CSC312 Capstone project Assignment 4
CtrlAltElite

Amaar Behardien, Amir Davids, Siphiwe Clinton Mthebule,
Vanick Wilfred Jouemo Semegni, Mickyle James, Reid Ebden
4125732, 4165406, 4041743, 4110829, 4143342, 4115316
4125732@myuwc.ac.za, 4165406@myuwc.ac.za,
4041743@myuwc.ac.za, 4110829@myuwc.ac.za,
4143342@myuwc.ac.za, 4115316@myuwc.ac.za

ABSTRACT

The aim of our capstone project is to overhaul the online thrift store by introducing a machine learning algorithm to provide sellers with a suggested price for their items and to create a database in which administrators can manage the web application. To obtain this goal, we are implementing the agile process model. We have planned thoroughly and have developed different models that will aid us in the construction of the additional features. Moving on we will develop, test, and implement the additional features.

1 INTRODUCTION

Thrifting is the act of shopping at stores that sell secondhand items at discounted prices. While thrifting has been around for ages it has become extremely popular in recent years. Our group (CtrlAltElite) decided to jump onto this opportunity by providing a platform in which anyone can sell their unwanted clothes online. This platform will also allow people to thrift from anywhere, on any device with ease. What do we call it? ThriftIt.

2 COMMUNICATION

2.1 Objective:

The objective is to improve on the online thrift store by creating a database whereby users and products can be managed and adding a feature that allows sellers to efficiently sell products by suggesting a price for their product that ensures the sale of their product while making as much money as possible.

2.2 STAKEHOLDERS

1. Customers
2. Sellers
3. Website Administrators
4. Payment Gateway Providers
5. Courier partners

2.3 FUNCTIONAL REQUIREMENTS

2.3.1 User Registration

- Users (buyers and sellers) need to be able to create accounts.
- User accounts should be secure and require authentication.

2.3.2 Product Catalog

- A catalog of products must be displayed to users.

2.3.3 Product Description

- Provide a detailed description of the product on sale, including the brand, size, images, and a condition rating.

2.3.4 Shopping Cart

- The shopping cart needs to display products that users add to their cart as well as the total of all products in the cart.

2.3.5 Checkout and Payment

- Customers need to be able to enter shipping information, and credit card information to make their purchase.
- An email should be sent to customers containing their order information.

2.3.6 User Profiles

- Users can edit their address, contact and personal information.

2.3.7 Product Management

- Provide a panel for administrators to manage products.
- Be able to add and remove products from the catalog as well as store and retrieve product information such as name, description and images associated with the product.

2.3.8 Process User Requests

- Allow users to submit a request to have their clothing item sold through the online thrift store.

2.3.9 Help Option

- Provide a section that explains how the online thrift store works and how to sell their items.

2.3.10 Product Price Suggestion

- Provide a suggested selling price to sellers for their products that ensures a sale at an objectively fair price.
- Suggested price should be reasonable and within market range for the product.

2.4 NON-FUNCTIONAL REQUIREMENTS

2.4.1 Performance:

- The system should be able to efficiently handle a high number of listings and user interactions.
- User data and payment information should be handled securely.
- The system should respond quickly to searches, item views, and transactions.

2.4.2 Security:

- User data and payment information should be handled securely.

2.4.3 Usability:

- Provide intuitive and user-friendly interface for sellers to list items and for customers to buy items.
- Provide a user-friendly interface for administrators to manage products.

2.4.4 Scalability:

- The system should be able to scale to handle an increase in the number of users and listings.

2.4.5 Data Backup and Recovery:

- Regularly backup customer data and order history.
- Implement a disaster recovery plan.

2.5 Constraints

2.5.1 Budget:

- The project has a budget of R0, and therefore only free software can be used.

2.5.2 Legal Compliance:

- Ensure compliance with data protection regulations (e.g., POPIA).
- Abide by e-commerce regulations and consumer protection legislation.

2.6 Assumptions:

- All the technology required to create, manage, and host the online store will be available to use for free.

3 PLANNING

Given our objective, a meeting was held to discuss the scope of the project, objectives and how this would be done. The meeting resulted in the creation of four main objectives. The objectives are to fix bugs on the initial web-application, create a machine learning algorithm that suggests a selling price for a product, create a database to store and manage user and product data, and create a web scraper to retrieve data for the machine learning algorithm. It was also decided that we would be following the agile process model (*Figure 1*). The

waterfall approach was chosen to create a product that can be improved as much as possible to ensure a quality final product and to cater for change.

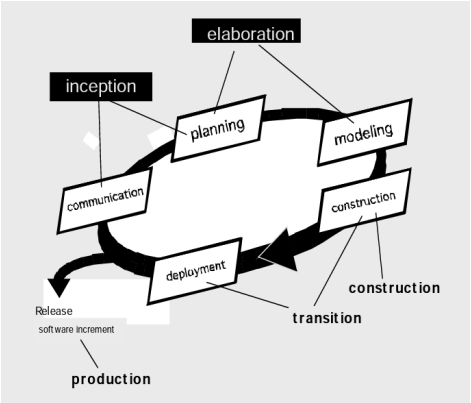


Figure 1 Agile process Mode

3.1 Timeline

In parts of our planning efforts, we have created an initial timeline (*Figure2*) so that we can track and control our project.

28 Sep	Decide on groups can change as the project progresses. Assignment 2: Design Model - fixes from feedback Prototype			
	DB Group: Clint & Mickyle Designing DB structure. Created Sequel server	Web Scrapper Group: Amir Will be focusing on Assignment 2 Fixes	ML Group: Ried Vanik Looking for data to use.	Mini Project Fixes Group: Ammear Fix CSS And create pages for sell items input Use in Assignment 3 video
1 Oct	Review Assessment 2 Feedback from each group			
	DB Group: Feedback on Implementation	Web Scrapper Group: Feedback on research	ML Group: Research / find data	Mini Project Fixes Group: CSS Feedback
2Oct	Assessment 2 Design Model/ Prototype			
	DB Group: Feedback on Implementation	Web Scrapper Group: Feedback on development	ML Group: Research / find data	Mini Project Fixes Group: CSS Review
5 Oct				
	DB Group: Create DB Tables for items and Customers	Web Scrapper Group: Feedback on development	ML Group: Research how to connect model to website	Mini Project Fixes Group: Email feature
9 Oct				
	DB Group: Feedback on tables	Web Scrapper Group: Feedback on development	ML Group: Feedback Build model	Mini Project Fixes Group: Email feature feedback
12 Oct				
	DB Group:	Web Scrapper Group: Start gathering data and connect to DBs built	ML Group: Build model	Mini Project Fixes Group: Add
16 Oct				
			ML Group:	

Figure 2 Timeline

4 MODELING

4.1 Use Cases

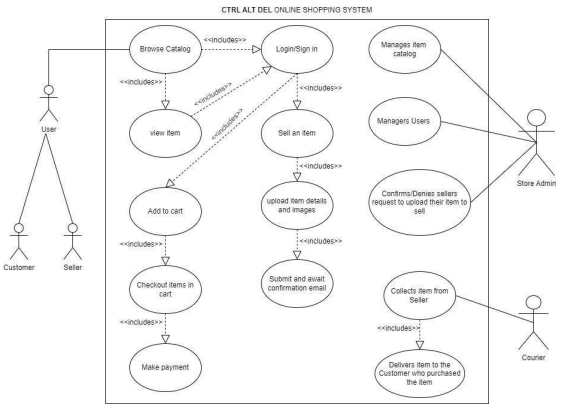


Figure 3 Use Case Diagram

4.2 Use Case Description

4.2.1 Browse catalog:

The customer requests to view products in the catalog. The system displays the products available to the customer.

4.2.2 View item:

The customer clicks on a product of interest. The system displays more information about the product including a description of the product, price, condition, and size of the product.

4.2.3 Add to Cart:

When a customer decides to purchase a product, they add it to their cart and the system keeps track of all the products the customer would like to buy as well as a total price of all products in the cart.

4.2.4 Checkout items in cart:

When the customer has added all the items they would like to buy to their cart and is ready to make the purchase they request a checkout. They will then have to fill in their credit card information. The card information will then be verified and processed. If successful, the checkout will be completed.

4.2.5 Make Payment:

After entering their banking details, they will be prompted to verify the payment through their banking app for the payment to go through.

4.2.6 Login/ Sign in:

The customer can log in by entering their username and password.

4.2.7 Sell an Item:

Users that would like to sell a product through the store request to go to the sell item page. The sell item page will be displayed to the customer.

4.2.8 Upload item details and images:

Sellers can upload item descriptions and photographs, set prices, and arrange shipment for products they would like to sell.

4.2.9 Submit and await confirmation email:

After a customer enters the product information, they submit their request and will receive an email confirming their request.

4.2.10 Manage item catalog:

A store admin will add/edit/remove products from the online store.

4.2.11 Manage Users:

A store admin will manage user account issues for the store.

4.2.12 Accept or Reject product requests:

A store admin will review seller product requests and will either accept or reject the request.

4.2.13 Collect product from seller:

If a seller chooses to, they can courier their product instead of dropping it off in person.

4.2.14 Deliver product/s to customer:

A courier service will be outsourced to deliver the product to buyers.

4.3 Class Diagram

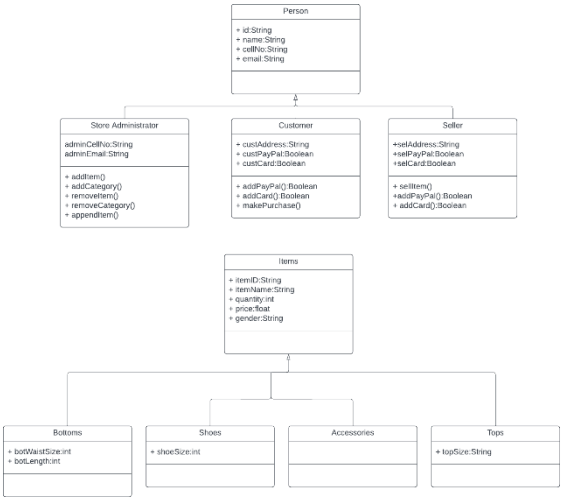


Figure 4 Class Diagram

4.4 Database Design

We have identified 5 different entities in this database. They include:

- Users (userID (PK), username, userCellNo, userPassword, userEmail, userAccount, userBankName, userAddress)
- Baskets (basketID (PK), userID (FK))
- Items (itemID (PK), itemDesc, itemSize, itemStatus, itemBrand, itemPrice, itemGender, userID (FK), basketID (FK))
- Cards (userID (FK), cardNumber (PK), cardCVV, cardExpDate)
- Invoices (invID (PK), basketID (FK), invDate, invDesc, invTotal)

Each USER can sell zero or many ITEMS. Each ITEM must be sold by one USER.

Each USER can have only one BASKET. Each BASKET belongs to only one USER.

Each USER can have zero or many CARDS. Each CARD must belong to one USER.

Each BASKET can contain zero or many ITEMS. Each ITEM can belong to zero or many BASKETS.

Each BASKET can contain zero or many INVOICES. Each INVOICE belongs to only one BASKET.

BasketQueue has been created as an associative entity to resolve the Many-to-Many relationship between ITEMS and BASKETS.

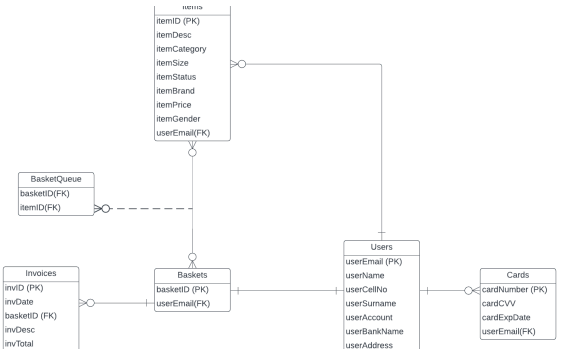


Figure 5 ER diagram

5 DESIGN MODEL

As the purpose is to improve on the online thrift store by adding features, only the design for pages related to the new features will be included in the design model.

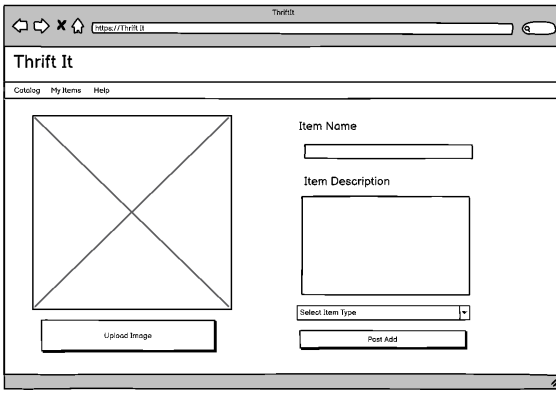


Figure 6

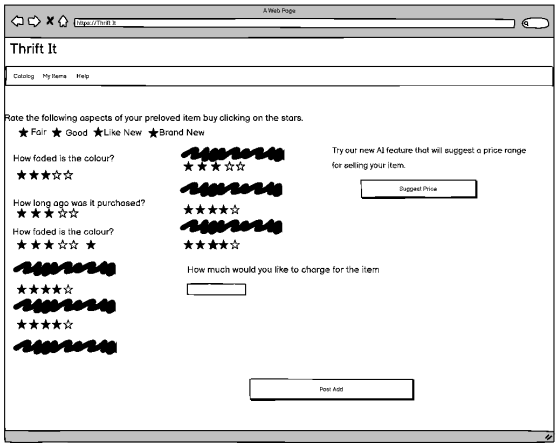
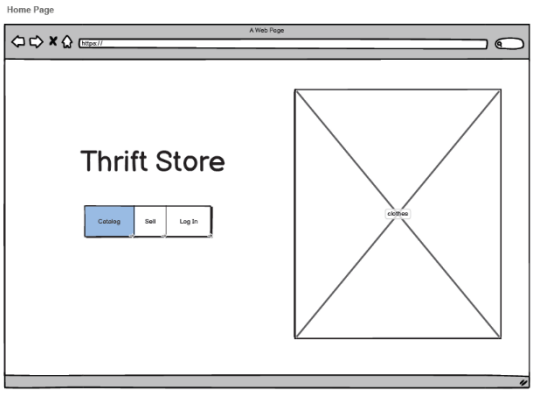
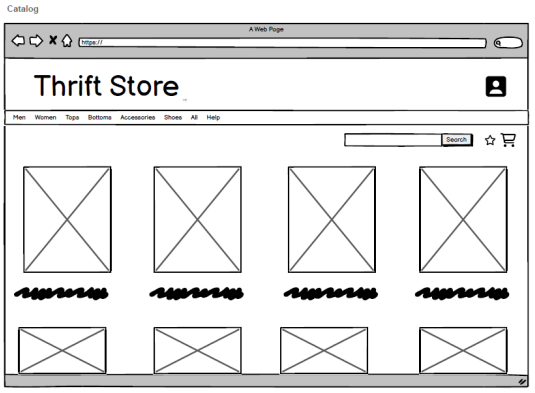
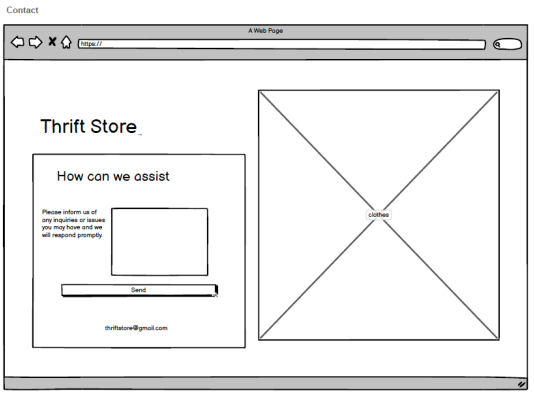


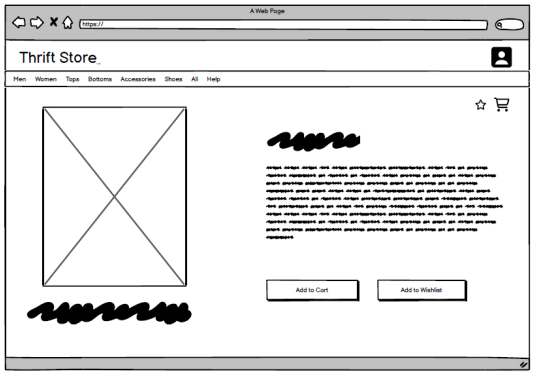
Figure 7



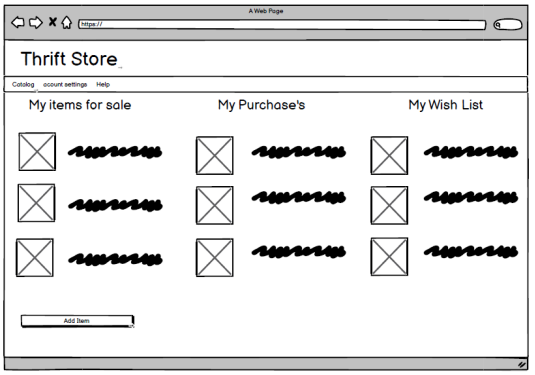




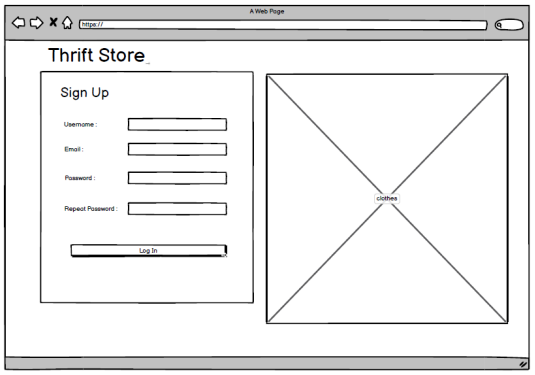
View Item



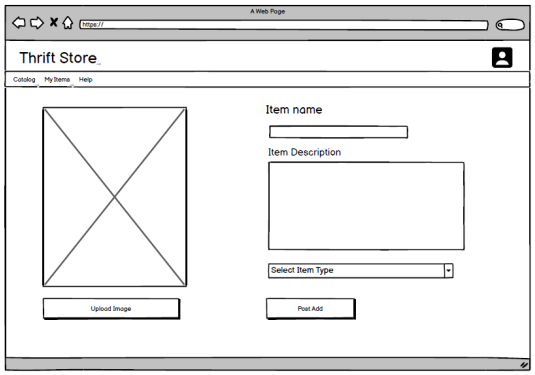
User Acount



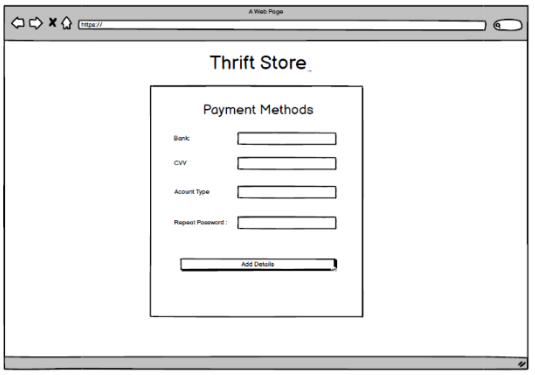
Sign Up



Sell Item



Payment method



6 IMPLEMENTATION MODEL

6.1 Feature 1: Uploading and Describing Clothing Items

The primary update to the platform involves a user-friendly interface for uploading clothing items. In figure 7, users will be able to upload images of their products by simply clicking on an "Upload Image" button. This image upload is complemented by description boxes to provide essential details about the item, including its name, description, and type. To proceed, users can click a "Post Ad" button, effectively submitting the image file along with its associated information.

This feature facilitates a seamless process for users to contribute their clothing items for evaluation. The inclusion of image uploads and detailed descriptions ensures that the company receives comprehensive information, improving the evaluation process.

6.2 Feature 2: Star Rating and Pricing Estimation

Figure 7 builds upon the first, offering users the opportunity to rate their clothing items before listing them for sale. This feature aims to reduce manual labour and expedite the quality confirmation process. Users will have the option to assign star ratings based on various factors such as colour fading, wear and tear, and the item's age. Additionally, users can estimate the selling price they desire for the item.

By allowing users to rate their items, the platform can collect valuable data regarding the quality and condition of each product. This data can significantly enhance the efficiency of the evaluation process, as it reduces the need for extensive manual inspection.

6.3 Feature 3: AI-Powered Pricing Suggestions

One of the most innovative additions to the second page incorporates elements of machine learning (ML) and artificial intelligence (AI). This advanced feature utilises the data collected from user ratings, descriptions, and images to suggest a price range or even a specific single price for the item. The AI-driven pricing suggestions consider various factors, including the item's condition, market trends, and historical data.

By implementing this AI-driven pricing feature, the platform empowers users with valuable insights into the potential market value of their clothing items. This not only simplifies the listing process but also ensures that users have a realistic expectation of the selling price, improving overall satisfaction.

7 MENTAL MODEL

7.1 Homepage

The homepage can be seen as the entrance to the store. It's the first thing that users see when they visit the webpage. The homepage displays recently added products and offers a taste of the store's offerings.

7.2 Navigation Bar

Imagine the navigation bar as boards navigating customers to different product categories, like "Men's," "Women's," "Tops," "Bottoms," and more.

7.3 Product Details Page

View the product details page as the sales assistant. When a customer views the product details page they are provided with more information, such as size and colour options, that a sales assistant provides.

7.4 Shopping Cart

See the shopping cart as a physical shopping cart that holds the products that you are going to purchase.

7.5 Checkout Page

Picture the checkout page as the physical till where you pay for your products. The customer can review the items in their cart, and fill in payment and address details, to complete the purchase.

7.6 Search Bar

Consider the search bar as a helpful employee, ready to help to find certain products throughout the store. Users can search for specific clothing items.

7.7 Filter

The filter option can be seen as going to an aisle in a store that holds a specific category of items. Customers can filter products by certain categories.

7.8 Seller Request page

View the seller request page as pitching a product for a store to sell on your behalf. Users pitch their products to be sold on the website.

7.9 Mental Model Overview

This mental model enhances the customers' shopping experience by giving them a familiar process to make it easier to purchase items through the website. It helps them to traverse the website which ensures a good experience.

The development of our assignment followed a strict and planned out approach, paying close attention to the principles and methods amongst many more involved in software engineering. several processes were to be completed before our submission, the tasks to be done included building our regression function for our machine learning aspect of the assignment, compiling data for the regression model, building the web scraper to gather additional data, adjusting our website, and completing the database. We followed an iterative process flow as we held a meeting many days ago, broke up the tasks into individual portions and planned another meeting a few days after to touch base on our design and proposal for the implementation of our tasks. We then completed our tasks and held another meeting to present them to each other. As seen above we regularly met in discussion of advancements made in the project. Some of the key software engineering principles that we followed included principles that guide process and practice, communicating processes, modeling processes and construction processes.

7.10 Principles that guide process and practice.

We implemented the idea of dividing and contouring our main task, where we split up into effective teams that would handle portions of our goal. We were split into machine learning, database, and web groups where Vanick, Amir and Reid handled the machine learning, Clinton and Mickyle handles the database, and Amaar handles the web work. Additionally, here are some crucial principles that we followed:

- Communication principles: Each meeting was well prepared as it was discussed prior as to what we'd be meeting for and what the topic of discussion would be. Additionally, each meeting was mostly

facilitated by Ameer to achieve a structured discussion.

- Modeling principles: Design models were made well in advance to visualize the user interfaces for our website.

- Construction principles: Our regression function for our machine learning component of the assignment was being tested on dummy datasets while in development as data was still being collected for our website.

7.11 Machine Learning features

7.11.1 Data Collection

The requirements for this model would be data. We collected data off of resale websites that have already been established on the internet for a number of years, some data was collected manually for variation and other was collected using a web scraper.

7.11.2 Price Suggestion

Our main goal for this submission was building our AI model for suggested feedback on price when a user would want to sell an item on our website and the web scraper to gather data for it. This was achieved by implementing multiple regression where we built a regression model. We built the regression model using a machine learning model in node.js and used the data gathered from the web scraper to teach the algorithm. The web scraper would primarily be used to fetch data from yaga.co.za which is an online marketplace for pre-loved clothing items. This is the perfect website to collect data for our model. Yaga dynamically loads its products and thus a web scraper that can fetch data from a dynamic web website was needed. To do this the python ‘Requests-HTML’ library was used. The data was retrieved using functions from this library, then formatted and stored in a csv file which was made available to train the machine learning model.

7.11.3 Image Validation

Used Googles teachable machine to create an image classifier model to identify clothes and items from an image. This was created to validate if uploaded products are clothing items or not. Our goal for this assignment was to achieve a complete website that includes all of our planned ideas in a timely manner. As explained above the inclusion of technical reviews, project tracking and project control helped us achieve our goal.

8. Umbrella Activities

8.1 Project Management

Throughout this project, we have been implementing Project Management by making use of Change Management and Project Schedule Management, and by reviewing our designs and models after each phase. We have implemented several changes to our Use Case Diagram and ERD diagram. We also changed the DBMS that we use to create our DB, switching from SQL Server to MySQL.

An example of Project Schedule Management is our Timeline that has been indicated above in Figure 2.

8.2 Quality Assurance

When completing a phase or a certain task in this project, we have to evaluate whether or not our deliverables/output meets the desired requirements. For example, as the application was being developed, our database had to be reconstructed as there were some possibilities for optimisation. We have removed the attribute of userID and made userEmail the new primary key of the USERS table. This decision was made because userID was an unnecessary attribute as a user’s email address is already unique. So to avoid redundancy and to optimize the usage of space, we have replaced userID. Our diagrams such as our ERD and Class Diagrams, had to be changed as new information or new requirements were introduced.

9. Software and Hardware Implementations

9.1 Software

In the development of our online thrift store, we've employed a powerful combination of software tools to deliver an exceptional user experience.

- Node JS: At the core of our application, Node.js serves as the robust server-side JavaScript runtime, handling the backend logic and seamlessly integrating the "ml-regression-multivariate-linear" module, which powers our machine learning model for clothing pricing. Node.js plays a pivotal role in efficiently processing user requests, managing data, and facilitating communication with various databases.
- React: a widely adopted JavaScript library, was our choice to create a responsive and visually appealing user interface. Its component-based architecture simplifies the process of designing an engaging and user-friendly website. Real-time rendering capabilities offered by React ensure a dynamic and interactive shopping experience for our users.

- Firebase: To manage our data effectively, we turned to Firebase for user authentication and real-time database updates. Firebase's authentication services guarantee secure user registration and login processes, while the real-time database facilitates instant item listings and updates. Additionally, we integrated MySQL for robust and structured data storage, enabling us to efficiently manage and query our extensive thrift store inventory.

Main Pages Test Cases	**Pass /Fail**	
Test Home page	Pass	
Test Catalog page	Pass	
Test Sell page	Pass	
Test Checkout page	Pass	
Test Login / Sign Up page	Fail	When this page is loaded a blank screen is displayed for the Login / Sign up component of this page.
Test Successful Purchase page	Pass	

This powerful combination of Node.js, React, Firebase, MySQL, and the "ml-regression-multivariate-linear" module has allowed us to create a feature-rich online thrift store. Users can seamlessly buy and sell items while enjoying a visually pleasing and fluid shopping environment. Our tech stack ensures scalability, security, and the capability to accommodate a growing user base, making our platform a go-to destination for thrift shoppers and sellers alike.

9.2 Hardware

Moreover, we are committed to providing a responsive, mobile-first experience. Recognizing the importance of catering to users on the go, we've designed our platform to ensure that users can access the thrift store seamlessly from smartphones and tablets, enabling them to shop and list items from virtually anywhere. In our development process, we followed an agile methodology, providing individual laptops to our team members, facilitating remote work and effective collaboration. GitHub served as a central hub for version control, making it easy to manage codebase changes, merge contributions, and track issues. For collaboration on design assets, documentation, and project-related files, we relied on Google Drive. This combination of hardware and collaboration tools fostered a cohesive development process and encouraged a high level of productivity among our team members.

10.TESTING

The following test cases where evaluated to assess the design model , interface and functionality of the website. The design model is evaluated to uncover any accessibility issues in completing use cases. The interface is tested to asses design and structure of the website an the functionality test cases are designed to assess the functional components of the website.

10.1 Design Model Test Cases

Accessibility Test Cases	**Pass /Fail**	**Comment**
Is Home page accessible?	Pass	
Is Catalog page accessible?	Pass	
Is Sell page accessible?	Pass	
Is Checkout page accessible?	Pass	
Is Login / Sign Up page accessible?	Fail	When this page is loaded a blank screen is displayed for the Login / Sign up component of this page.
Is Successful Purchase page accessible?	Pass	

10.2 Interface Test

Product Details Test Cases	**Pass /Fail**	**Comment**
Check product title	Pass	

Check product description	Pass	
Check product image	Pass	
Check product pricing	Pass	
Test image select	Pass	
Test Buy Now (From item details)	Fail	Opens Cart

Components Test Cases	**Pass/Fail**
Check Navbar for Logo and Cart (bag) icon on all pages.	Pass
Check Home page for Banner image and text.	Pass
Check Home Page contains moving recommended items.	Pass
Check items details contains moving recommended items.	Pass

10.3 Functionality Tests

Cart Test Cases	**Pass /Fail**
Test Open Cart	Pass
Test Buy Now (from cart)	Pass
Test Remove from cart	Pass
Check Cart Total	Pass
Test Add to Cart	Pass

Sell Item Test Cases	**Pass /Fail**	**Comment**
Test Input fields	Pass	
Ensure all necessary Inputs are entered before submitting.	Pass	
Test Upload Image	Pass	
Test Machine Learning feature	Pass	
Test Submit via email	Pass	Delay between submitting and receiving email.

Test Login/ Sign Up	**Pass/Fail**	**Comment**
Test Log in check username and password.	Fail	Not Accessible
Test Log in forgot password.	Fail	Not Accessible
Test Sign Up create password	Fail	Not Accessible
Test Sign Up repeat password	Fail	Not Accessible
Test Sign Up validate email	Fail	Not Accessible

Price Prediction Test Cases	**Pass/ Fail**	**Comment**
Collecting input	Pass	Once submitted, inputs are sent to the server to handle the prediction.
Loads the model	Pass	Server loads the models needed to make predictions.
Makes Prediction	Pass	Functions required to make predictions work when given the valid inputs.
Accurate Predictions	Fail	Very limited data set used to train the model, hence the predicted price is often inaccurate or unreasonable predictions.
Outputs predicted Price	Pass	Server responds with predicted price which is displayed for the user to see.
Error handling	Pass	In a case of invalid inputs given, then predictions won't be made on those inputs.

10.4 Browser Compatibility Test Cases

Test Functionality in the following Browsers	**Pass /Fail**

Chrome	Pass
Microsoft Edge	Pass
Opera	Pass

11 Database Implementation

11.1 SQL Statements

Create Tables

We will first create the tables that we will use in this application. The tables have been numbered in the sequence they should be created to ensure referential integrity:

11.1.1 USERS table

```
create table if not exists USERS(
        userEmail nvarchar(50) NOT NULL,
        userSurname text NOT NULL,
        userName text NOT NULL,
        userCellNo text NOT NULL,
        userAccount text,
        userBankName text,
        userAddress text,
        PRIMARY KEY(userEmail)
);
```

11.1.2 CARDS table

```
create table if not exists CARDS(
        cardNumber nchar(12) NOT NULL,
        cardCVV nchar(3) NOT NULL,
        cardExpDate nchar(5) NOT NULL,
        userEmail nvarchar(50) NOT NULL,
        PRIMARY KEY(cardNumber),
        FOREIGN KEY(userEmail) REFERENCES USERS(userEmail)
);
```

11.1.3 ITEMS table

```
create table if not exists ITEMS(
        itemID int  NOT NULL AUTO_INCREMENTAL,
        itemDesc text NOT NULL,
        itemCategory text NOT NULL,
        itemSize text,
        itemStatus text NOT NULL,
        itemBrand text NOT NULL,
        itemPrice money NOT NULL,
        itemGender nchar(1),
        userEmail nvarchar(50) NOT NULL,
        PRIMARY KEY(itemID),
        FOREIGN KEY(userEmail) REFERENCES USERS(userEmail)
);
```

11.1.4 BASKETS table

```
create table if not exists BASKETS(
        basketID int NOT NULL AUTO_INCREMENT,
        userEmail nvarchar(50) NOT NULL,
        PRIMARY KEY(basketID),
        FOREIGN KEY(userEmail) REFERENCES USERS(userEmail)
);
```

11.1.5 BASKETQUEUE table

```
create table if not exists BASKETQUEUE(
        dateAdded text NOT NULL,
        basketID int NOT NULL FOREIGN KEY REFERENCES BASKETS(basketID),
        itemID int NOT NULL FOREIGN KEY REFERENCES ITEMS(itemID),
);
```

11.1.6 INVOICES table

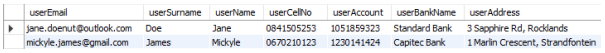
```
create table if not exists INVOICES(
        invID int NOT NULL AUTO_INCREMENT,
        invDate text NOT NULL,
        invDesc text NOT NULL,
        invTotal money NOT NULL,
        basketID int,
        PRIMARY KEY(invID),
```

FOREIGN KEY(basketID) REFERENCES BASKETS(basketID)

);

11.2 Table Data

11.2.1 USERS Table

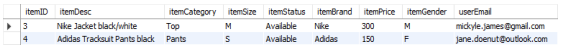


userEmail	userSurname	userName	userCellNo	userAccount	userBankName	userAddress
jane.doenut@outlook.com	Doe	Jane	0841505253	1051859323	Standard Bank	3 Sapphire Rd, Rocklands
mickyle.james@gmail.com	James	Mickyle	0670210123	1230141424	Capitec Bank	1 Marlin Crescent, Strandfontein

This statement is what is used to insert an item into the USERS table:

“insert into USERS values(‘”+ userEmail+”’,’”+ userSurname+”’,’”+ username+”’, ‘”+ userCellNo + “’,’” + userAccount + ”’,’” + userBankName + “’,’” + userAddress +”’);”

11.2.2 ITEMS Table



itemID	itemDesc	itemCategory	itemSize	itemStatus	itemBrand	itemPrice	itemGender	userEmail
3	Nike Jacket black/white	Top	M	Available	Nike	300	M	mickyle.james@gmail.com
4	Adidas Tracksuit Pants black	Pants	S	Available	Adidas	150	F	jane.doenut@outlook.com

This statement is what is used to insert an item into the ITEMS table:

“insert into ITEMS values(‘”+ itemDesc+”’,’”+ itemCategory+”’,’”+ itemSize+”’, ‘Available’,’”+ itemBrand + “’,” + itemPrice + ”,’” + itemGender + “’,’” + userEmail +”’);”

11.2.3 CARDS Table

cardNumber	cardCVV	cardExpDate	userEmail
501241242...	722	09/26	mickyle.j...
501290095...	643	10/24	jane.do...

This statement is what is used to insert an item into the CARDS table:

“insert into CARDS values(‘”+ cardNumber+”’,’cardCVV+”’,’”+ cardExpDate+”’,’” + userEmail +”’);”

11.2.4 BASKETS Table

basketID	userEmail
2	jane.doenut@outlook.com
1	mickyle.james@gmail.com

This statement is what is used to insert an item into the BASKETS table:

“insert into BASKETS values(“+basketID+”,’”+ userEmail+”’);”

11.2.5 BASKETQUEUE Table

dateAdded	basketID	itemID
1/06/2023	1	4
25/05/2023	2	3
NULL	NULL	NULL

This statement is what is used to insert an item into the BASKETQUEUE table:

“insert into BASKETQUEUE values(‘”+ dateAdded+”’,”+basketID+”,”+ itemID+”);”

11.2.6 INVOICES Table

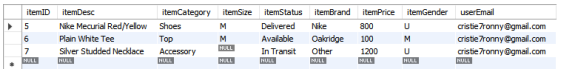
invID	invDate	invDesc	invTotal	basketID
1	01/10/...	Nike Ja...	300	2
2	25/9/2...	Adidas...	150	1

This statement is what is used to insert an item into the INVOICES table:

“insert into INVOICES values(‘”+ invDate+”’,’invDesc+”’,”+ invTotal+”,” + basketID +”);”

11.3 SQL QUERIES

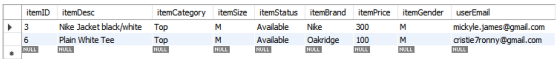
11.3.1 SELECT * FROM items WHERE userEmail= ‘cristie7ronny@gmail.com’;



itemID	itemDesc	itemCategory	itemSize	itemStatus	itemBrand	itemPrice	itemGender	userEmail
5	Nike Mecurial Red/Yellow	Shoes	M	Delivered	Nike	800	U	cristie7ronny@gmail.com
6	Plain White Tee	Top	M	Available	Oakridge	100	M	cristie7ronny@gmail.com
7	Silver Studded Necklace	Accessory	NULL	In Transit	Other	1200	U	cristie7ronny@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This query is used to filter out items being sold and only displays items sold by a specific user. This can be used in the user profile.

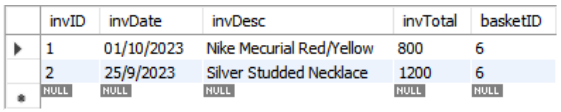
11.3.2 SELECT * FROM items WHERE itemSize = ‘M’ AND itemCategory=’Top’;



itemID	itemDesc	itemCategory	itemSize	itemStatus	itemBrand	itemPrice	itemGender	userEmail
3	Nike Jacket black/white	Top	M	Available	Nike	300	M	mickyle.james@gmail.com
6	Plain White Tee	Top	M	Available	Oakridge	100	M	cristie7ronny@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This query can be used to do a filtered search for items that have a specific size. In this example we are searching for items that are available in a medium size and are tops.

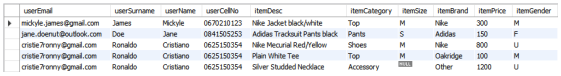
11.3.3 SELECT * FROM invoices WHERE basketID = 6;



invID	invDate	invDesc	invTotal	basketID
1	01/10/2023	Nike Mecurial Red/Yellow	800	6
2	25/9/2023	Silver Studded Necklace	1200	6
NULL	NULL	NULL	NULL	NULL

This query is used to find the purchase history of a user based on their basketID. In this example, we have gathered all the items purchased by the user who owns the basket of basketID = 6.

11.3.4 SELECT userEmail, userSurname, userName, userCellNo, itemDesc, itemCategory, itemSize, itemBrand, itemPrice, itemGender FROM users NATURAL JOIN items;



userEmail	userSurname	userName	userCellNo	itemDesc	itemCategory	itemSize	itemBrand	itemPrice	itemGender
mickyle.james@gmail.com	James	Mickyle	0670210123	Nike Jacket black/white	Top	M	Nike	300	M
jane.doenut@outlook.com	Doe	Jane	0841505253	Adidas Tracksuit Pants black	Pants	S	Adidas	150	F
cristie7ronny@gmail.com	Ronaldo	Cristiano	0625150354	Nike Mecurial Red/Yellow	Shoes	M	Nike	800	U
cristie7ronny@gmail.com	Ronaldo	Cristiano	0625150354	Plain White Tee	Top	M	Oakridge	100	M
cristie7ronny@gmail.com	Ronaldo	Cristiano	0625150354	Silver Studded Necklace	Accessory	NULL	Other	1200	U

This query displays all the items that are being sold and which users are selling them. This can be used to the admin's benefit, which displays the seller's information to the admin, to be able to contact them. This can also be used in the item information page for buyers to be able to contact the sellers as well.

Field name	Data type	Field Le
Item ID	Int	20
ItemDesc	Text	100
ItemCategory	Text	20
ItemSize	Text	20
ItemStatus	Text	20
ItemBrand	Text	20
ItemPrice	nChar	1

Field name	Data type	Field Length
ItemGender	nvarChar	50
UserEmail	Int	10
dateAdded	text	50
BasketID	Int	20
invID	Int	100
InvDate	Int	10

11. 4 DATA DICTIONARY

Field name	Data type	Field Le
UserEmail	Int	10
dateAdded	text	50
BasketID	Int	20
invID	Int	100
InvDate	Int	10
invDesc	Text	100
invTotal	Money	100
userName	Text	50
userSurname	text	50
userCellNo	Int	10
userEmail	Text	20

Field name	Data type	Field Length
userAccount	text	20
userBankName	Text	10
userAddress	Text	20

Field name	Data type	Field Length
userAddress	Text	20
cardNumber	nchar	16
cardCVV	Int	5
cardExpDate	Int	10