

C868 – Software Capstone Project Summary

Task 2 – Section C



Capstone Proposal Project Name: _____ SapbaShop _____

Student Name: _____ Christopher Smith _____

Table of Contents

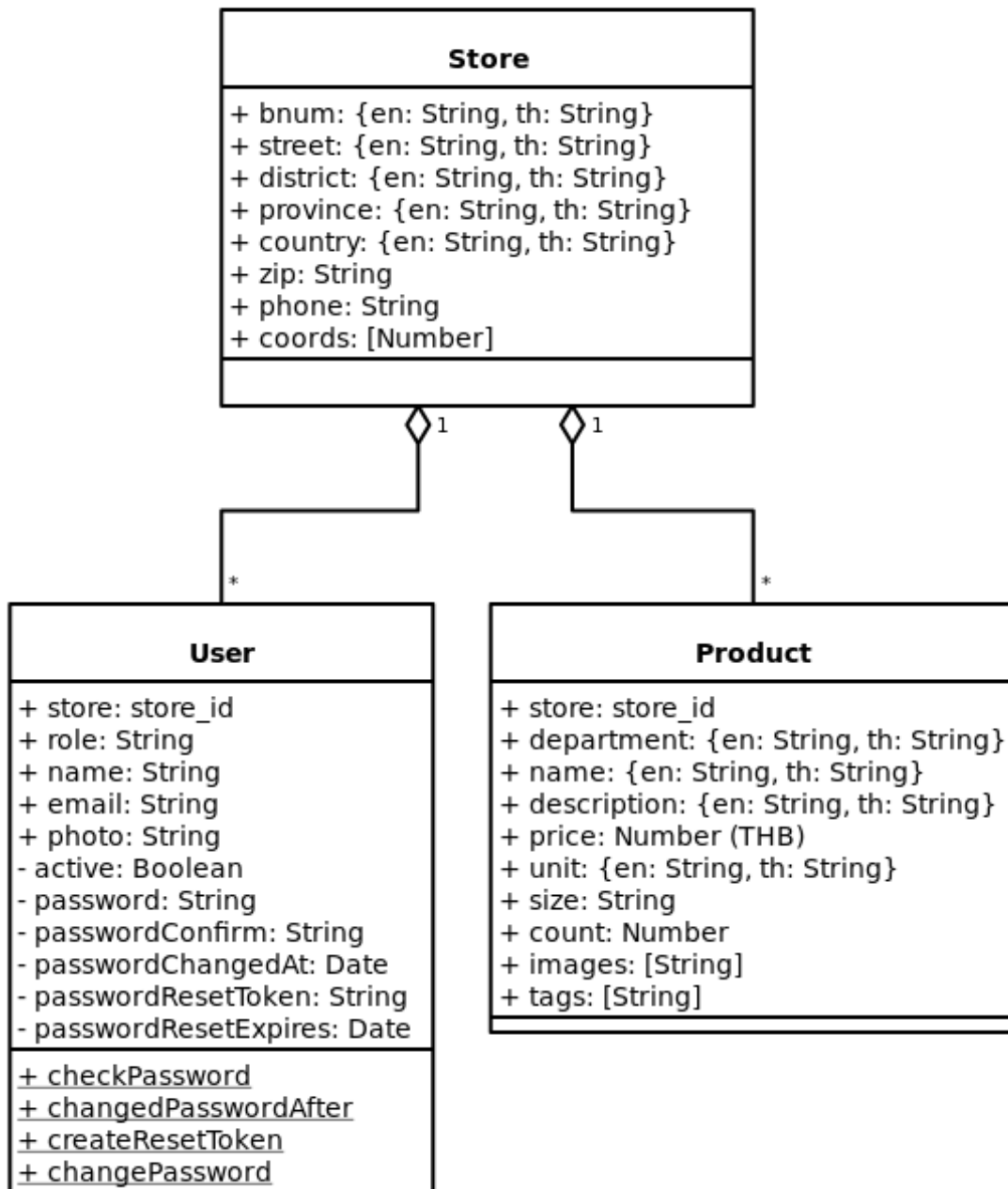
Application Design and Testing	3
Class Design	3
UI Design.....	4
Unit Test Plan	8
Product Model – Black box Testing.....	8
Product Model – White box Testing	8
Source Code	9
User Guide	10
Viewing the application	10
Logging In for the first time.....	10
Modifying your account	11
Account Info	11
Change Password	11
Account Settings	11
Navigation	12
The Dashboard	13
Stores	14
Adding a Store	15
Modifying a Store	16
Products.....	17
Adding a Product.....	18
Modifying a Product.....	19
Users	20
Adding a User	21
Modifying a User	22
Logging Out.....	23
Troubleshooting	24
Can't find the home page?	24
Logging in, Failed to Fetch?.....	24
Help! Admin deleted or deactivated their account!	24

Application Design and Testing

Class Design

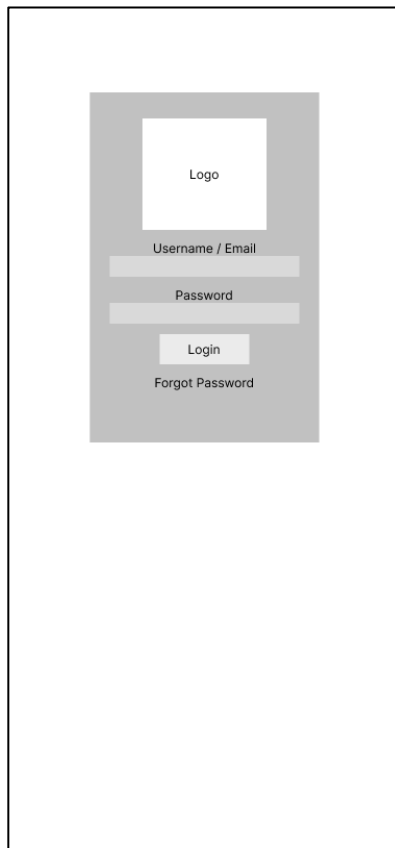
The following is the class diagram for the application's three models: Store, User, and Product. The User and Product model has 1 to many associations with the Store model, but only the product has a composite association.

cls Detailed Class Design



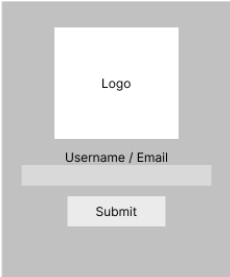
UI Design

The initial design was a low-fidelity desktop wireframe. Once informed the client uses various sized devices, we redesigned it to a mobile-first design. Below are the mobile wireframes. Not all pages have wireframes, as the plan was to reuse them for the other model's lists and forms. Again, the main focus during this project was speed and functionality to deliver a working product to the client as they wish to expand their footprint.



Login

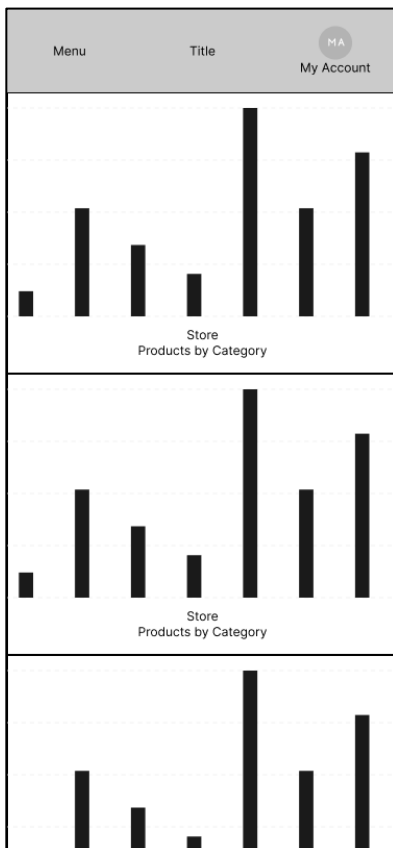
Simple login page to enter the app. Logo at the top. Form inputs for email and password. Then login button to submit the form and a link for users who forgot their password.



A login form for the 'Forgot Password' feature. It consists of a light gray rounded rectangle containing a white square placeholder for a logo at the top. Below the logo is a text label 'Username / Email' followed by a light gray input field. At the bottom of the form is a light gray button labeled 'Submit'.

Forgot Password

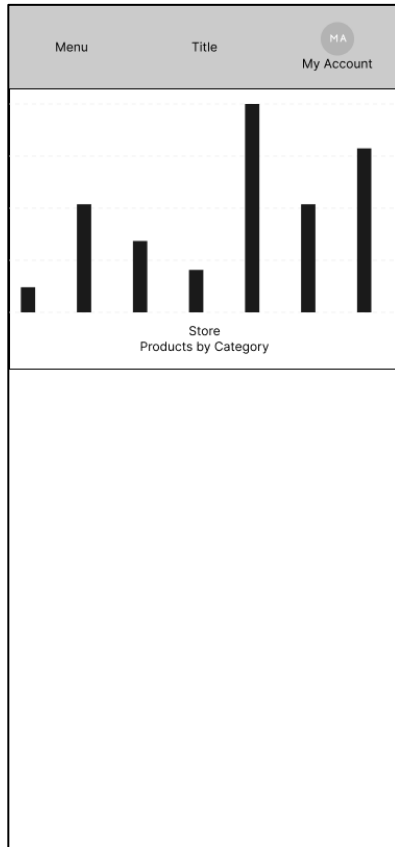
Design displays a Logo for the app with an input field for the user's email and a button to submit the form.



Admin Dashboard

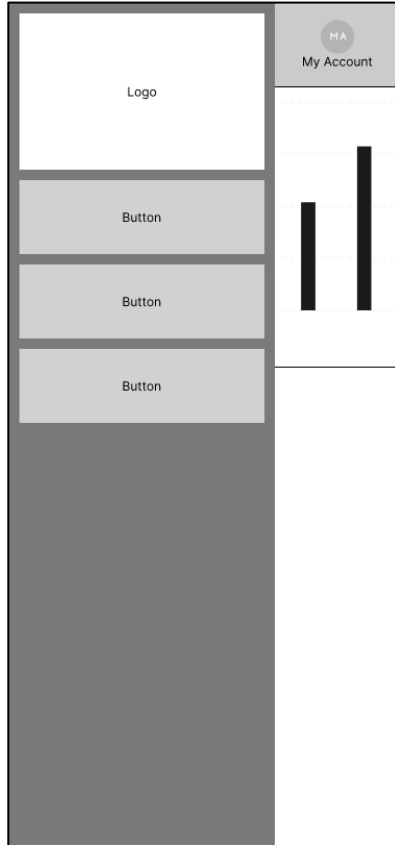
The page shows an upfront report of inventory levels across all stores.

The header, placed at the top on all views, displays the menu button to the top left, the title in the middle, and the user's account image at the top right.



Manager Dashboard

The page provides an upfront report of inventory levels for the manager's store.



Side Navigation Menu

The menu opens from the left side with button options for the other pages.

SapbaShop – “A Shop for Everything”

Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count
Department	Product name	Count

Add Product

Item List View

The view displays a list of products. It includes options to search and add a new one. It will be reused for the other models.

Item

item field

item field

item field

item field

item field

Delete Cancel Save

Item Form View

Form view for any of the models. The selected Item name is placed at the top, followed by the item image and the associated fields. At the bottom are buttons to delete, cancel or save the form.

Unit Test Plan

This section will describe the unit test plan for one of the models to show at a basic level how our code is validated to ensure quality results. To conduct this test, we will use Jest, a testing framework. The model for our example will be the Product Model.

Product Model – Black box Testing

The first tests will use a Black box methodology, meaning that we are testing as if we don't know how the code for the Product Model works. According to the design, we see a Product must be assigned a store. In the first test, we will give the product a store along with the rest of the required fields and expect it to pass. In the second test, we will omit the store field and label the test as failing, as we expect the test to fail without an assigned store.

```
16
17 describe('Product Model Tests', () => {
18   // Create a Product
19   it('Should create a product', async () => {
20     const product = await Product.create(productTest);
21     expect(product).toBeInstanceOf(Product);
22     ptm.checkProperties(product, productTest);
23   });
24
25   // Store - Req
26   describe('Testing Store field', () => {
27     it.failing('Should require a store', async () => {
28       const { store, ...product } = productTest;
29       await Product.create(product);
30     });
31   });
32 }
```

Now when executing the tests, we will see that both tests pass. The attempt to create a new Product without having an assigned store failed as expected, so it passed the test.

```
PASS tests/products/productModel.test.js
  Product Model Tests
    ✓ Should create a product (7 ms)
    Testing Store field
      ✓ Should require a store (2 ms)
```

Product Model – White box Testing

The subsequent tests will utilize the White box testing methodology, the opposite of Blackbox, meaning that we know the inner workings of the code and expect it to perform a certain way. This time, we will recheck it for the error message to ensure the user gets a clear, actionable message.

```
31   it('Required Error message should be', async () => {
32     const { store, ...product } = productTest;
33     await Product.create(product).catch((err) => {
34       expect(err.message).toBe(
35         'Product validation failed: store: A product must belong to a store.'
36       );
37     });
38   });
39 }
```


Further tests will all be using White Box testing. We will test with an invalid Store ID, a valid ID that does not match any store in the test database, and the error message associated with these tests.

```
39   it.failing('Store should throw invalid id', async () => {
40     productTest.store = '62d5da789de84f9f98a126';
41     await Product.create(productTest);
42   });
43   it.failing('Should throw error if id does not match a store', async () => {
44     productTest.store = '62d7e3a12d87f39787295a53'; // a user id
45     await Product.create(productTest);
46   });
47   it('Invalid Store error message should be', async () => {
48     productTest.store = '62d7e3a12d87f39787295a53'; // a user id
49     await Product.create(productTest).catch((err) => {
50       expect(err.message).toBe(
51         '62d7e3a12d87f39787295a53 is not a valid store.'
52       );
53     });
54   });
55 });
```

Now rerun the tests, all together. All tests should pass.

```
PASS tests/products/productModel.test.js
Product Model Tests
  ✓ Should create a product (7 ms)
  Testing Store field
    ✓ Should require a store (2 ms)
    ✓ Required Error message should be (2 ms)
    ✓ Store should throw invalid id (3 ms)
    ✓ Should throw error if id does not match a store (3 ms)
    ✓ Invalid Store error message should be (1 ms)
```

These tests are just a tiny snippet of the Product Model test suite. Further checks ensure all fields work as expected for the Product Model and the rest of the models.

Source Code

Source Code will be included as a compressed file. The source code's root directory will contain a README.md with installation instructions. The User Guide in the next section will show how to navigate and use the application.

User Guide

This guide covers how to use the application after installation. For information on installing the application, please follow the README.md that is in the root directory. Installation instructions can also be found with the source code on [GitHub](#).

Viewing the application

If you haven't already, use "npm start" to start the application. After the server is running the application, go to the domain you configured it to in your web browser. Once you reach the login screen, please continue to the next step.

Next Steps:

- [Logging In for the first time](#)

Troubleshooting:

- [Can't find the home page?](#)

Logging In for the first time

During the first time running the app with an empty database. The app initializes with a new admin user. Use the following credentials to log in:

Email: admin@sapbashop.com

Password: pass1234

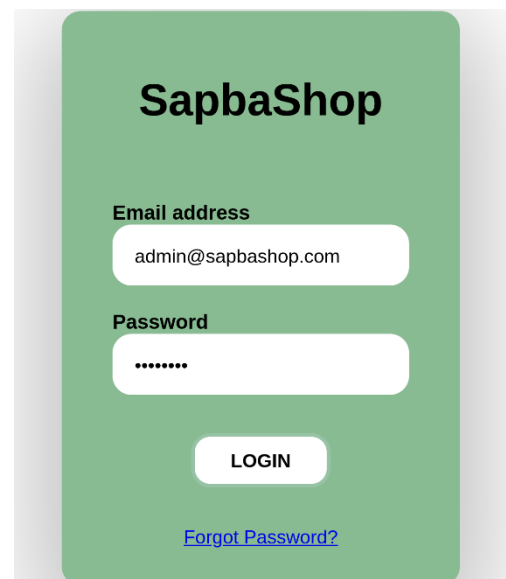
Once logged in, you will see the Admin Dashboard. Before adding a store, we recommend you update the account info and change your password.

Next Steps:

- [Modifying your account](#)
- [Navigation](#)
- [Adding a Store](#)

Troubleshooting:

- [Logging in, Failed to Fetch?](#)



SapbaShop

Email address
admin@sapbashop.com

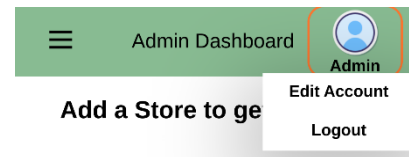
Password
.....

LOGIN

[Forgot Password?](#)

Modifying your account

To modify your account, select your profile icon in the header at the top right-hand corner. A drop-down menu should appear, and then select "Edit Account."



 The image shows the "Account Info" form. It has three tabs: "Info", "Password", and "Settings". The "Info" tab is active. It contains a profile icon, a "Store" dropdown menu with the text "Choose a Store", a "Name" text input field with the value "admin", and an "Email" text input field with the value "admin@sapbashop.com". At the bottom is a green "UPDATE" button.

Account Info

Upon entering "Edit Account," you should automatically see your account info.

The store field will display the store associated with the account. Once a store has been added, a drop-down option will appear to change your store. This field is not required.

Fill out the rest as necessary.

When ready, select Update to change account info.

Select the Password tab on the "Edit Account" Page to change your password.

Then enter your current password, followed by your new password and the confirmation, and press update.

 The image shows the "Change Password" form. It has three tabs: "Info", "Password", and "Settings". The "Password" tab is active. It contains a "Current Password" text input field with masked characters, an "Update Password" section with a "New Password" text input field and a "Password Confirmation" text input field, both with masked characters. At the bottom is a green "UPDATE" button.

Change Password

 The image shows the "Account Settings" form. It has three tabs: "Info", "Password", and "Settings". The "Settings" tab is active. It contains a "Delete Account" button with a red "DELETE" label.

Account Settings

Under Settings, users can delete their accounts if they wish. This will place the account into a deactivated state for the Admin to delete their account.

Later to be implemented will be an option to automatically cleanup deactivated users during off hours. The deactivation function was to prevent accidental deletion and will require an admin to reactive account if necessary.

Next Steps:

- [Navigation](#)
- [Adding a Store](#)

Troubleshooting:

- [Help! Admin deleted or deactivated their account!](#)

Navigation

The menu navigation is available on every page after login. To open the menu navigation, select the three horizontal bars (aka hamburger icon) in the header bar, located in the upper left-hand corner of the page. Once pressed, a menu should open from the left with options to navigate to other pages



Getting Started:

We will navigate to the Stores page to add our first store.

Next Steps:

- [The Dashboard](#)
- [Stores](#)
- [Products](#)
- [Users](#)

The Dashboard

The Dashboard is the first page seen after logging in. The page differs depending on the user's role as it tailors to their scope of responsibilities. Once the application is populated with stores and products, it will show current inventory reports as bar charts.



The Administrator will see reports for all stores.

The Manager and Employee will only see a report for their store.

Next Steps:

- [Stores](#)
- [Products](#)
- [Users](#)

Stores

The Stores page displays a grid of all stores in the database. You can select a store from this page to view the details and modify them if you have proper permissions. See [Modifying a Store](#) for more information.

As an Administrator, you can also add a store by pressing the plus icon on the upper-left-hand side of the section.



Getting Started:

Since we do not have any stores, we will add one since it is a prerequisite for all other actions.

Next Steps:

- [Adding a Store](#)
- [Modifying a Store](#)

Adding a Store

This page is only available to administrators.

The form is intuitive. Each field has inputs for both languages as the shop has international customers. Language checking is not in place in the chance that translations are not available. The only exception to the dual language requirement is the numeric fields. Only English numbers are accepted. For the phone number field, please use the local format. Press save when complete.

For any reason, if you wish to cancel, press the X icon button in the top left of the section to return to Stores.

SapbaShop Stores Admin

New Store Form

Building Number / Village

English: 123 Moo 10

Thai: 123 หมู่ 10

Street

English: Chiang Mai-Hangdong Rd

Thai: ถ.เชียงใหม่-หางดง

District

English: Hang Dong

Thai: หางดง

Province

English:

Getting Started:

Now once you have created a Store, you can start adding your products from the products page

Next Steps:

- [Products](#)
- [Adding a Product](#)

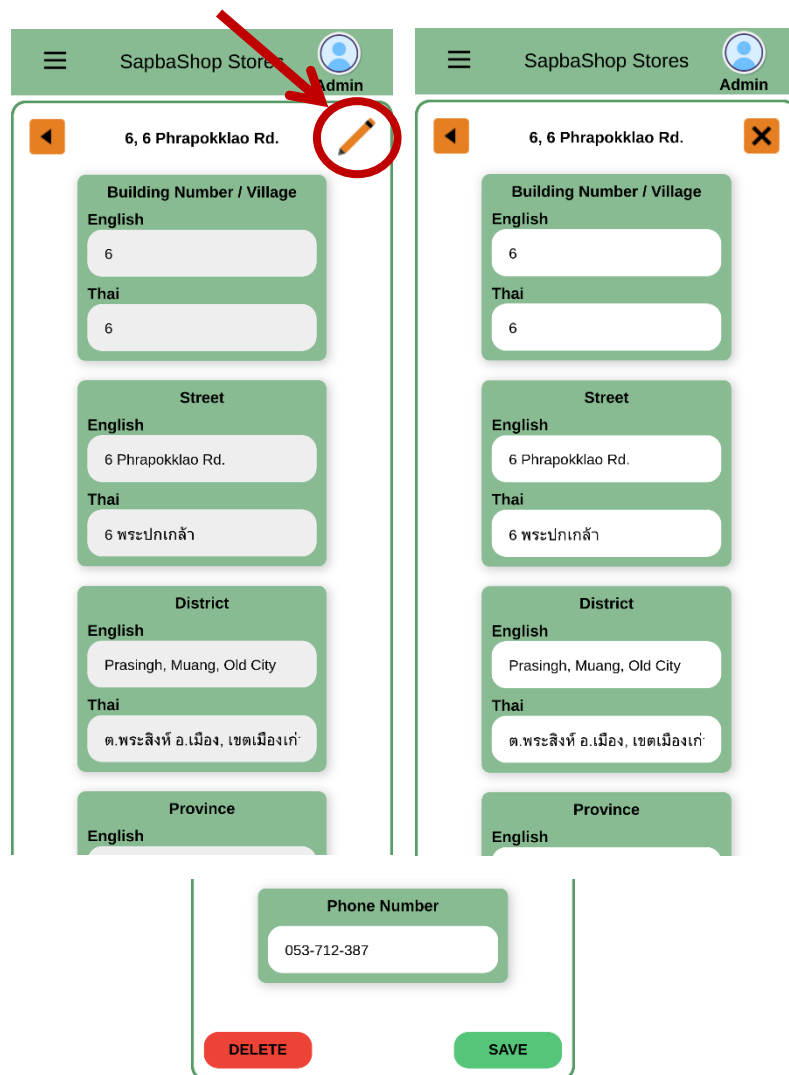
Modifying a Store

Modifying a store is only available to an Administrator or Manager.

Once viewing a store's details, you can modify the store if you see the pencil in the upper right-hand corner of the section. When editing, the fields will unlock, and the pencil will become an X icon, allowing you to cancel editing. When complete, press the save button at the bottom to complete the changes.

As an administrator, you can delete a store once in the edit view by scrolling to the bottom and pressing the delete button.

WARNING: Deleting a store also deletes all products associated with this store. However, it does not delete users.



The image displays two side-by-side screenshots of the SapbaShop Stores app interface, illustrating the process of modifying a store.

Left Screenshot (Edit View): The top bar shows "SapbaShop Stores" and "Admin". Below the header, the store name "6, 6 Phrapokklao Rd." is displayed. A red circle highlights a pencil icon in the upper right-hand corner of the form section, indicating the edit function. The form contains several input fields for store details, each with English and Thai labels:

- Building Number / Village:** English (6), Thai (6)
- Street:** English (6 Phrapokklao Rd.), Thai (6 พระปกเกล้า)
- District:** English (Prasingh, Muang, Old City), Thai (ต.พระสิงห์ อ.เมือง, เขตเมืองเก่า)
- Province:** English ()

Right Screenshot (View View): The top bar shows "SapbaShop Stores" and "Admin". The store name "6, 6 Phrapokklao Rd." is displayed. An 'X' icon is visible in the upper right-hand corner of the form section, indicating the cancel function. The form contains the same input fields as the left screenshot.

Bottom Section: A "Phone Number" field is shown with the value "053-712-387". Below this are two buttons: "DELETE" (red) and "SAVE" (green).

Products

The products page contains a grid of all products across all stores and displays the English and Thai name for each product. The main feature of the products page is the search function. First, you can search by name or department. Then, once you find your product, select it to view more details or modify it.

For more technical users, behind the scenes, it uses a text index on the name and department of each product. In addition, the product name is broken into ordered combinations of letters, allowing for a partial word search.

Administrators and Managers will have the option to add a new product. To add a new product, press the plus icon in the upper left-hand corner of the section.



Getting Started:

To add our first product, select the plus icon on the upper left corner of the section.

Next Steps:

- [Adding a Product](#)
- [Modifying a Product](#)

Adding a Product

This feature is only available to Administrators and Managers.

All fields, like the store form, are dual language fields, except the store section at the bottom of the page. The store section of the product form is where you assign the quantity, price, and size (if applicable) to a store. To add the product to an additional store, press the plus icon at the top right corner of the section. The red tag with an X allows you to remove the store row if needed. Finally, press the save button to complete adding a product.

New Product Form

Product Name

English: Broccoli Crowns

Thai: บล๊อคโคโล่

Product Description

English: Local grown, edible green vegetable.

Thai: ปลูกในประเทศไทย

Product Department

- Grocery
- ร้านค้า
- Clothing and Accessories
- เสื้อผ้าและเครื่องประดับ

Product Department

- Grocery
- ร้านค้า
- Clothing and Accessories
- เสื้อผ้าและเครื่องประดับ

Unit

English: ea

Thai: ชิ้น

Store	Size	฿	Qty
6, 6 Phrapokk	N	00	00

SAVE

2022 © Chris Smith
App icons by [icons8](#)

Getting Started:

Next, we will add our employees from the Users page.

Next Steps:

- [Users](#)
- [Adding a User](#)

Modifying a Product

Modifying an existing product's details is available to all employees.

When viewing a product, press the pencil icon in the upper right-hand corner of the section to unlock the fields and start editing. If you wish to cancel editing, you can press the X icon that replaced the pencil. The form is all dual language except for the store section.

Only administrators and Managers can add or delete a store row from here. To add the product to an additional store, press the plus icon at the top right corner of the section. The red tag with an X allows you to remove the store row if needed. Finally, press the save button to complete adding a product.

At the bottom, you can delete the product at all the stores with the delete button.



Users

The users' page displays all users in a table format. Information is limited to the size of the screen—the larger the screen, the more information is displayed. At the largest, four columns will display the user's role, name, email, and store. The store column can be empty if they are not assigned a store.

Users have four roles: Administrator, Manager, Employee, and Customer. Of course, the management app does not authorize Customers. However, there are plans to be implemented later with a customer-facing site.

Users that are greyed out and have an (X) by their role are deactivated users.

Select a row to view the user's details.

Only Administrators will have the option to add and modify users.

Getting Started:

Select the plus icon in the upper left to add an employee or manager and continue to the "Adding a User" page.

Next Steps:

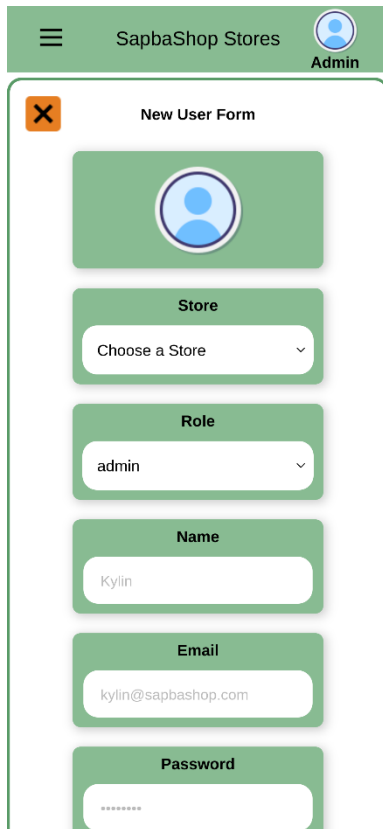
- [Adding a User](#)
- [Modifying a User](#)

Role	Name
Admin	Admin
Customer	Customer
Employee	Employee
Manager	Manager
Customer (X)	Customertwo

Adding a User

This feature is only available to Administrators.

The new user form is straightforward, and a store assignment is not required. Give the user a role, name, email, and password with confirmation and save. Of course, you will have to give the user the password to log in, but they can change it later.



The screenshot shows the SapbaShop Admin interface. At the top, there is a header bar with a hamburger menu icon, the text 'SapbaShop Stores', and a user profile icon labeled 'Admin'. Below the header, there is a 'New User Form' modal. The form has a title bar with a close button (X) and the text 'New User Form'. The form contains several input fields: a 'Store' dropdown menu with the text 'Choose a Store', a 'Role' dropdown menu with the text 'admin', a 'Name' text input field with the text 'Kylin', an 'Email' text input field with the text 'kylin@sapbashop.com', and a 'Password' text input field with masked characters '*****'.

Getting Started:

That was the basics. Now that you've added a store, product, and user, you can visit the Dashboard to see your current inventory report. That's the end of the tour. If you have any questions, please reach out to our support team.

Next Steps:

- [The Dashboard](#)
- [Logging Out](#)

Modifying a User

This feature is only available to Administrators.

If you need to modify an account, the option is here. Then, like the rest, start editing by selecting the pencil in the upper right-hand corner.

You may notice that an additional field appears at the bottom of the form, the status field. The status switch allows the Administrator to deactivate an account without completely deleting it. Or can reactivate it in the chance that someone deactivates their account.

At the bottom is the choice of deleting the user permanently or saving the modifications.

The image displays two side-by-side screenshots of the SapbaShop user management interface, specifically the 'Employee' form.

Left Screenshot (Editing Mode): The header bar shows 'SapbaShop Users' and 'Admin'. The form title is 'Employee' with a pencil icon in the top right corner. The form contains the following fields:

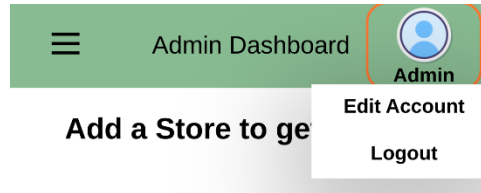
- Active:** A green button with a white circle and a blue person icon.
- Store:** A dropdown menu showing '51, Mae Rim-Samoeng Old I'.
- Role:** A dropdown menu showing 'employee'.
- Name:** A text input field containing 'employee'.
- Email:** A text input field containing 'employee@sapbashop.com'.

Right Screenshot (View Mode): The header bar shows 'Employee' and a close icon (X) in the top right corner. The form contains the following fields:

- Active:** A green button with a white circle and a blue person icon.
- Store:** A dropdown menu showing '51, Mae Rim-Samoeng Old I'.
- Role:** A dropdown menu showing 'employee'.
- Name:** A text input field containing 'employee'.
- Email:** A text input field containing 'employee@sapbashop.com'.
- Status:** A toggle switch that is currently turned on (green).

Logging Out

To log out of the app, select your account icon in the header at the upper right-hand corner of the page, and a drop-down will appear. Select logout, and you will see the login page after successfully logging out.



Troubleshooting

Can't find the home page?

If you are testing the application and using the same machine as the server, you will most likely find the home page at <http://localhost:8000> or <https://localhost:8000> (if you added TLS/SSL certs). If you configured a different port, replace 8000 with your configured port.

The server console will display the protocol and port used.

If you are not using the same machine and do not have DNS configured with a domain name for the server, then you will need to use the IP address of your server.

If you are still having issues connecting, please check your firewall settings.

Logging in, Failed to Fetch?

Please verify in your ".env" file that the protocol, domain name, and port in the `SAPBA_API_URL` variable match.

Additionally, please note that the app currently does not support and has not tested the use behind a reverse proxy. Reverse proxy support will be in future releases.

Help! Admin deleted or deactivated their account!

There are only two ways to fix this issue

1. Another admin restores the deactivated admin account or creates a new account.
2. If you do not wish to delete the database and there is no other Admin. You will need to manually make an entry in the database to restore the deactivated admin account or create a new admin account.