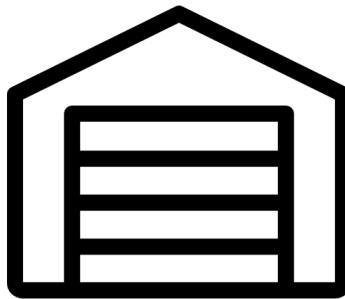


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SPRING 2023**



**TEAM STOCKERS
GENERAL INVENTORY MANAGEMENT**

**IVAN CUARENTA
JOSEPH SALAS
JEREMY WALKER
RODNEY RAMIREZ**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	07.16.2023	IC, JS, JW, RR	document creation
0.2	08.02.2023	JW	Edited Front end, Back-end, Database, Encryption, Price Prediction, and updated images
1.0	8.7.2023	JW	Final Review

CONTENTS

1	Introduction	6
2	System Overview	6
3	Front End	7
3.1	Front End Hardware	7
3.2	Front End Operating System	7
3.3	Front End Software Dependencies	7
3.4	Landing Page	7
3.5	Dashboard	8
3.6	Login	9
3.7	Registration	9
4	Back-end Layer Subsystems	11
4.1	Back-end Hardware	11
4.2	Back-end Operating System	11
4.3	Back-end Software Dependencies	11
4.4	Adding/updating asset	11
4.5	Removing asset	12
4.6	Checkout asset	13
4.7	Logout	14
4.8	Prediction	14
4.9	Generate Report	15
4.10	Layer Hardware	15
4.11	Layer Operating System	15
4.12	Layer Software Dependencies	15
4.13	Generate Report	16
5	Database Layer Subsystems	17
5.1	Database Hardware	17
5.2	Database Operating System	17
5.3	Database Software Dependencies	17
5.4	Company	17
5.5	User	18
5.6	Table	19
6	Encryption Layer Subsystems	21
6.1	Encryption Hardware	21
6.2	Encryption Operating System	21
6.3	Encryption Software Dependencies	21
6.4	Encrypt data	21
6.5	Decrypt data	22

7	Price Prediction Layer Subsystems	24
7.1	Price Prediction Hardware	24
7.2	Price Prediction Operating System	24
7.3	Price Prediction Software Dependencies	24
7.4	Pricing Algorithm	24

LIST OF FIGURES

1	System Architecture	6
2	Front End layer diagram	7
3	Dashboard subsystem diagram	8
4	Front End layer diagram	9
5	Front End layer diagram	10
6	Back-end diagram	11
7	Back end diagram	12
8	Back-end diagram	13
9	Back-end diagram	14
10	Back-end diagram	15
11	Back-end diagram	16
12	Database diagram	17
13	Database diagram	18
14	Database diagram	19
15	Encryption diagram	21
16	Encryption diagram	22
17	Price Prediction diagram	24

LIST OF TABLES

1 INTRODUCTION

This application is a general inventory management system for managing user's inventory. The general inventory management system allows users to track, view, and edit inventory, generate reports of selected assets that yields name, category, image, serial number, description, and prices.

2 SYSTEM OVERVIEW

The system contains multiple layers that utilize MongoDB (a database program) which houses the user's credentials and item information. The front end will be where the users can log in, create an account, and view the dashboard to use the system. The back-end layer will be responsible for providing users with price predictions, logging out, and encryption. The encryption layer will be responsible for encrypting and decrypting sensitive data about the assets the users put into the inventory system. The price prediction will have a model that is trained to predict the current market value of the assets the user has in their inventory. The database will hold the user account login id and encrypted password as well as the inventory tables for each user. The reporting and analytics will format and generate the report as specified by the user.

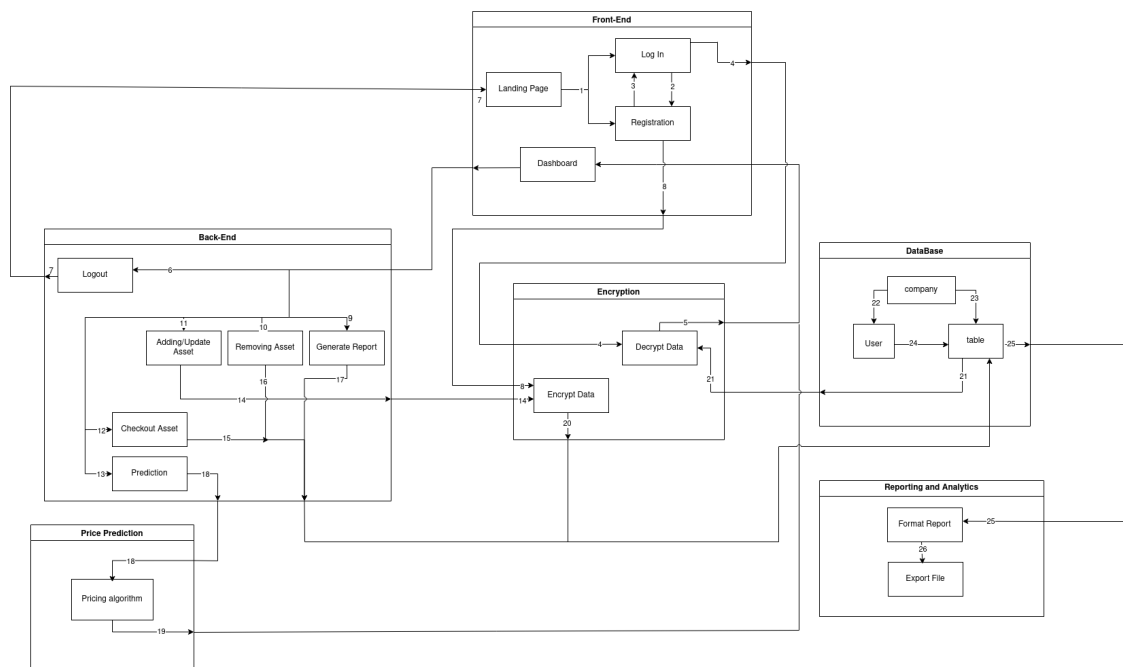


Figure 1: System Architecture

3 FRONT END

The table sub-layer is responsible for directing a user to the login page or the registration page

3.1 FRONT END HARDWARE

A basic Digital Ocean Droplet with 4 Intel vCPUs, 8 GB RAM, and 160 GB disk hard drive.

3.2 FRONT END OPERATING SYSTEM

Ubuntu 22.1- x64

3.3 FRONT END SOFTWARE DEPENDENCIES

- MUI v 5.13.1
- Node.js 18.15.0
- Next.js 9.5.0

3.4 LANDING PAGE

This subsystem is responsible for allowing the user to enter the login page for a returning user or enter the registration page of the website so the user can register an account.

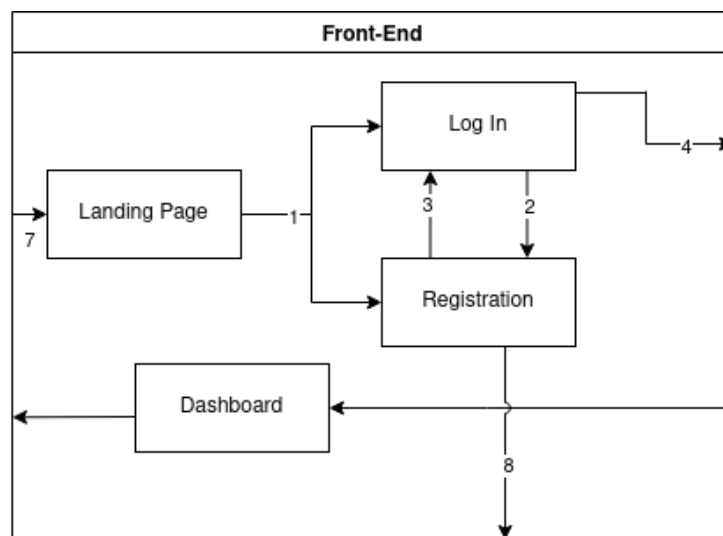


Figure 2: Front End layer diagram

3.4.1 LANDING PAGE HARDWARE

N/A

3.4.2 LANDING PAGE OPERATING SYSTEM

N/A

3.4.3 LANDING PAGE SOFTWARE DEPENDENCIES

N/A

3.4.4 LANDING PAGE PROGRAMMING LANGUAGES

- typescript 5.0.2

3.4.5 LANDING DATA STRUCTURES

There are no classes and data structures located on the landing page.

3.4.6 LANDING PAGE DATA PROCESSING

There is no data processing as the landing page directs the user to the sign-in page or the registration page.

3.5 DASHBOARD

This subsystem is responsible for allowing users to enter items into the table. Users when adding an item can enter the name of the item, type of category, images of the item, serial number that the item may have, and export selected items to a pdf file. Users can add, edit, and remove items.

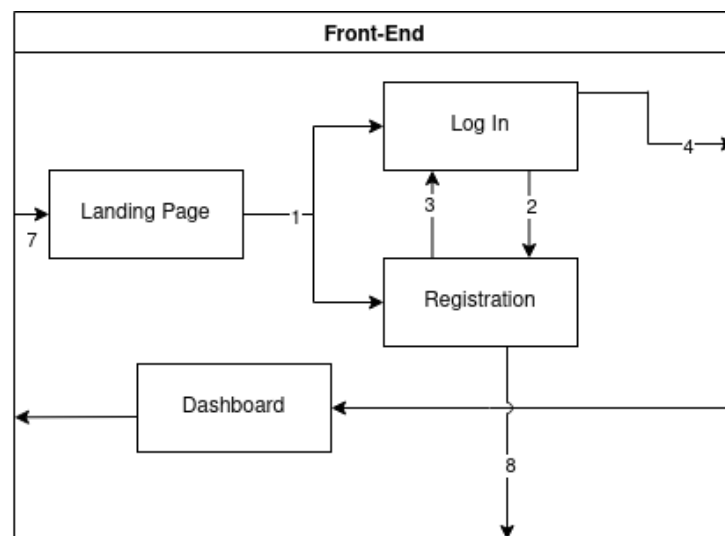


Figure 3: Dashboard subsystem diagram

3.5.1 DASHBOARD HARDWARE

N/A

3.5.2 DASHBOARD OPERATING SYSTEM

N/A

3.5.3 DASHBOARD SOFTWARE DEPENDENCIES

- jspdf 2.5.2
- React 18.2.0

3.5.4 DASHBOARD PROGRAMMING LANGUAGES

- typescript 5.0.2

3.5.5 DASHBOARD DATA STRUCTURES

All data input is stored in an object and sent to the database where it's stored and is fetched when a user logs into the system.

3.5.6 DASHBOARD DATA PROCESSING

Prompt the user to enter their item name, category, image, serial number, and description. Users can edit or delete the item.

3.6 LOGIN

This subsystem is responsible for authenticating that the user's email and password exist in the database.

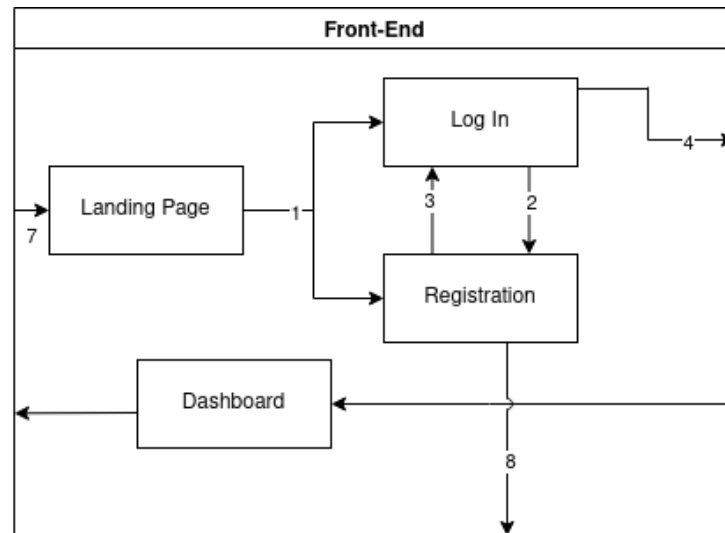


Figure 4: Front End layer diagram

3.6.1 LOGIN HARDWARE

N/A

3.6.2 SUBSYSTEM OPERATING SYSTEM

N/A

3.6.3 LOGIN SOFTWARE DEPENDENCIES

N/A

3.6.4 LOGIN PROGRAMMING LANGUAGES

- typescript 5.0.2

3.6.5 LOGIN DATA STRUCTURES

Data is sent as an object and is checked in the database to see if it matches any login information so the user can enter their own inventory.

3.6.6 LOGIN DATA PROCESSING

Prompt the user to enter their email and password associated with the account.

3.7 REGISTRATION

This subsystem is responsible for sending the user's email and password so it can be saved in MongoDB.

3.7.1 REGISTRATION HARDWARE

Only requires MongoDB to keep the user's registration information saved.

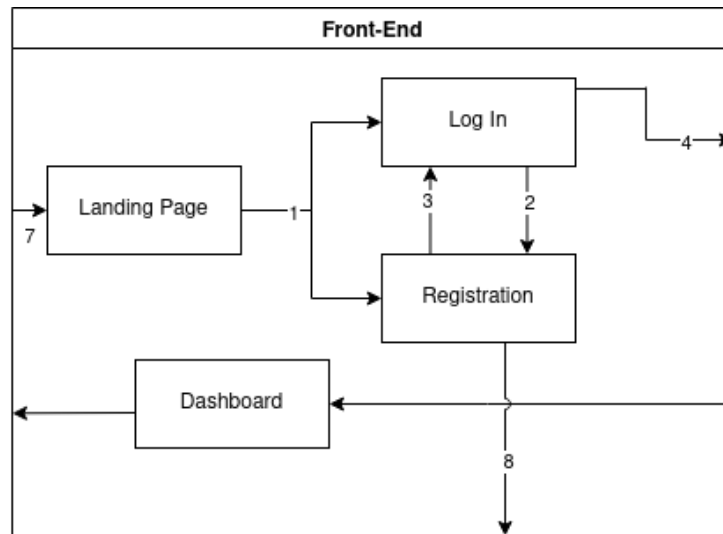


Figure 5: Front End layer diagram

3.7.2 REGISTRATION OPERATING SYSTEM

N/A

3.7.3 REGISTRATION SOFTWARE DEPENDENCIES

N/A

3.7.4 REGISTRATION PROGRAMMING LANGUAGES

- typescript 5.0.2

3.7.5 REGISTRATION DATA STRUCTURES

The data is held in an object and sent to the database.

3.7.6 REGISTRATION DATA PROCESSING

Prompt the user to enter their email and password so it can be used to sign up and sign in for the website.

4 BACK-END LAYER SUBSYSTEMS

The back-end layer is responsible for allowing users to input, update, or remove assets from the database as needed. Users will also be able to obtain reports on their current inventory status. The price prediction will be sent to and from the back end as needed. The back end will also handle logging out of the system.

4.1 BACK-END HARDWARE

A basic Digital Ocean Droplet with 4 Intel vCPUs, 8 GB RAM, and 160 GB disk hard drive.

4.2 BACK-END OPERATING SYSTEM

Ubuntu 22.10 x64

4.3 BACK-END SOFTWARE DEPENDENCIES

- mongoose 7.0.3
- next 13.2.4
- npm 9.7.2
- node 20.4.0

4.4 ADDING/UPDATING ASSET

Adding/updating asset Responsible for allowing the user to add or update an item. The user fills out the data fields and the data is stored as an object and sent to the appropriate MongoDB server.

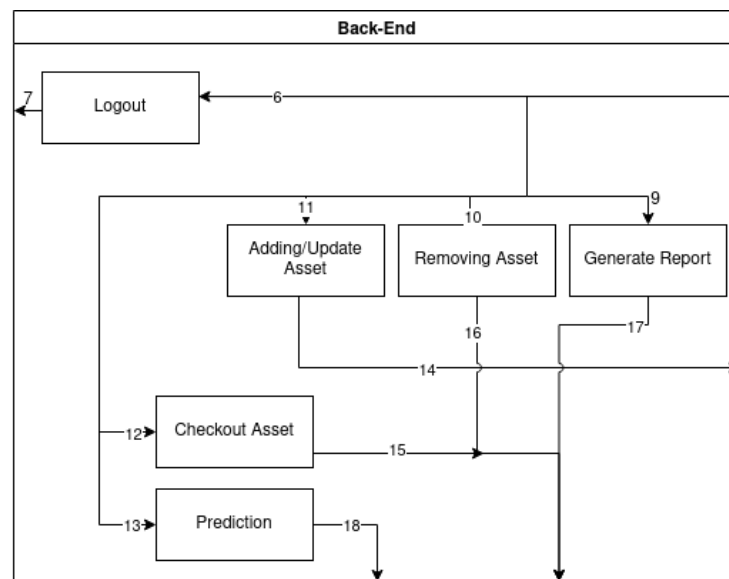


Figure 6: Back-end diagram

4.4.1 ADDING/UPDATING ASSET HARDWARE

N/A

4.4.2 ADDING/UPDATING OPERATING SYSTEM

N/A

4.4.3 ADDING/UPDATING ASSET SOFTWARE DEPENDENCIES

N/A

4.4.4 ADDING/UPDATING ASSET PROGRAMMING LANGUAGES

- typescript 5.0.2

4.4.5 ADDING/UPDATING ASSET DATA STRUCTURES

The item being added or updated is stored as an object with all the fields set to values that were determined by the user.

4.4.6 ADDING/UPDATING ASSET DATA PROCESSING

The user has elected to add or update an item. They insert the appropriate data into the data fields and click on add item.

4.5 REMOVING ASSET

This system will remove the item or items that a user selects to be removed.

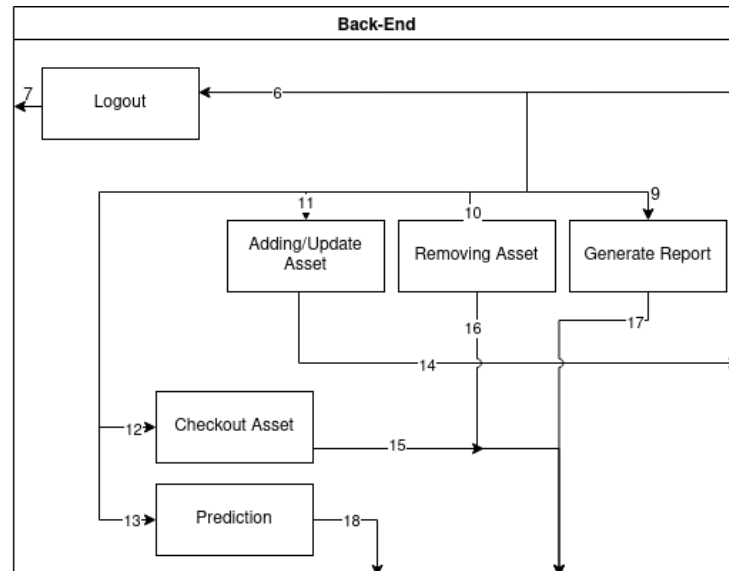


Figure 7: Back end diagram

4.5.1 REMOVING ASSET HARDWARE

N/A

4.5.2 REMOVING OPERATING SYSTEM

N/A

4.5.3 REMOVING ASSET SOFTWARE DEPENDENCIES

N/A

4.5.4 REMOVING ASSET PROGRAMMING LANGUAGES

- typescript 5.0.2

4.5.5 REMOVING ASSET DATA STRUCTURES

N/A

4.5.6 REMOVING ASSET DATA PROCESSING

The user chooses items they wish to remove from their inventory by checking a box and clicking on the delete item button. The MongoDB command to remove the items is executed and the items are removed from the database.

4.6 CHECKOUT ASSET

This subsystem will mark the asset as "checked out". This signifies that the asset is currently being used and is not in its designated storage location.

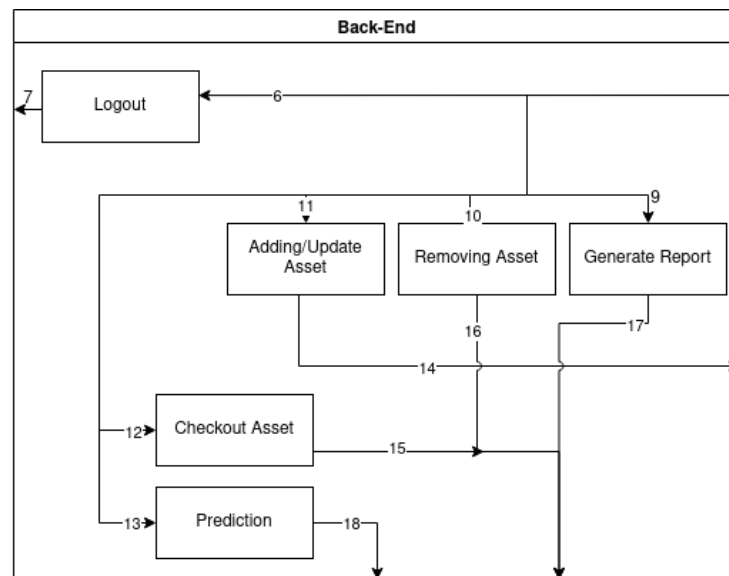


Figure 8: Back-end diagram

4.6.1 CHECKOUT ASSET HARDWARE

N/A

4.6.2 CHECKOUT OPERATING SYSTEM

N/A

4.6.3 CHECKOUT ASSET SOFTWARE DEPENDENCIES

N/A

4.6.4 CHECKOUT ASSET PROGRAMMING LANGUAGES

- typescript 5.0.2

4.6.5 CHECKOUT ASSET DATA STRUCTURES

N/A

4.6.6 CHECKOUT ASSET DATA PROCESSING

Changes the Boolean value of the checkout field for the asset in the MongoDB database.

4.7 LOGOUT

This subsystem will send the user to the Landing Page and will require them to log in in order to return to the dashboard.

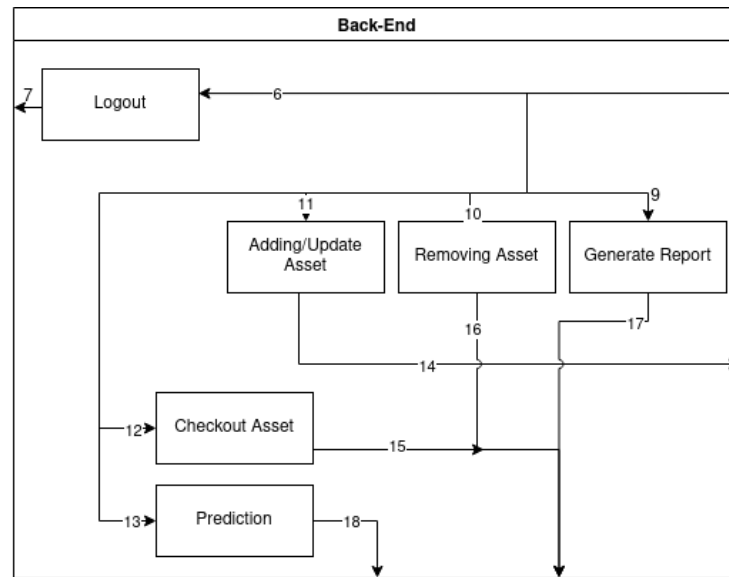


Figure 9: Back-end diagram

4.7.1 LOGOUT HARDWARE

N/A

4.7.2 LOGOUT OPERATING SYSTEM

N/A

4.7.3 LOGOUT SOFTWARE DEPENDENCIES

N/A

4.7.4 LOGOUT PROGRAMMING LANGUAGES

- typescript 5.0.2

4.7.5 LOGOUT DATA STRUCTURES

N/A

4.7.6 LOGOUT DATA PROCESSING

Remove the Web Token that allows the user to enter the dashboard without logging in.

4.8 PREDICTION

The subsystem will send a request to the droplet hosting the Price Prediction Model and will display the predicted price of the asset.

4.8.1 PREDICTION HARDWARE

N/A

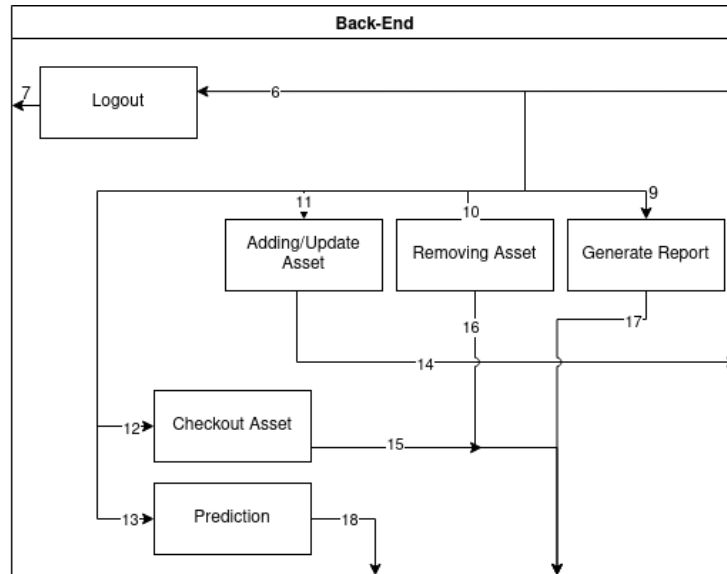


Figure 10: Back-end diagram

4.8.2 REMOVING OPERATING SYSTEM

N/A

4.8.3 PREDICTION SOFTWARE DEPENDENCIES

N/A

4.8.4 PREDICTION PROGRAMMING LANGUAGES

- typescript 5.0.2

4.8.5 PREDICTION DATA STRUCTURES

N/A

4.8.6 PREDICTION DATA PROCESSING

Send a list of the asset attributes to the Price Prediction Layer.

4.9 GENERATE REPORT

The responsibility of the generate report sub-layer is to handle the generation of a pdf report of selected items.

4.10 LAYER HARDWARE

MongoDB server which houses the user's saved items

4.11 LAYER OPERATING SYSTEM

No operating system is needed only a web browser.

4.12 LAYER SOFTWARE DEPENDENCIES

A description of any software dependencies (libraries, frameworks, etc) required by the layer.

4.13 GENERATE REPORT

This subsystem is responsible for generating a pdf report of checked items from the table. The report will contain the name of the item, the category, the description, and the serial number of the item.

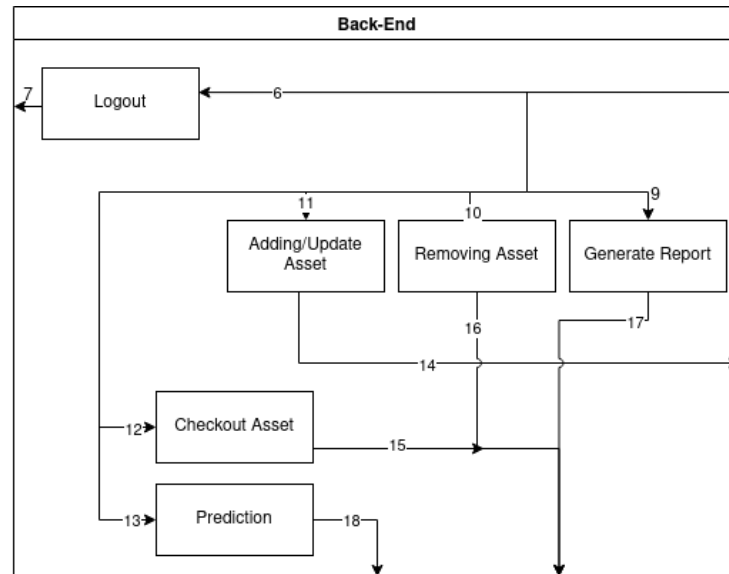


Figure 11: Back-end diagram

4.13.1 SUBSYSTEM HARDWARE

N/A

4.13.2 SUBSYSTEM OPERATING SYSTEM

N/A

4.13.3 SUBSYSTEM SOFTWARE DEPENDENCIES

N/A

4.13.4 SUBSYSTEM PROGRAMMING LANGUAGES

- Typescript 5.0.2

4.13.5 SUBSYSTEM DATA STRUCTURES

There are no data structures when exporting the items to a pdf.

4.13.6 SUBSYSTEM DATA PROCESSING

User clicks on a box button on the items that they wish to export to a pdf file.

5 DATABASE LAYER SUBSYSTEMS

Holds the data for each company, users within the company, and tables of data. Users can look up inventory, remove inventory, or generate reports with the data contained in this layer.

5.1 DATABASE HARDWARE

A basic Digital Ocean Droplet with 4 Intel vCPUs, 8 GB RAM, and 160 GB disk hard drive.

5.2 DATABASE OPERATING SYSTEM

Ubuntu 22.10 x64

5.3 DATABASE SOFTWARE DEPENDENCIES

- MongoDB 5.0.2
- mongoose 5.0.2
- npm 7.0.3
- node 20.4.0

5.4 COMPANY

Holds the data for each company, users within the company, and tables of data. Users can look up inventory, remove inventory, or generate reports with the data contained in this layer.

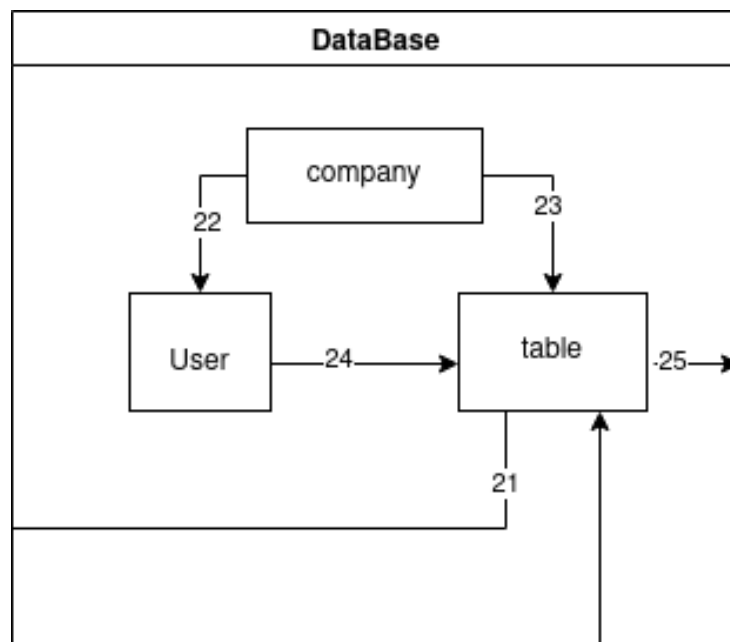


Figure 12: Database diagram

5.4.1 COMPANY HARDWARE

N/A

5.4.2 COMPANY OPERATING SYSTEM

N/A

5.4.3 COMPANY SOFTWARE DEPENDENCIES

N/A

5.4.4 COMPANY PROGRAMMING LANGUAGES

- typescript 5.0.2

5.4.5 COMPANY DATA STRUCTURES

The company is stored as an object and then put into the database.

5.4.6 COMPANY DATA PROCESSING

The company will sign up on the website by creating a username and password.

5.5 USER

Holds the data for each company, users within the company, and tables of data. Users can look up inventory, remove inventory, or generate reports with the data contained in this layer.

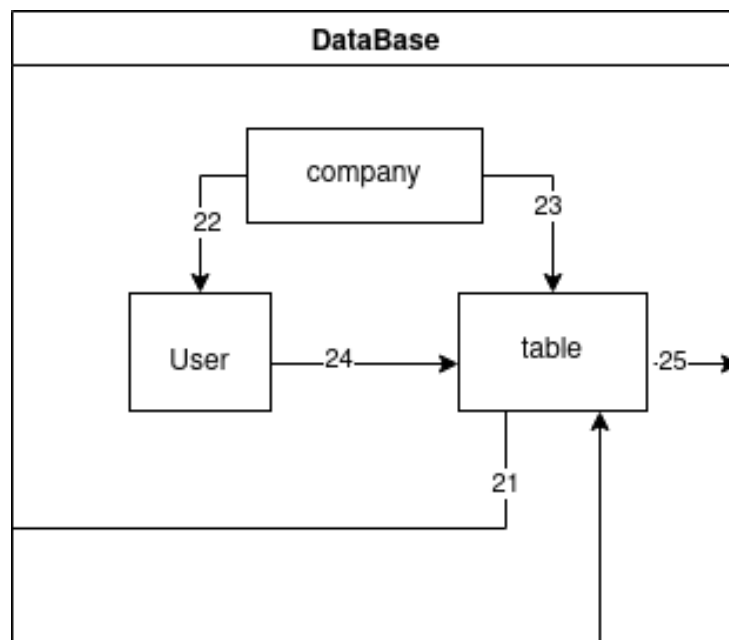


Figure 13: Database diagram

5.5.1 USER HARDWARE

N/A

5.5.2 USER OPERATING SYSTEM

N/A

5.5.3 USER SOFTWARE DEPENDENCIES

N/A

5.5.4 USER PROGRAMMING LANGUAGES

- typescript 5.0.2

5.5.5 USER DATA STRUCTURES

The user is stored as an object and then put into the database that is assigned to the company the user is with.

5.5.6 COMPANY DATA PROCESSING

When the user is being created the user name is the email and they pick a unique password. Both values go into the appropriate text fields.

5.6 TABLE

The table subsystem is responsible for holding the items the user puts into the appropriate data fields. There are also tables for the company's login credentials, the users associated with each company, and the user's login credentials.

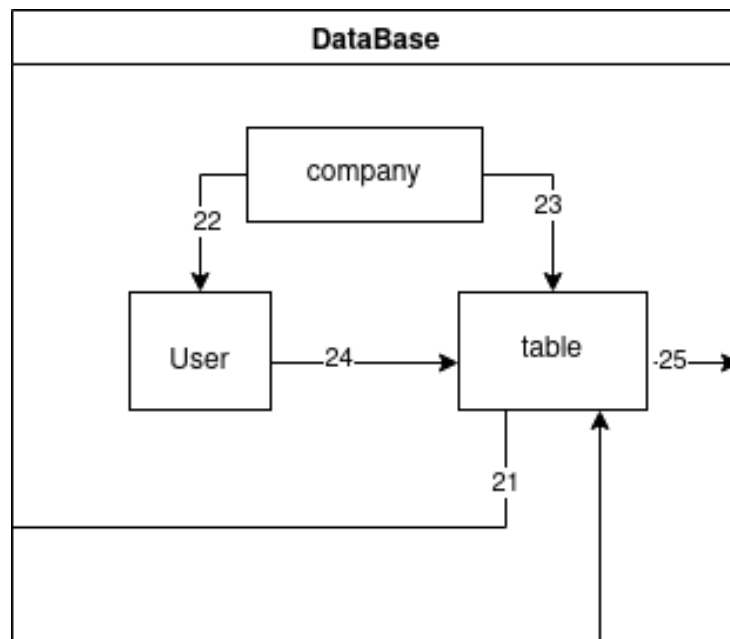


Figure 14: Database diagram

5.6.1 TABLE HARDWARE

N/A

5.6.2 TABLE OPERATING SYSTEM

N/A

5.6.3 TABLE SOFTWARE DEPENDENCIES

N/A

5.6.4 TABLE PROGRAMMING LANGUAGES

- typescript 5.0.2

5.6.5 TABLE DATA STRUCTURES

There is a predefined schema for each table.

5.6.6 TABLE DATA PROCESSING

The user fills out the appropriate data fields and either adds the item, creates a new company account, or creates a new user.

6 ENCRYPTION LAYER SUBSYSTEMS

The encryption layer is where the subsystems that encrypt and decrypt the data exist. If certain data fields are subject to encryption and decryption they will pass through this layer.

6.1 ENCRYPTION HARDWARE

A basic Digital Ocean Droplet with 4 Intel vCPUs, 8 GB RAM, and 160 GB disk hard drive.

6.2 ENCRYPTION OPERATING SYSTEM

Ubuntu 22.10 x64

6.3 ENCRYPTION SOFTWARE DEPENDENCIES

- mongoose 7.0.3
- mongoose-field-encryption 6.1.0
- npm 9.7.2
- node 20.4.0

6.4 ENCRYPT DATA

This subsystem is responsible for encrypting data as needed

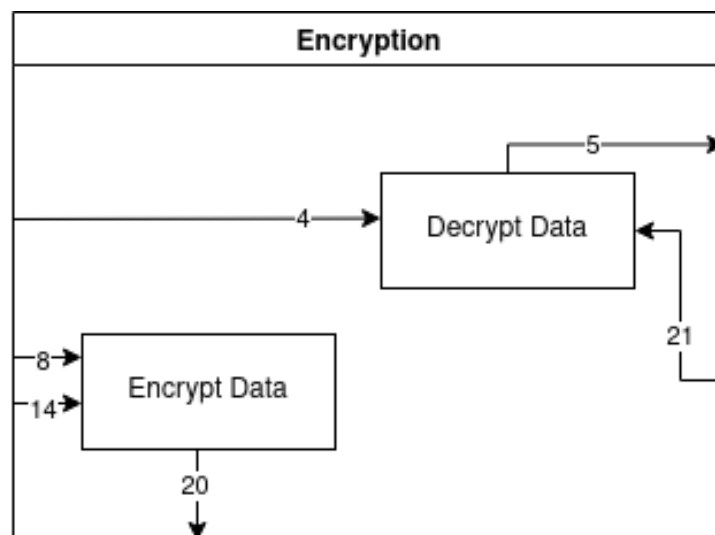


Figure 15: Encryption diagram

6.4.1 ENCRYPT DATA HARDWARE

N/A

6.4.2 ENCRYPT DATA OPERATING SYSTEM

N/A

6.4.3 ENCRYPT DATA SOFTWARE DEPENDENCIES

N/A

6.4.4 ENCRYPT DATA PROGRAMMING LANGUAGES

- typescript 5.0.2

6.4.5 ENCRYPT DATA DATA STRUCTURES

Data to be encrypted is stored as a string or object that will be processed by a saltGenerator function.

6.4.6 ENCRYPT DATA DATA PROCESSING

The objects or other types are converted to strings using JSON.stringify. The AES-256-CBC algorithm is used to encrypt the data. The fields to be encrypted are encrypted with a secret and the random salt function.

6.5 DECRYPT DATA

1) This subsystem is responsible for decrypting the encrypted data so the user can read the data and not the random string.

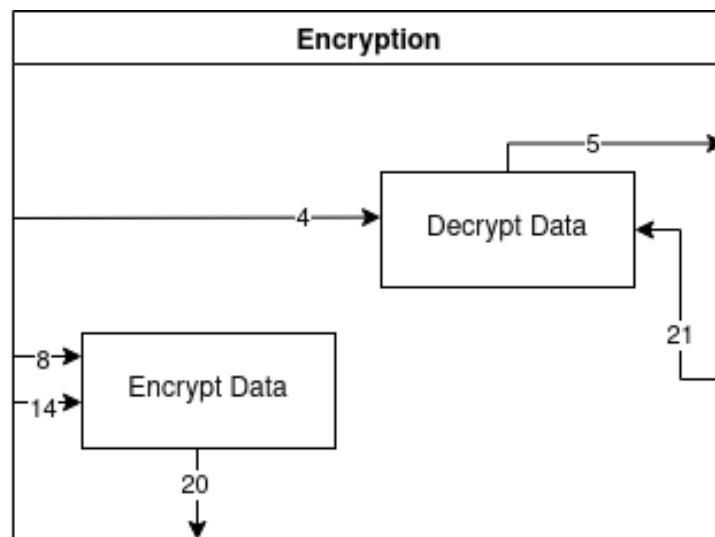


Figure 16: Encryption diagram

6.5.1 ENCRYPT DATA HARDWARE

N/A

6.5.2 ENCRYPT DATA OPERATING SYSTEM

N/A

6.5.3 ENCRYPT DATA SOFTWARE DEPENDENCIES

N/A

6.5.4 ENCRYPT DATA PROGRAMMING LANGUAGES

- typescript 5.0.2

6.5.5 ENCRYPT DATA DATA STRUCTURES

The data is held in an encrypted string called cyphertext.

6.5.6 ENCRYPT DATA DATA PROCESSING

The AES-256-CBC algorithm that mongoose-field-level encryption uses decrypts the string.

7 PRICE PREDICTION LAYER SUBSYSTEMS

The back-end layer is responsible for allowing users to input, update, or remove assets from the database as needed. Users will also be able to obtain reports on their current inventory status. The price prediction will be sent to and from the back end as needed. The back end will also handle logging out of the system.

7.1 PRICE PREDICTION HARDWARE

A basic Digital Ocean Droplet with 4 Intel vCPUs, 8 GB RAM, and 160 GB disk hard drive.

7.2 PRICE PREDICTION OPERATING SYSTEM

Ubuntu 22.10 x64

7.3 PRICE PREDICTION SOFTWARE DEPENDENCIES

- mongoose 7.0.3
- next 13.2.4
- npm 9.7.2
- node 20.4.0

7.4 PRICING ALGORITHM

Adding/updating asset Responsible for allowing the user to add or update an item. The user fills out the data fields and the data is stored as an object and sent to the appropriate MongoDB server.

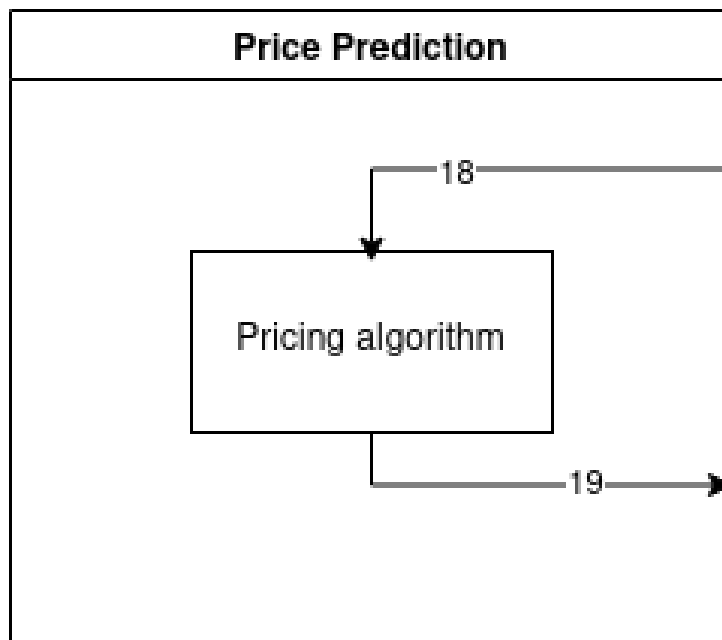


Figure 17: Price Prediction diagram

7.4.1 PRICING ALGORITHM HARDWARE

N/A

7.4.2 PRICING ALGORITHM OPERATING SYSTEM

N/A

7.4.3 PRICING ALGORITHM SOFTWARE DEPENDENCIES

N/A

7.4.4 PRICING ALGORITHM PROGRAMMING LANGUAGES

- typescript 5.0.2

7.4.5 PRICING ALGORITHM DATA STRUCTURES

The item being added or updated is stored as an object with all the fields set to values that were determined by the user.

7.4.6 PRICING ALGORITHM DATA PROCESSING

The user has elected to add or update an item. They insert the appropriate data into the data fields and click on add item.

REFERENCES