



CP2106 (Orbital)

Peh Hoe Khim Marcus
Ee Kar Hee, Nicholas

Team Name:

Markeet

Level of Achievement:

Artemis

Year:

2021

TABLE OF CONTENTS

Poster	3
Code Base	4
Deployment	4
Motivation	4
Aim	4
User Stories	5
Features and Timeline	6
Core features:	6
Features to be completed by Milestone 1 (31 May):	6
Features to be completed by Milestone 2 (28 Jun):	6
Features to be completed by Milestone 3 (26 Jul):	6
Possible future feature implementations:	7
Outline	7
Timeline	7
Wire Frame	8
Tech Stack	8
Comparison with other apps	8
Shopify POS (https://www.shopify.com.sg/pos)	8
Testing	9
Summary of test	9

Poster

SALES &
INVENTORY
MANAGEMENT AT
YOUR
FINGERTIPS!

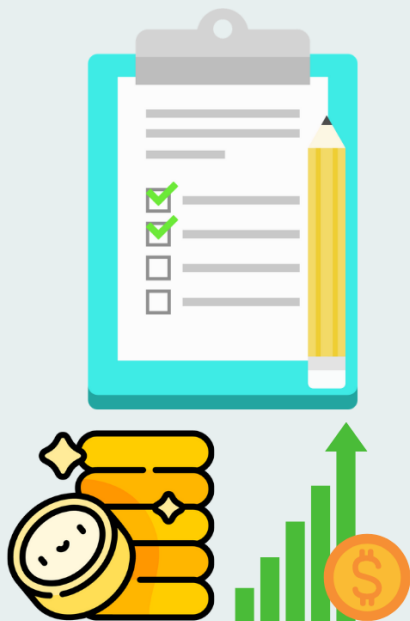
- ✓ Sales Report
- ✓ Inventory Tracking
- ✓ Cashier Interface
- ✓ Telegram Bot Integration
- ✓ No ads

ABOUT US

Free-to-use, web-based Point of Sale system with real-time detailed sales report and inventory count.

Telegram integration allows customers to easily check store locations and stock inventory.

markeet



Developed with



Scan here to find out more

Orbital'21 Markeet
Marcus & Nicholas

Code Base

<https://github.com/marcuspeh/Markeet>

Deployment

<https://markeet.herokuapp.com/>

Motivation

Ever run a small shop on Carousell or a flea market store but have no idea how to keep track of inventory and profit? Or perhaps you want to find out easily how much profit you are making? Enter Markeet, a free Point Of Sale (POS) system that can help you solve the issue.

Or are you a customer, traveling down to a shop to buy a certain item, only to find out the item is out of stock. If the shop uses Markeet, time spent travelling to the shop can be saved too. You can easily check the stock of the item on your phone and even find the nearest shop that sells the item.

Aim

The aim of the project is to benefit small retail shops by creating an open source and free to use Point Of Sales (POS) system. Many small businesses currently use traditional methods to keep track of sales and conduct stock-taking, be it due to budget constraints or the small scale of the business. As such, there is value in implementing a POS system that can be used easily anywhere at almost no cost. All that's needed is an internet connection and a device to access the internet.

Customers can also check inventory count for the item, instead of wasting a trip down to the shop only to realise that it is out of stock. They can also locate the nearest shop that has stocks for the item they are looking to buy.

User Stories

1. As a stock taker for a shop, I want to be able to keep track of the stock easily without having to count it manually and be alerted when I need to restock.
2. As a small business owner, I want to be able to get data from the monthly sales that can help my business grow, for example profits and best selling items.
3. As a business owner with multiple outlets, I want to be able to find out which shops are making a profit or loss.
4. As a cashier, I want to have a system that is easy to use and convenient.
5. As an accountant, I want to be able to manage the finances of the company easily, instead of having to go through books and papers to find the information needed.
6. As a customer, I want to be able to find out if the item I want to buy is available before I make my way down to the shop to purchase it. I also want to be able to locate the nearest shop that sells the item I'm looking for

Features and Timeline

A **Web-based Point Of Sale (POS) system** that is free to use. It will help businesses generate real-time detailed sales reports and inventory count, on top of normal cashier operations.

The **Telegram Bot** provides a quick way for customers to check for stocks in a particular store and locate the nearest shop with the item they want to buy. The Telegram Bot will share the same database as the POS system

Core features:

- (1) **Dashboard page** - The main page when a user logs in to the website. They will be able to see details on their shop, like inventory count, latest sales and other notifications. Links to other pages will also be available on that page.
- (2) **Cashier page**- Where cashiers can enter sales in
- (3) **Inventory page** - For live tracking of inventory. Options to add in inventory count and remove inventory will be available too. Updating inventory by CSV will also be supported.
- (4) **Sales report page** - To see the profit/loss of the shop as well as the best/worst selling item. Other details regarding such as a breakdown of sales per day etc will also be shown.
- (5) **Telegram Bot** - For customers to find out if the item is in stock. They can also locate the nearest shop with the items they are looking to buy.

Features to be completed by Milestone 1 (31 May):

- Design of the system
- Integrated frontend and backend system with a login feature
- (1) "Settings" page for changing settings
- (2) Barebones cashier page
- (3) Inventory list displaying all the inventory in the database
- (3) Ability to add in new stocks and sending it to the database

Features to be completed by Milestone 2 (28 Jun):

- (1) Display information about the shop
- (2) Entering of sales
- (3) Sorting of inventory list based on different method
- (3) Deduction of stock based on sales
- (3) Alert when stocks are running low
- (4) Ability to see basic sales report such as profit/lost
- (4) Sales information will be updated once sales are entered
- (5) Telegram bot basic layout and menu

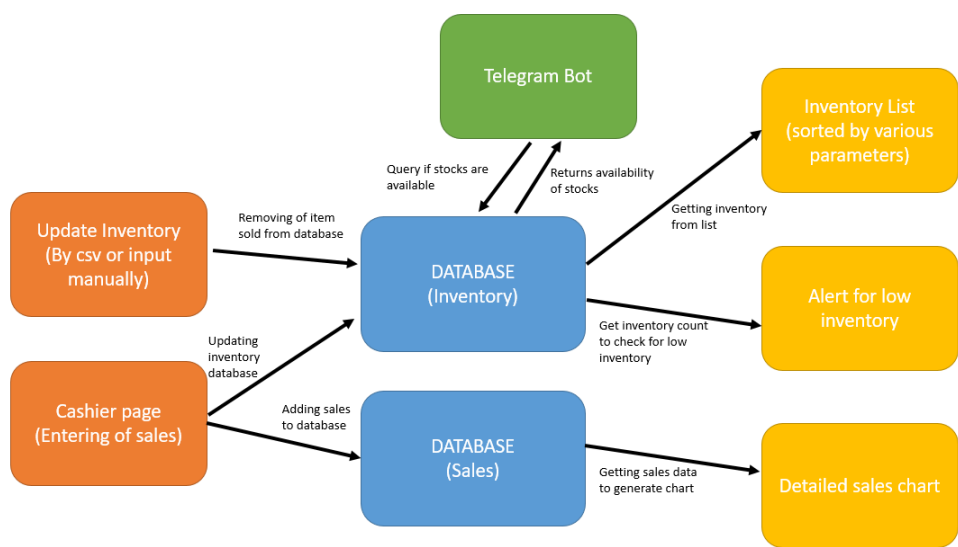
Features to be completed by Milestone 3 (26 Jul):

- (1) Add in options for settings such as Telegram bot support
- (3) Prediction of how much stock to order to replenish
- (4) Charts based on sales
- (5) Telegram bot support for customers to check stock
- (5) Telegram bot support for finding the nearest shop with stock

Possible future feature implementations:

- Machine learning Telegram Bot
- Introductory tutorial page for newcomers
- Support for multiple stores per account
- Employees account for the shop

Outline



Timeline

Dates	To-dos	Done
10 May - 16 May (Lift Off)	Complete ideation and architecture design	<input type="checkbox"/>
17 May - 23 May	Build minimal system with login	<input type="checkbox"/>
24 May - 30 May	Implement inventory	<input type="checkbox"/>
31 May (Milestone 1) - 13 Jun	Implementing of core features	<input type="checkbox"/>
14 Jun - 20 Jun	Complete the basic CRUD features	
21 Jun - 27 Jun	Debugging and basic tests	
28 Jun (Milestone 2) - 11 Jul	Integration of database with Telegram Bot	
12 Jul - 18 Jul	Further testing and debugging	
19 Jul - 25 Jul	Polishing up UI and features	
26 Jul (Milestone 3) - 25 Aug	Refinement	

Wire Frame

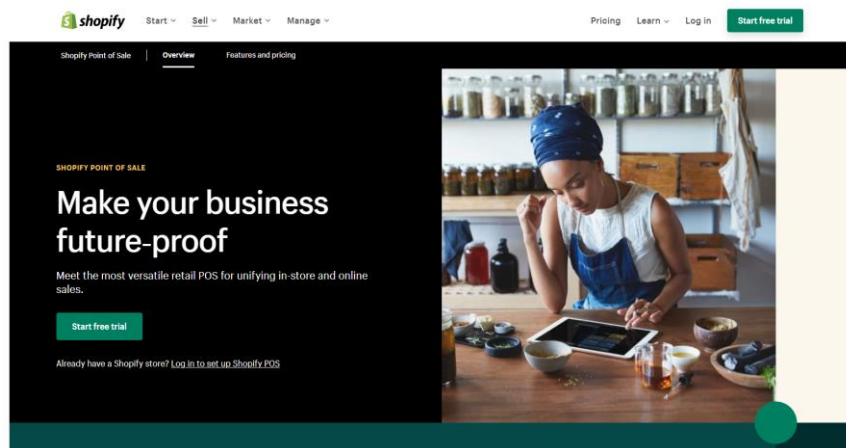
<https://www.figma.com/file/mmQDZh2OWd4JVPEAiMECol/markeet?node-id=0%3A1>

Tech Stack

1. MERN Full stack (for website)
 - a. MongoDB
 - b. Express
 - c. React
 - d. Node JS
2. Redux (for website)
3. Telegram API (for Telegram Bot)
4. Python (for Telegram Bot)
5. Git & GitHub (for version control)
6. Jest (for integration testing)

Comparison with other apps

Shopify POS (<https://www.shopify.com.sg/pos>)



While shopify might be a very big ecommerce website and its POS system is no doubt one of the most user friendly additional services, its POS system is only limited to shopify users. In addition, shopify POS does not support Android devices but only IOS devices. It is also quite expensive to use its full features. Markeet, on the other hand, is completely free to use while enjoying most of the features present in shopify's POS. The most important aspect of markeet is the ability to be used on any devices, regardless of Android, IOS or windows.

Testing

We are using Jest (<https://jestjs.io/>) in order to write unit testing for react.

The testing mainly focus on the following:

- 1. **User input.** We tested if each button is working as expected. For example, clicking on the "inventory" button actually changes the screen to the inventory page instead of others.
- 2. **Entry input.** We tested possible user input to test if the webapp is performing as expected. For example, when registering as a new user, if the user enters an invalid email like "qwerty" instead of "qwerty@domain.com", the website should actually prompt as an error

Summary of test

Feature	Things to test	Test	Expected	Result
Login	Able to register for an account	Opening the "register new user" page	Page should load properly	
		Registering with all empty fields	All the fields should show various error messages. <i>Name:</i> "Name field is required" <i>Email:</i> "Email field is required" <i>Password:</i> "Password field is required"	
		Registering with invalid email eg "qwerty"	Error message will be shown "Email is invalid"	
		Registering with password that is too short	Error message will be shown "Password must be at least 6 characters"	
		Password entered does not match	Error message will be shown "Passwords must match"	
		Registering with an email that is already registered	Error message will be shown "Email already exists"	
		Registering with all valid inputs	Account will be registered	
	Able to log in	Logging in with none	All the fields should	

		of the fields filled in	show various error message <i>Email:</i> "Email field is required" <i>Password:</i> "Password field is required"	
		Logging in with an invalid email eg "qwerty"	Error message will be shown "Email is invalid"	
		Password does not match email	Error message will be shown "Password incorrect"	
		Email is not registered	Error message will be shown "Email not found"	
		Logging in with correct login credentials	Dashboard will be loaded	
Navbar	Navigation bar works as planned	Dashboard button brings the user to dashboard page	Dashboard page loads when clicked	
		Cashier button brings the user to cashier page	Cashier page loads when clicked	
		Inventory button brings the user to inventory page	Inventory page loads when clicked	
		Sales report button brings the user to sales report page	Sales page loads when clicked	
		Logout button sign the user out	Logout button should log the user out and bring them back to the login page	
	Visibility of navigation bar	User is logged in	Navigation bar show up even in landing page	
		User is not logged in	Navigation bar does not show up	
Dashboard	Able to access the website	User is logged in	Dashboard should be shown	
		User is not logged in	Unable to access dashboard and	

			redirected to log in page	
Cashier	Able to access the website	User is logged in	Cashier should be shown	
		User is not logged in	Unable to access Cashier and redirected to log in page	
Inventory	Able to access the website	User is logged in	Inventory should be shown	
		User is not logged in	Unable to access inventory and redirected to log in page	
	Table to show inventory	There is item in the inventory	Details of the item will be shown in the table	
		There is no item in the inventory	A line stating that there is no inventory will be shown	
	Adding products to inventory (create)	Able to add in new inventory by manually filling up the form	A new product will be added to the inventory and shown	
		Clicking add product button or close will close the modal	Modal to add inventory will be closed	
		Able to add inventory by using CSV	All the new products in the CSV are uploaded	
	Editing product (update)	Product details can be shown after clicking on the edit button	Details are written on the form as a "hint".	
		Able to update the item	Clicking save will save the changes to the product	
	Deleting product	Clicking on "delete?" button should change it to a confirmation button	Button changes to the confirmation button of "delete" and "cancel"	
		Clicking on the confirm delete button will delete the item	Product is removed from the inventory and won't be shown	
Sales	Able to access the website	User is logged in	Sales page should be shown	

		User is not logged in	Unable to access sales page and redirected to log in page	
--	--	-----------------------	---	--