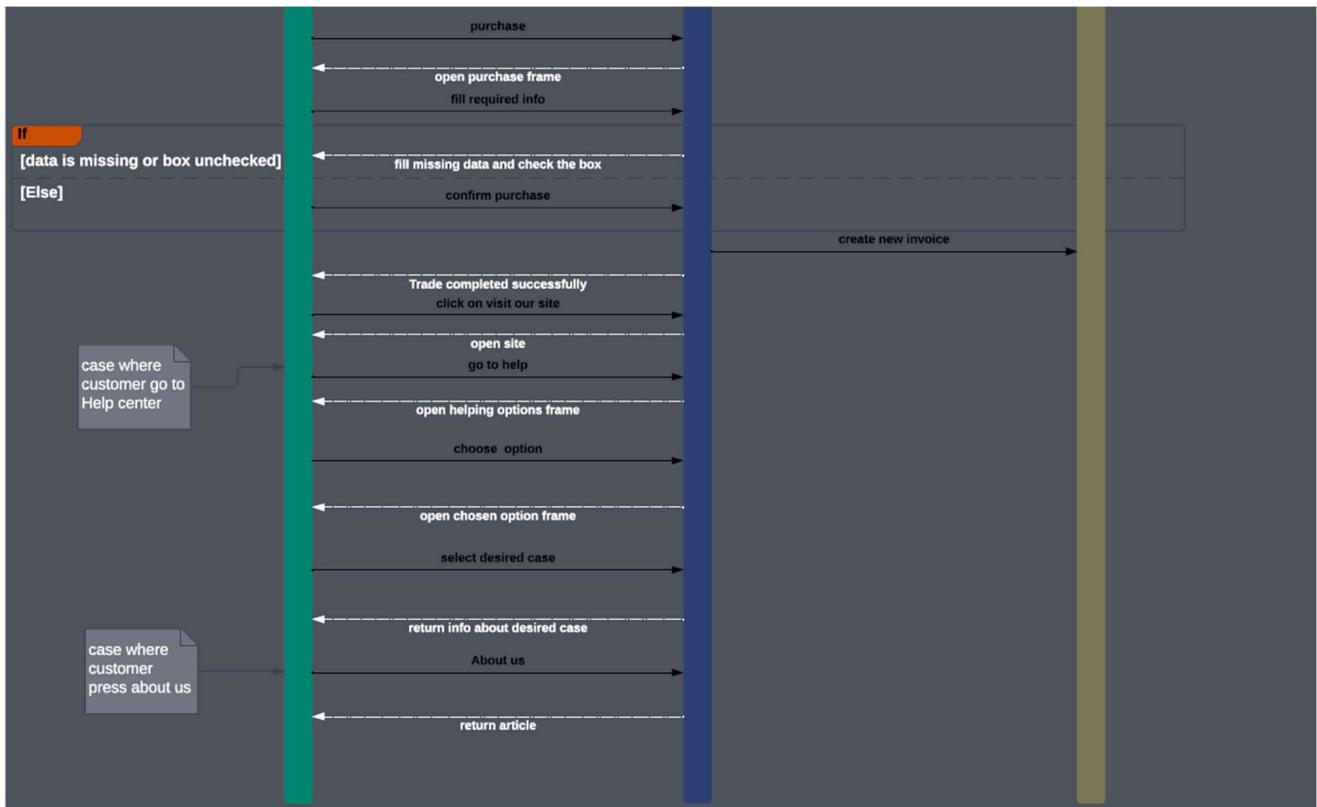


Figure 27: Customer sequence diagram (2)

After the customer is done deciding the items he wants, he will now go to the purchase page where he fills the required info to complete the purchase process, and the purchase can't be confirmed if there is any missing info.

Now concerning the **help center** customer can choose the type of help he wants from the system to help him with such as knowing disease description and ways to treat it, as well as viewing information about lands types characteristics and how to improve it.

As for the **about us** page, the system shows an article about importance of trees and nature and their effect on human life.

## ADMIN SEQUENCE DIAGRAM

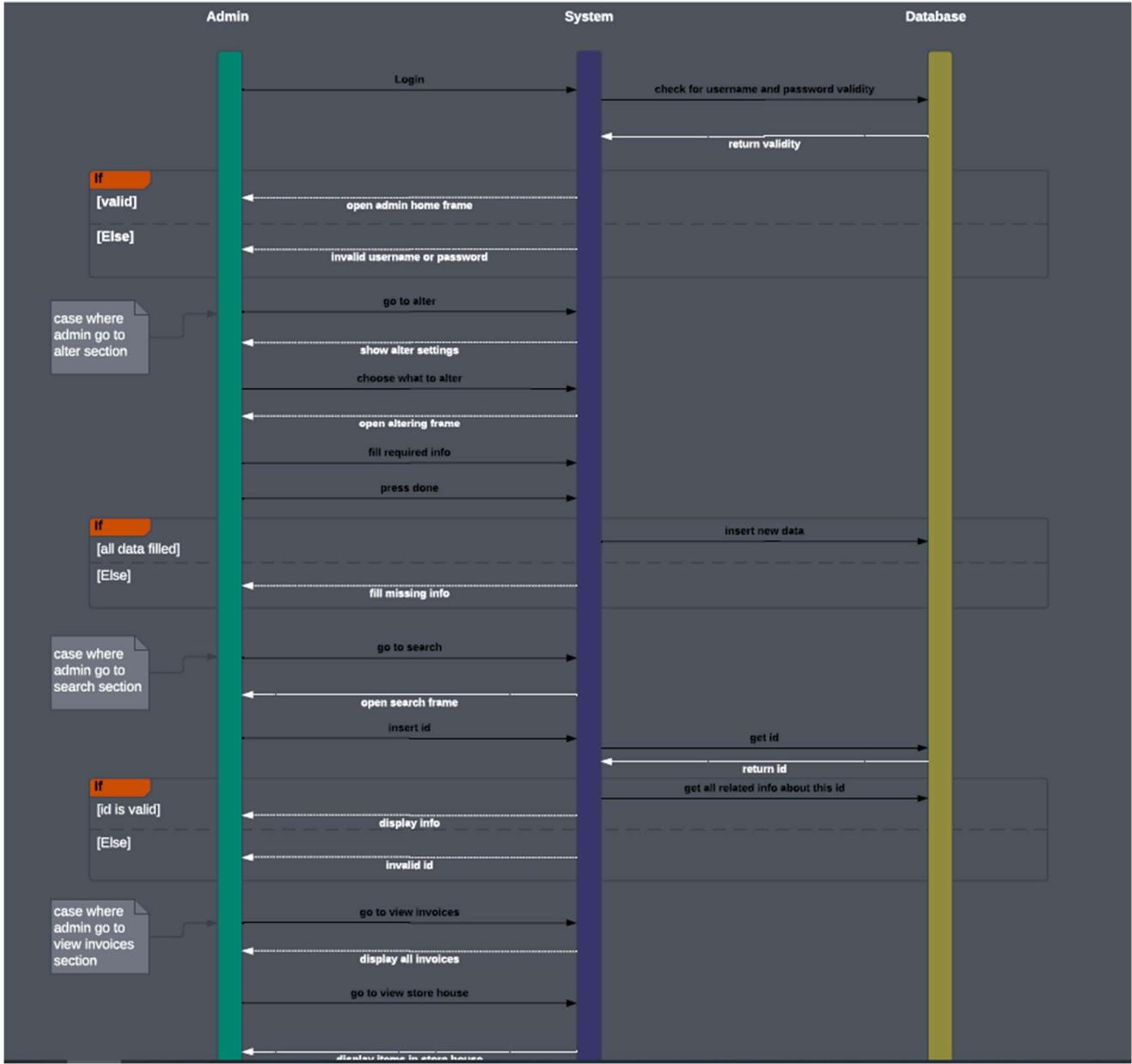


Figure 28: Admin sequence diagram

This is the admin sequence diagram showing basic details about how the system, database, and admin interacts with each other to complete the desired actions done by admins. As usual the admin must log in through an existing account, if the entered username and password doesn't exist in the database, the system informs an invalid username or password message to the admin, else if valid the system takes the admin directly to the admin home page where he **can** add/delete items, workers, and admins as well as change price and quantity of any item through filling the required info.

As for the **view option**, the admin can view all invoices purchased by customers and view the remaining items in the storehouse.

As for the **search for** section, the admin can search for any item in the database by entering a valid ID.

## CONCEPTUAL DIAGRAM

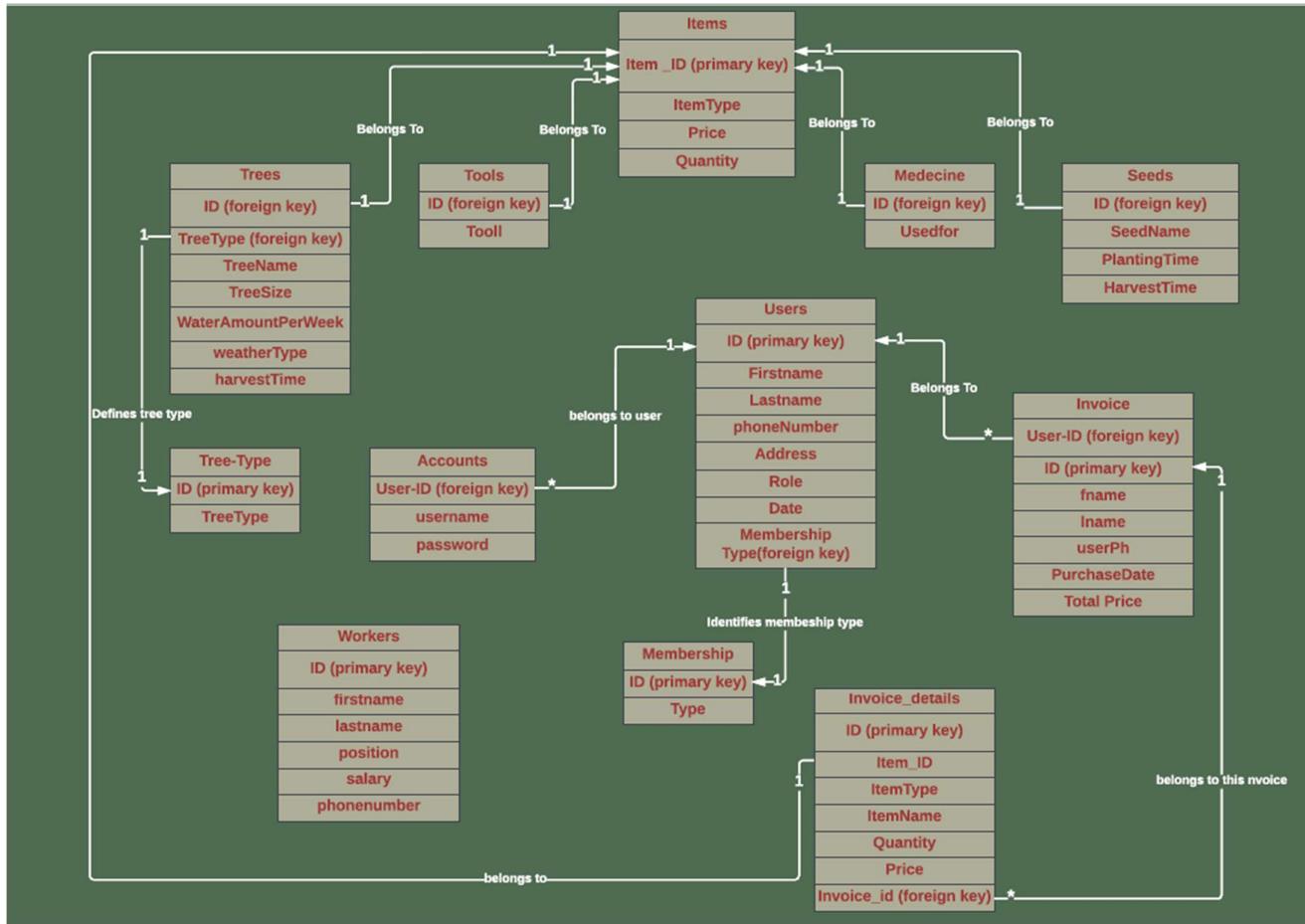


Figure 29: Conceptual diagram

This conceptual diagram shows the relations between the database tables where:

- Every item belongs (exists) to the items table once.
- Every user may have several invoices and 1 or more account.
- Every invoice may contain 1 or more items included inside it.
- Every user can select only 1 type of membership subscription.
- Every tree can be of one tree-type (fruit tree, privacy tree, evergreen tree, or shrub tree).

## CUSTOMER USE-CASE DIAGRAM

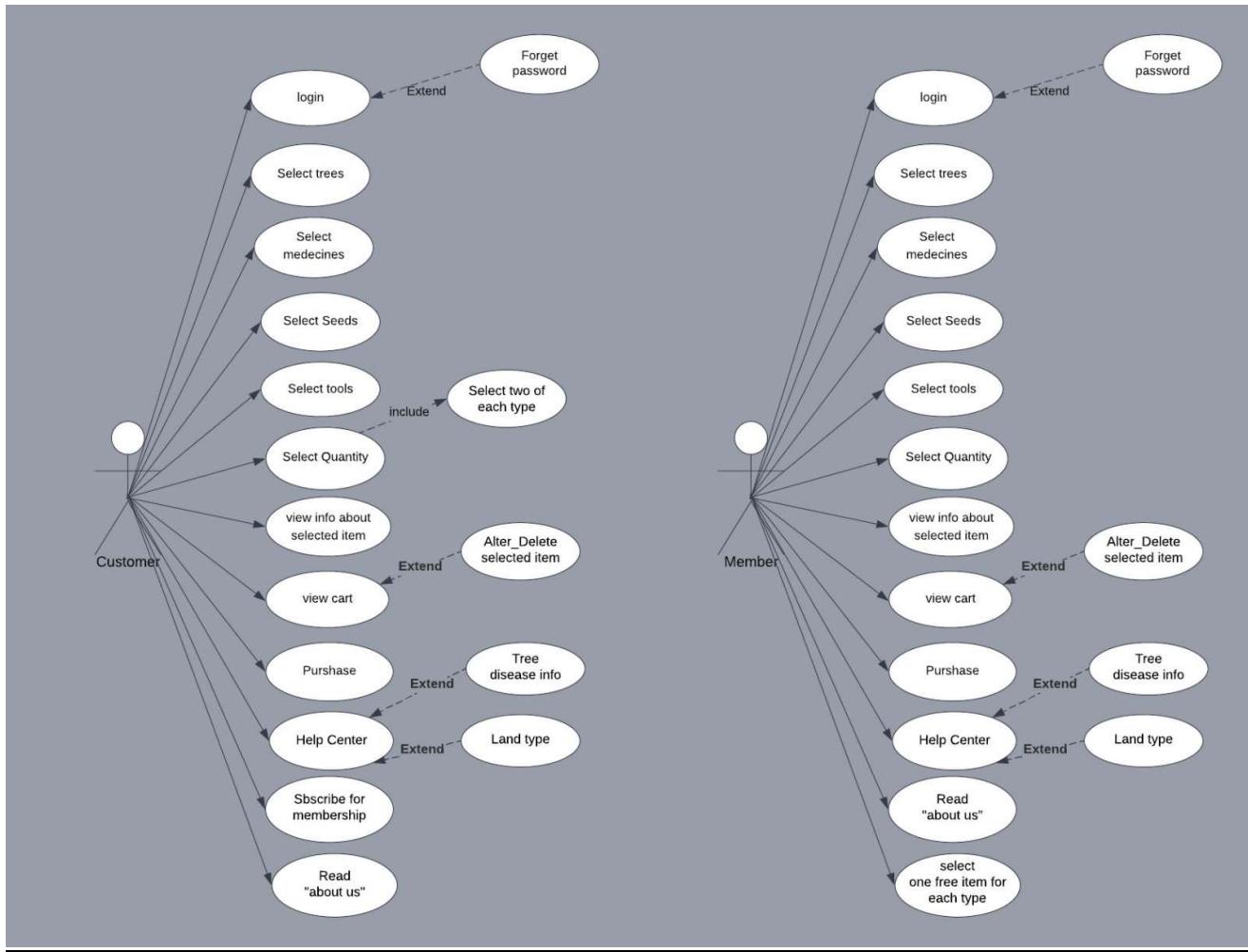


Figure 30: Customer use-case diagram

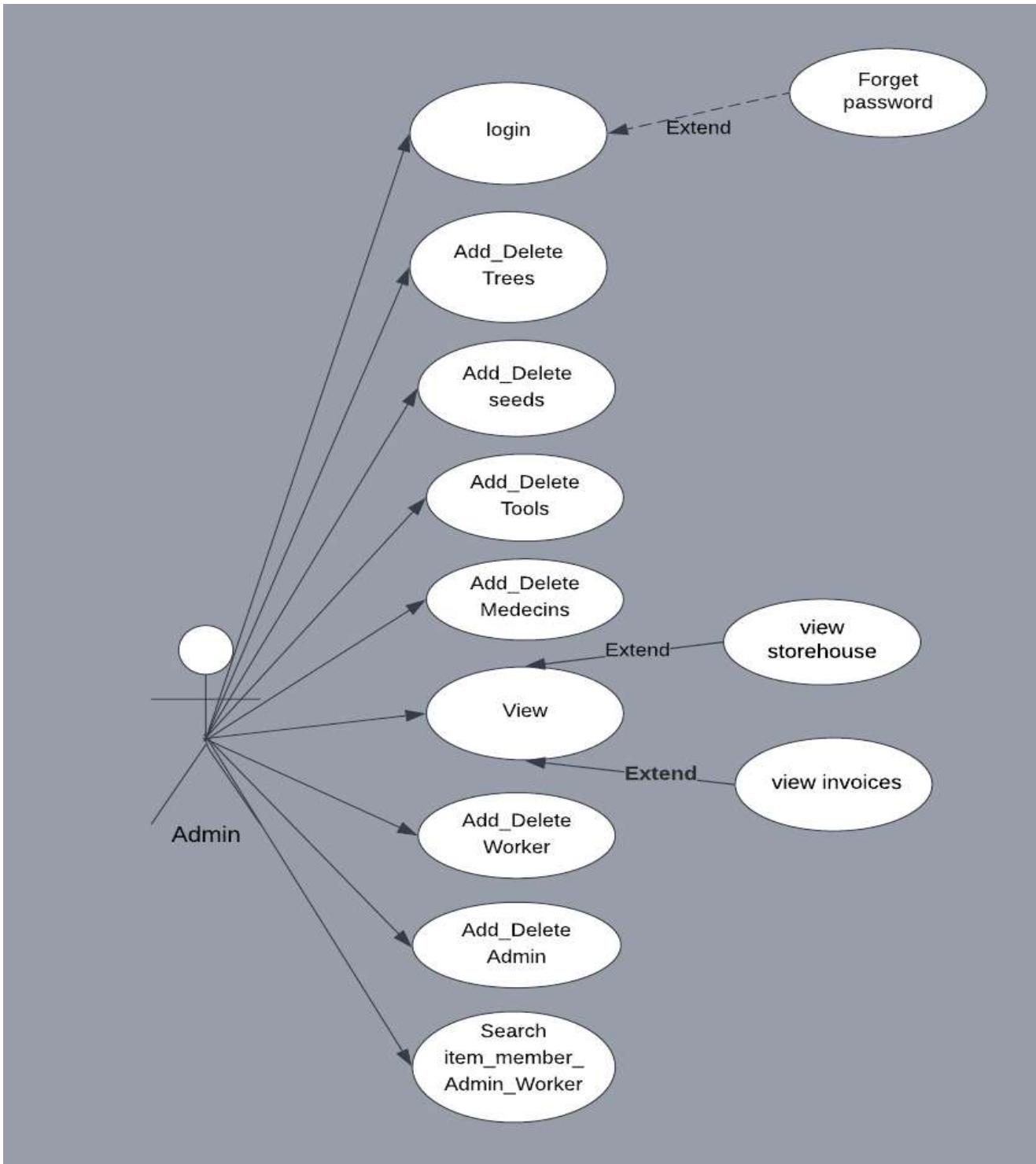
Here in this Use-case we can notice that both customer and member can:

- Login.
- Select: trees/tools/medicines/seeds.
- Select quantity (**max quantity = 2 for customers**) / (**freely for members**).
- View info about any selected item in the cart.
- View and alter (**delete or change quantity**) cart.
- Purchase
- Go to help center that includes both **tree disease info** and **land types**
- Read the **about us** article.

Furthermore, **customers** can: subscribe to membership.

**Members** can: select one free item each month of subscription.

## ADMIN USE-CASE DIAGRAM



Here we can notice that admin can:

- Add or delete: trees/tools/medicines/seeds/workers/co-admins.
- View: invoices purchased by customers/storehouse.
- Login.

## RELATIONAL DATABASE DIAGRAM



Figure 31: Relational Database

This relational database includes:

- 1) **12 tables:**
- 2) Tables (**Trees**, **Tools**, **Medecine**, **Seeds**, **Invoice-details**) are all connected to the **Items** table.
- 3) Table **Tree-Type** is connected to **Trees** table.
- 4) Table **Users** is connected to table **membership**.
- 5) Tables **Accounts** and **Invoice** are connected to table **Users**.
- 6) Table **Invoice-details** is connected to table **Invoice**.

## CONCLUSION AND PERSPECTIVE

During the implementation of this program, I faced a lot of problems that are very important and unnoticeable, some of these problems:

- When customers buy a specific product and add it to the cart, and then go back to the shop and buy the same product and adding it again to the cart, the cart displays 2 rows of this same product instead of keeping this product in one row but changing the quantity instead.
- Changing the product total price in the cart immediately after altering the quantity in the cart.
- Allowing the customer to select only a maximum quantity = 2 for any product.
- The member was able to select more than 1 free item.
- Finding suitable pictures and backgrounds also took a very long time and a lot of researches.

And many other **DETAILED** problems. Thankfully all these problems have been solved carefully to make this application as efficient and precise as possible.

If I had the chance and time to make this application better, I would like to do some improvements such as:

- Adding some animations to make the application more attractive and modern looking.
- Expanding the shop's scalability in order to add more products.
- Activating the shipping and purchasing services and make it a real application so anyone can really buy products online.
- Changing the design and make it a little bit more attractive and comfortable.

## BIBLIOGRAPHY

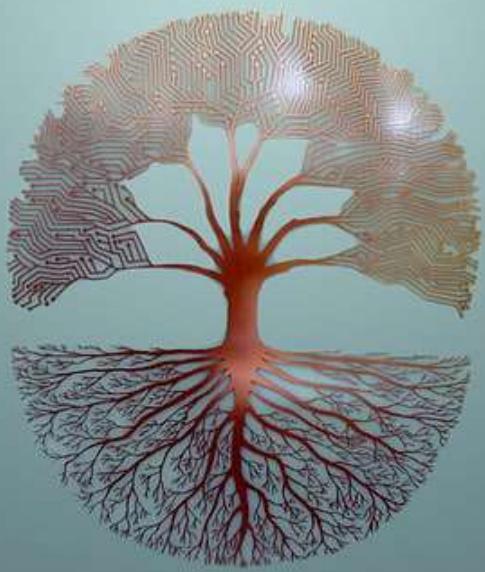
Sources helped in fixing problems that faced this application:



Stack Overflow is a question-and-answer website for professional and enthusiast programmers. It is the flagship site of the Stack Exchange Network. It was created in 2008 by Jeff Atwood and Joel Spolsky. It features questions and answers on a wide range of topics in computer programming.

<https://stackoverflow.com/>

- In addition to some YouTube videos from different channels.



GREENLAND