2022-09-14

Web-based POS and inventory management system for SMEs, retail stores, and restaurants

Choudhury, Tashfeen Mustafa

Independent University, Bangladesh

Web-based POS and inventory management system for SMEs, retail stores, and restaurants

By

Tashfeen Mustafa Choudhury

Student ID: 1821069

Supervisor:

Mr. Sanzar Adnan Alam

Lecturer

Department of Computer Science and Engineering

Independent University, Bangladesh

September 14, 2022

Final Submission

Department of Computer Science & Engineering Independent University, Bangladesh

Attestation

I Tashfeen Mustafa Choudhury bearing ID-1821069 state that this report is completely made by myself under the supervision of my respected faculty member Sanzar Adnan Alam. I have completed and submitted this document for the partial fulfilment of the requirement for the Degree of Computer Science from Independent University, Bangladesh (IUB). All the works here are the result of my work experience in YogiPrice and study from different blogs, tutorials, research papers etc that are referenced in the report.

Tashfeen — 14 September 2022

Signature — Date

Tashfeen Mustafa Choudhury

Name

Acknowledgment

First and foremost, I am grateful to Almighty Allah as I am able to complete the report within due time.

I am extremely grateful to the COO Mr. Ishtiaque Ahmed Sazid, YogiPrice as I was given an opportunity to work with their excellent development team as an intern. They have helped me to carry on with my work in a professional environment. This report on my work could not have been possible without the help and guidance of my respected supervisor Sanzar Adnan Alam. I would also like to address my parents support and prayers for me that helped me to continue with my work. Finally I must express my cordial gratitude to all my faculty members to help me carry on with the whole bachelors degree.

Letter of Transmittal

September 14, 2022
Sanzar Adnan Alam
Lecturer,
Department of Computer Science and Engineering,
Independent University, Bangladesh

Subject: Submission of Internship report.

Dear Sir, With due respect I would like to state that I am submitting my Internship Report to you. I started working with YogiPrice for a 3 month long period of time as a software development intern. I would like to inform you that I have worked on a ”Web-based POS system” for YogiPrice and made this report based on my work experience and under your supervision. I would be ever grateful to you if you would kindly evaluate my report. I therefore pray and hope that your honor would be kind enough to grant my application and oblige thereby.

Yours Sincerely,
Tashfeen Mustafa Choudhury
ID: 1821069

Evaluation Committee

Signature 

SANZAR ADNAN ALAM

Name

Supervisor

Signature 

Name AJMIRI SABRINA KHAN

Internal Examiner

Signature 

Name ISHTIAQUE AHMED SAZID

External Examiner

Signature 

Name MAHADY HASAN, PhD

Convener

Abstract

This project concerns playing a part in building a web-based POS and inventory management system for SMEs, retail stores, and restaurants and allowing them to record their everyday sales as well as manage inventory using our product. Through our system, we will also produce daily, monthly, weekly, and yearly sales and revenue reports for businesses that use our product.

Current software providers in this market mostly provide native software by installing the software manually on the computer of each branch of a business, which causes a lot of time in updating and adding features or changes to the software and uses a lot of manual hours and resources in making those changes to all the branches or outlets of a business. This can cause problems with the waste of resources, time, and efficiency which in turn can affect the business or user of the software in terms of money lost. Current options in this market also do not provide encapsulation of important and confidential business information from the hands of employees of the business or the employees of the software providers, which could also affect the business in the future.

With our solution, we want to provide a safer, customizable, and easy to use solution to businesses that depend on POS and inventory management systems at a cheaper or the same cost as the alternative solutions currently in the market.

Contents

Chapter 1

Problem Statement

This project report focuses on my work as an intern at a reputable software firm called “YogiPrice” where I work as a software developer. In this firm, I have been tasked with creating a web-based POS (point-of-sale) application built using the MERN (MongoDB, ExpressJS, ReactJS, and Node.js) stack.

This report elaborates on the work that I have completed under YogiPrice as my internship for the course CSE499 during the semester of Summer 2022 under the Department of Computer Science at Independent University, Bangladesh (IUB). The internship will continue for 3 months. As a result, this report will consist of an introduction to the project that I have been assigned from my workplace and related works and software currently existing locally.

1.1 Overview/Background of the Project

With unprecedented amounts of offline activity now moving to online in the world of business in Bangladesh, an increase in customer consumption which equates to large amounts of data, as well as large volumes of sales to manage through offline outlets for various types of businesses, more and more companies are looking for robust, easy-to-use software to manage their offline outlet sales. In other words, these companies are looking for highly dependable, accessible, scalable, and modifiable POS systems for their “points-of-sales”.

As a result, my company has surveyed the market and identified this gap in the market and has decided to build a POS system that also allows for inventory management alongside the management of staff, product catalogs for each point of sales, and various other features that allow for the robust day-to-day management of sales at their outlets. At YogiPrice, I have been assigned with the full-stack development of this software along with other responsibilities of a software engineer such as requirements analysis, system design, and analysis.

1.2 Objectives

The objective of this project is to build a robust POS system that is manageable, accessible, scalable, modifiable, and easy to use. This POS system will have the following features:

1. Provide business owners with to integration of the software with various business locations.
2. Ability to create and assign different user roles to various employees in different hierarchies in the company to allow efficient management of business processes.
3. Offer one place to manage their inventory needs for all branches or points-of-sales.
4. Record day-to-day sales/orders from their inventory from each branch.
5. Categorize sales/orders under different categories such as walk-in sales, and online sales (FoodPanda, Pathao, etc.)
6. Easy calculation to record each sale.
7. Ability to edit, add, manage, and remove various products.
8. Ability to edit, add, manage, and remove various marketing campaigns such as discounts, offers, etc.
9. Generate and analyze various daily, weekly, monthly, and yearly sales data and reports.
10. A secured platform that has all data of the businesses confidential from unwanted parties and follows proper user roles for each stakeholder involved within a business using this platform.
11. The system must also have a payment system for users who want to use the premium software or upgrade based on their needs.

1.3 Scope

As you can probably that the scope of the work is quite large. The goal of the project is to make a generic system that can apply to multiple specific types of businesses such as car dealerships, restaurants, online sellers, etc. All of the sales data generated for a business from each point-of-sale must be kept hidden from irrelevant stakeholders of that business and also secured and confidential from other actors. The system must also keep a record of the products sold from each branch, campaigns going on under each branch, sales that happen through those branches, the types of different sales that occur each day, as well as the ability to modify these specific details such as edit or completely remove them.

Chapter 2

Literature Review & Methodology

This project, named “POSman”, aims to revolutionize the current POS-system market in Dhaka, Bangladesh. POS-system, or Point-of-Sale systems, are usually bought by business owners and are typically installed or used by various branches or points-of-sale of the business. This means that each branch of a business must be able to use the POS system through their respective machines (phones or PCs). This also means that data must be shared and reconciled between each branch. POS systems or the benefits of a POS system are also used by various stakeholders and people involved in the business.

Currently, there are various problems with the POS systems that are provided by software vendors in the market. Most vendors sell their POS system either as a native app or as a web-based solution.

The problems with current native app solutions:

1. Native apps provided by the vendors require representatives to physically access the machines of each branch of a business and install the software. This can cause problems with customers/users starting to use the product without any hassle.
2. Changes to the software such as adding/removing products, offers, discounts, and sales channels require native app vendors to have to remotely access the machines of a branch of a business and then make the specific changes using AnyDesk or other remote access software. This can cause massive time delays in changing necessary business rules, processes, and information relating to everyday sales from their points of sale.
3. There is currently no way of specifying user roles for an organization/business using the software. Typically, only one account is created for a single native app installed in every branch. So other stakeholders involved in the business do not have access to the necessary business and sales data and reports. For example, reports and data generated in the native app cannot be accessed by head office or business owners and needs to be photographed and sent separately. This causes a delay in communication and transfer of important data for business needs. It also makes the sales and business data exposed to unwanted parties.
4. Current vendors don’t provide a safe and secure way to store their sales data on their systems and connected databases. This data is viewable by representatives of the software vendors and may fall into the wrong hands.
5. There is no inventory management system internally available in any vendor’s POS system and typically is custom-made or purchased separately. Most

businesses just use Excel and phone/email communication to keep inventory data in sync.

The problems with current web-based solutions:

Not many web-based solutions.

1. Web-based solutions are not readily accessible and require contact with vendors first.
2. Vendors make POS systems using e-commerce or CMS-based websites that already contain a generic POS system. This solution does not always factor in the individual needs of the customers.

2.1 Relationship with Undergraduate Studies

The following courses have helped me gain the necessary skills and understanding of software engineering and development that has given me the confidence to take on this project and be able to complete it. The courses from the Department of Computer Science that have helped me along during my degree are listed below.

2.1.1 Introduction to Basic Programming

This course taught me the basics of programming which includes data types, operators, functions, and the elementary concepts of computer science. It also introduced me to the concepts of problem-solving, algorithm design, and program construction.

2.1.2 Object-Oriented Programming

This course helped me understand the basics of design patterns and principles, how to abstract and encapsulate code, and understand in-depth how classes, objects, ADTs, composition, inheritance, polymorphism, and other concepts of OOP.

2.1.3 Database Management

The Database Management course allowed me to learn more about the development process, database architectural process, an in-depth understanding and practical experience with SQL as well as the use of database management systems.

2.1.4 System Analysis and Design

This course taught me various techniques used to design and conduct analysis of information systems, create models, learn project management, create data flow diagrams, UML, Object-Oriented Analysis, Feasibility analysis, etc.

2.1.5 Web Applications and Internet

This course helped me learn about HTML, CSS, & JavaScript and the basics of how the Internet works, and how modern web-based applications are built.

2.1.6 Software Engineering

This project-based course allowed me to learn first-hand how software is engineered from top to bottom, starting from the requirement analysis, project management, project and process metrics, risk analysis, configuration management, software quality and testing, client-server architecture, and web engineering.

2.1.7 Data Structure

This course helped me learn about how data can be stored and arranged and manipulated in cost-effective ways and which types of data structures to use for which scenarios.

2.2 Related works

While there are plenty of POS software applications around, many different ongoing projects by different organizations, and various research papers written on this topic however we will address only a select few.

- **3s-bd:** They are the leading providers of POS software to most shops and restaurants in Bangladesh. Currently, their software is a native app that has no security, no user roles for different stakeholders, and is hard to use and install. Their software, due to being various versions of a native app installed on a single computer is also hard to update and edit so it makes it hard, time-consuming, and cost-intensive for businesses, especially small businesses that are growing at a rapid pace in today's Bangladesh to grow and make necessary changes according to their growing and varying businesses process. Such as removing a simple item or discount code from the system requires contacting one of their representatives, scheduling a call, using AnyDesk to access the computer where the software is installed by the representative, and then making those changes.
- **MediaSoft:** MediaSoft is a software provider located in Bangladesh that provides a large array of software systems and POS systems for retail stores, and restaurants, however, their software is also a native solution.
- **Smart Software Ltd.**: This is another software provider that provides a POS system with a large array of features such as sales management, smart dokani, product management, etc. This is a web-based solution.
- **Feasibility Study of Web-Based POS System for SMEs in Bangladesh:** This research paper shows that there is a large number of SMEs in Bangladesh that are yet to properly utilize the benefits of a web-based POS system and that such a project is also feasible given the current climate.[1]

While other companies are providing POS systems as software and there are ongoing projects as well, most still have the problems that have been outlined above under the Literature Review intro section.

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

We are going to break down the development of this using a Work Breakdown Structure (WBS) that will allow us to divide a large and complex project into smaller portions to be able to complete tasks in an efficient, effective, and manageable way.

These smaller segments can then be assigned to specific teams and will help ensure collaboration, accountability, and productivity between different teams. This will also allow for efficient project management.

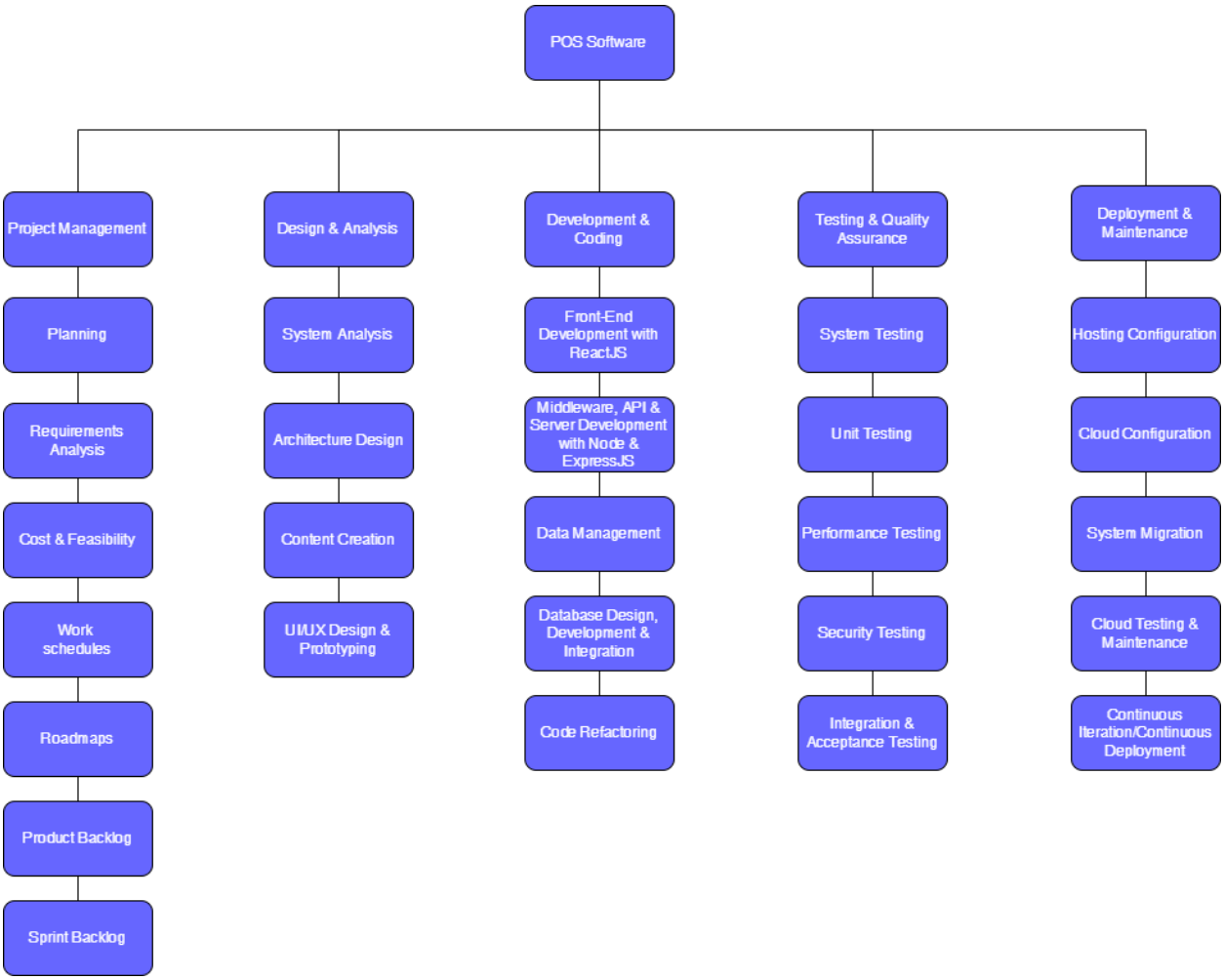

Figure 1 Work Breakdown Structure

As we can see this provides a clear picture of which teams are assigned to specific tasks as well as provides a visual and hierarchical view of the project and the scope of the project.

3.2 Process/Activity-wise Time Distribution

Our Process/Activity wise Time Distribution is as follows:

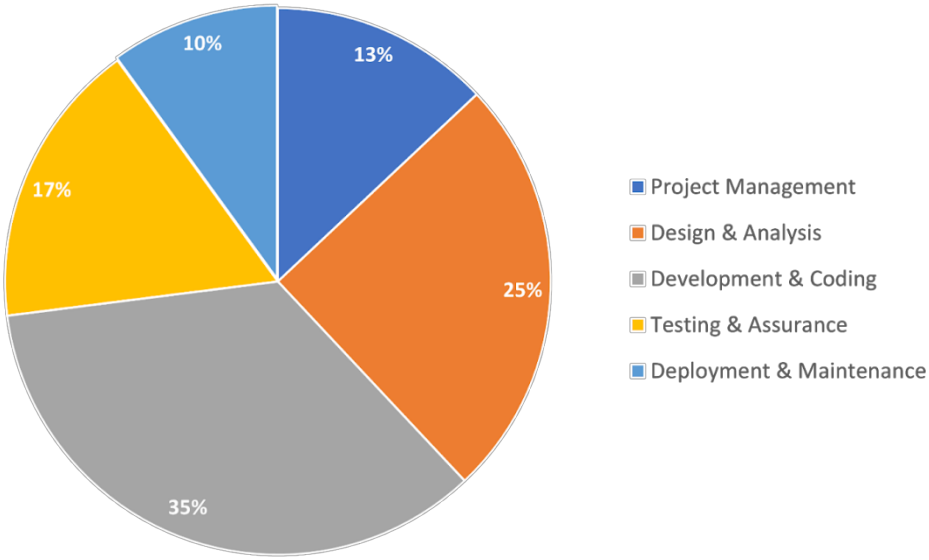

Figure 2 Process/Activity-wise Time Distribution

This is also provided in a table that shows each task, duration, and its dependencies.

Task	Description	Duration (in weeks)	Dependency
1	Requirement Analysis	2 weeks	None
2	Planning & Project Management	1 week	1
3	System Analysis & Design	2 weeks	2
4	UI/UX Design	2 week	3
5	Front-end Development	2 weeks	1,2,3,4
6	Back-end (server, API, & middleware) development	2 weeks	3,5
7	Debugging & Code Refactoring	1 week	5
8	Database Design	1 week	1,3
9	Front-end, Back-end, & Database Integration	1 week	5,6,7
10	Testing & Quality Assurance	2 weeks	5,6,7,8,9

11	Deployment & Maintenance	1 week	10
12	Initial Launch & Live Testing with test users	2 weeks	11
13	Continuous Integration & Deployment	N/A	12

Table 1. Breakdown of Process/Activity-wise Time Distribution

Here, I have a pie chart of how our process/activity wise time spent on the project is distributed as per the time taken for each individual process and the total prospective time for the total project to finish.

3.3 Gantt Chart

Gantt Chart is one of the most used projects management charts. It is used to draw out specific processes and tasks as per the time it will take to complete in weeks. It can also show a breakdown of subtasks that may look into how many days it will take to finish and the people assigned to it. In some cases, Gantt Charts may also contain the dependencies of the processes. A Gantt Chart is useful for monitoring specific tasks and resources, scheduling, and managing different teams in a project.

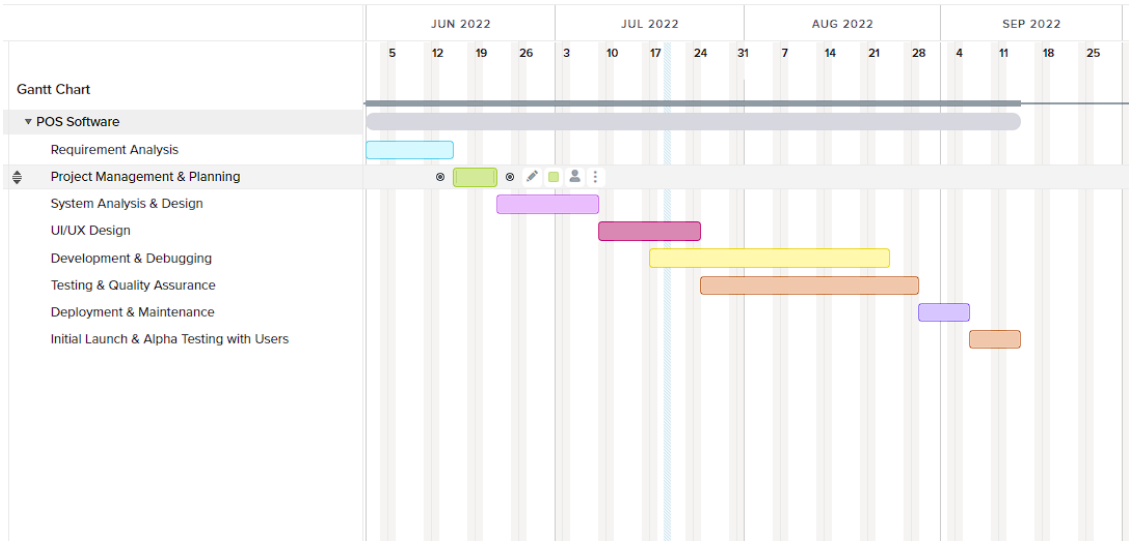

Figure 3 Gantt Chart

In the image above, we can see the entire Software Development Life Cycle and each of the phases the project will go through from start to finish. We use colors to differentiate between each task and also recognize that some tasks such as UI/UX Design, Development and Testing and Quality Assurance will overlap with each other as we are following a scrum methodology.

Chapter 4

Methodology and Software Development Lifecycle

The Software Development Life Cycle for the project must be implemented with a certain methodology. Some examples are Waterfall, Unified model, V Model, Spiral model, Agile etc. We have chosen the Agile methodology to undergo the SDLC process of our project. The methodology that we followed was the agile methodology.

4.1 Agile methodology

Agile Methodology is a people-focused, results-focused approach to software development that respects our rapidly changing world. It's centered around adaptive planning, self-organization, and short delivery times. It's flexible, fast, and aims for continuous improvements in quality, using tools like Scrum and eXtreme Programming. Agile revolutionized the tech industry in the late 20th and early 21st century and has now become trendy in other sectors as companies seek to become more versatile and adaptable.

It abandons the risk of spending months or years on a process that ultimately fails because of some small mistake in an early phase. It relies instead on trusting employees and teams to work directly with customers to understand the goals and provide solutions in a fast and incremental way.

Some agile values and principles consist of:

1. Build your projects and teams around the individuals, choosing people who can communicate and work together well.
2. Provide the team with regular intervals to reflect on their progress and figure out how to fine-tune their workflow to produce the best results.
3. You'll save time if you're constantly designing, testing, and improving software rather than writing out detailed documentation of what the software will do.
4. Set a shorter time scale for delivering working software (weeks rather than months) and use that product as your primary measure of progress on your project.

Chapter 5

Body of the Project

5.1 Requirements Analysis

Rich Picture Analysis

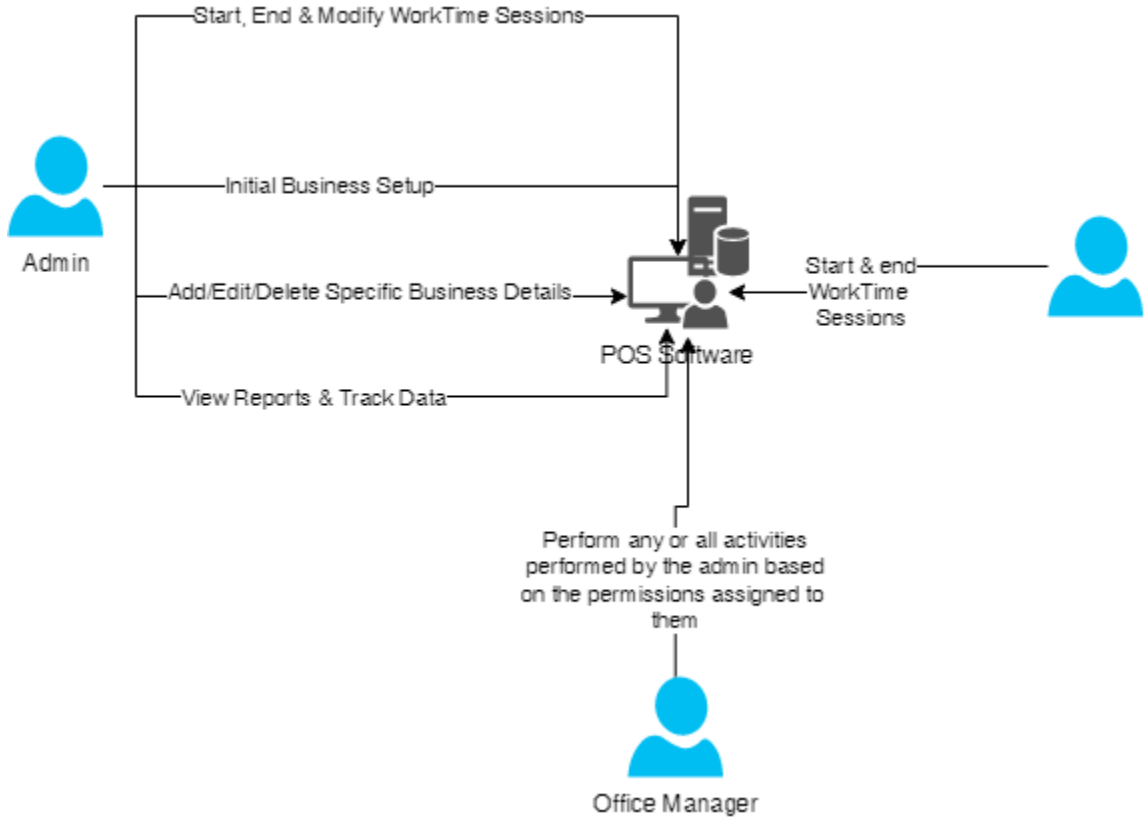

Figure 4 Rich Picture Diagram of the Proposed System

Six Elements Analysis

Process	**System Roles**					
	Human	**Non-comp hardware**	**Comp hardware**	**Software**	**Database**	**Network & comm.**
Signup/Login	Admin, Office Manager and Branch Manager can login	N/A	Computer	ReactJS, NodeJS, ExpressJS and MySQL	MySQL	Internet

Initial Business Setup	Admin	N/A	Computer	ReactJS, NodeJS, ExpressJS and MySQL	MySQL	Internet
Add/Edit/Delete Business Details	Admin, Office Manager and Branch Manager	N/A	Computer	ReactJS, NodeJS, ExpressJS and MySQL	MySQL	Internet
Create & Modify WorkTime Sessions	Admin, Office Manager and Branch Manager	N/A	Computer	ReactJS, NodeJS, ExpressJS and MySQL	MySQL	Internet
View Reports & Track Data	Admin, Office Manager and Branch Manager	N/A	Computer	ReactJS, NodeJS, ExpressJS and MySQL	MySQL	Internet

Table 2. Six Elements Analysis

Business Process Model

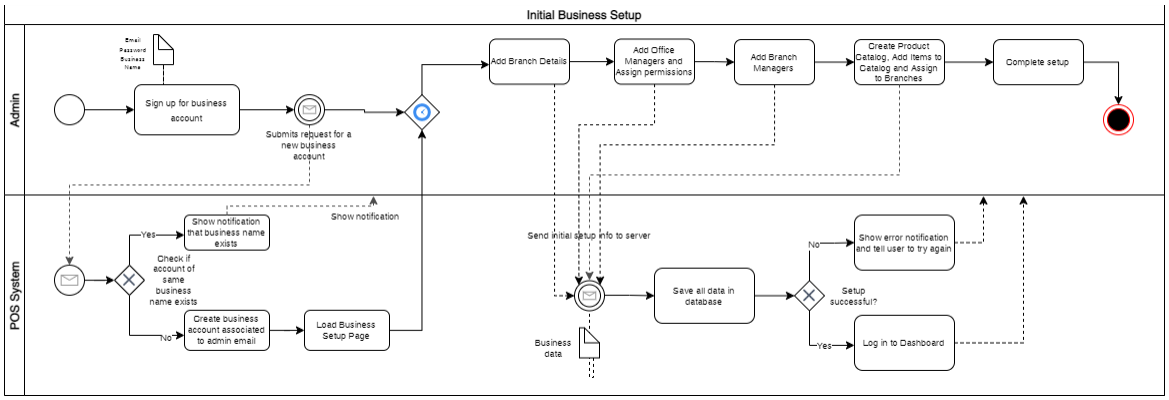

Figure 5 Initial Business Setup

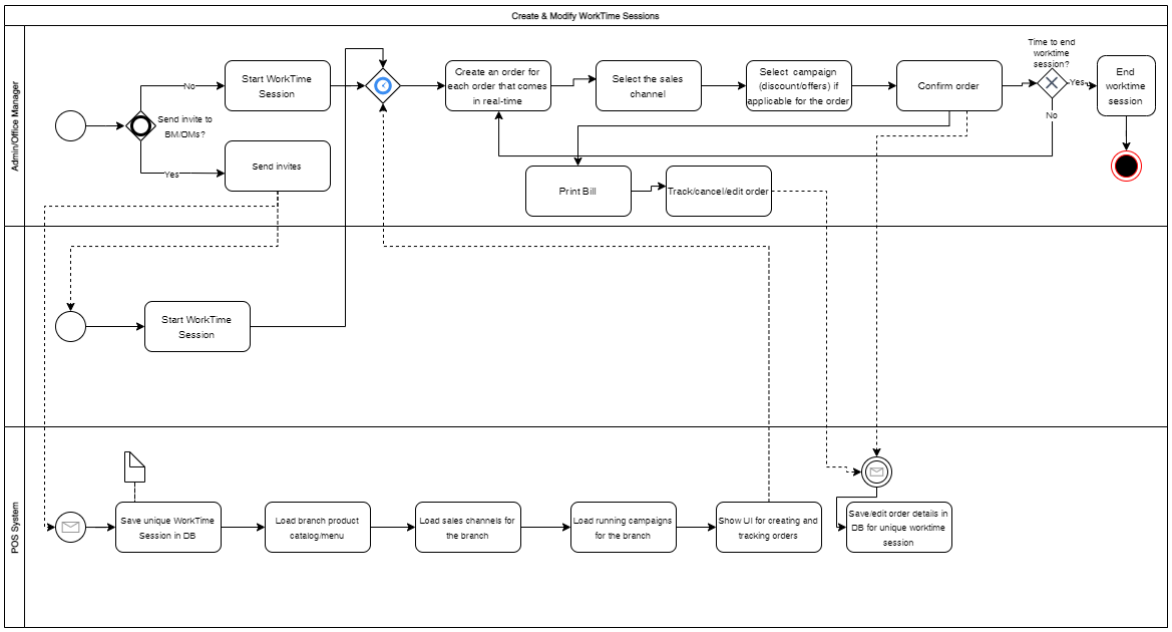

Figure 6 Create and Modify WorkTime Session

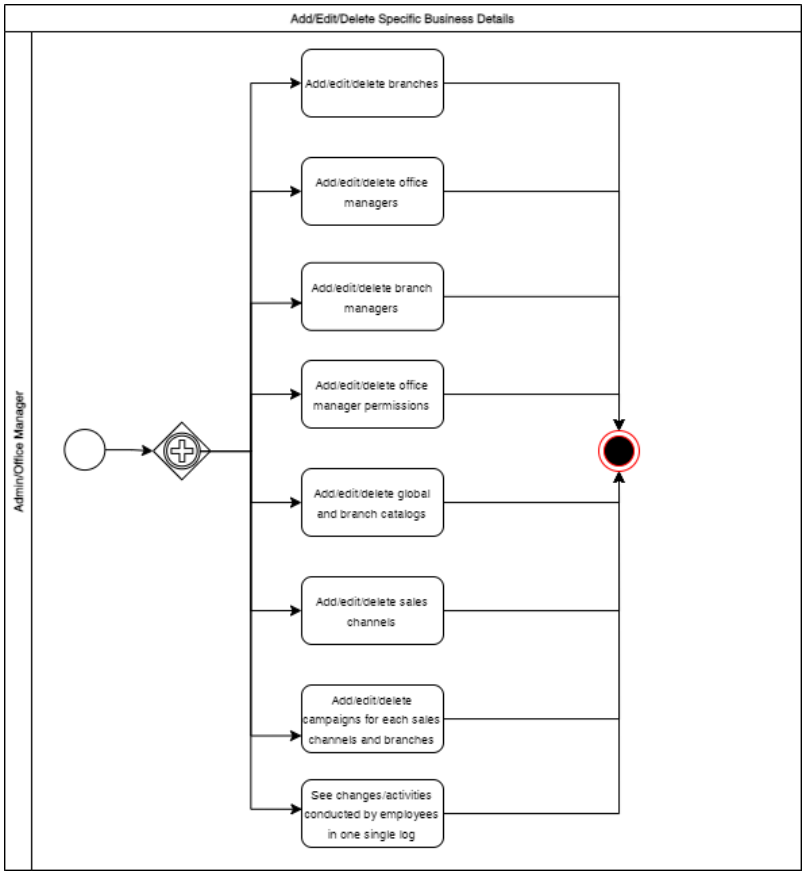

Figure 7 Add,Delete,Edit Specific Business Details

5.2 Data Modeling

Business Rules

Admins/Business Owners

Initial Business Setup

This is done only by the admin/owner of a particular business.

- Admin creates a business account using their name and email. Inputs business information such as number & names of branches.
- Sends office and branch manager role invites to relevant business employees with relevant permissions.
- Creates product catalog, adds product items to the catalog, and assigns parts of the product catalog to each branch.

Add/Edit/Delete Specific Business Details

- The users can create/edit/delete branch details.
- The users can create/edit/delete or revert role permissions of office managers.
- The users can create/edit/delete categories and items to the product catalog and assign the product catalog to one or multiple branches as a branch catalog. There they can also separately edit specific branch catalogs.
- The users can create/edit/delete sales channels (e.g. Foodpanda, Pathao, Walk-in/Dine-in, etc.) and assign sales channels for each branch.
- The users can create/edit/delete campaigns (discounts and offers) for each sales channel.

Create & Modify Worktime Sessions

- The users can create, end, view, edit, and delete work sessions during everyday business operations at their points-of-sale.
- The users from a branch will be shown products from their specific branch catalog.
- The users will be able to create orders and assign items to the order and apply the sales channels.
- The system will show the total cost of all the items for the order. If sales channels contain any discounts or offers that will be applied to the total cost.
- Orders can be created, canceled, initiated, and completed. At initiation, a bill is printed if applicable.

View Reports & Track Data

- Admins can see a table where they can see orders come in real-time.
- Admin can see real-time analytics (e.g. sales, orders) from each branch.
- Generate reports e.g. daily, weekly, monthly, and yearly sales, sales of items, sales through sales channels, sales of different product catalogs, sales from each branch, and overall all branches.
- Admins can assign office managers to see real-time analytics and generate reports.

Office Managers

Office Managers are managers/individuals who are in charge of multiple points of sales or report directly to the admins/business owners. As such, the business owner can assign or unassign specific privileges. Office Managers will be able to perform all the activities mentioned by an admin given the right permissions and privileges.

Branch Managers

Create & End WorkTime Sessions.

Branch Managers are managers/individuals in charge of a branch. They will only be able to create and end WorkTime sessions

- The users can create, end, view, edit, and delete work sessions during everyday business operations at their points-of-sale.
- The users from a branch will be shown products from their specific branch catalog.
- The users will be able to create orders and assign items to the order and apply the sales channels.
- The system will show the total cost of all the items for the order. If sales channels contain any discounts or offers that will be applied to the total cost.
- Orders can be created, canceled, initiated, and completed. At initiation, a bill is printed if applicable. Upon completion, a notification is sent to the user.

Data Definitions

This is an exhaustive list of all the entity types that will exist in the POS software database

Businesses

Contained in a separate database for each business that signs up for a business account
id (integer): unique numeric id to identify each business
name(string): name of the business account; maximum 200 characters

The following entities will be contained in separate databases made for each business that signs up for a business account.

Business Admin

id (integer): uniquely identifies the admin of a business account
first_name (string): first name of the admin; maximum 50 characters
last_name (string): last name of the admin; maximum 50 characters
password (use bcrypt algorithm, binary 60): self-descriptive;
email (string): email address of the admin
phone (string): phone number of the admin; only accepting local numbers

Office Manager

id (integer): uniquely identifies the office manager of a business account
first_name (string): first name of the manager; maximum 50 characters
last_name (string): last name of the manager; maximum 50 characters
email (string): email address of the office manager
password (encrypted alphanumeric string): self-descriptive; maximum 200 characters

Branch Manager

id (integer): uniquely identifies the branch manager of a business account
first_name (string): first name of the branch manager; maximum 50 characters
last_name (string): last name of the branch manager; maximum 50 characters
email (string): email address of the branch manager
password (encrypted alphanumeric string): self-descriptive; maximum 200 characters

Branches

id (integer): uniquely identifies a single branch of a business account
branch_name (string): a string that can describe a branch as per the user's wish; maximum 100 characters

Permissions

id (integer): uniquely identifies a permission given to a manager
permission_description (string): detailed description of the permission
permission_name (string): name of the permission

Products

id (integer): uniquely identifies each product
product_name (string): product name; maximum 100 characters
product_description (string): description of the product; maximum 1000 characters
product_price: (floating point): price of the product; 2 decimal places

Combos (Parent: Products)

id (integer): uniquely identifies each combo
products (array): list of all products in the combo
combo_price (float): can be derived from product prices in the array or included separately

Categories

category_id (integer): uniquely identifies each category
category_name (string): name of the category; maximum 100 characters

Branch Product Catalogs

branch_id (integer): identifies branch
products (array): array of products assigned to that branch

Sales Channels

id (integer): uniquely identifies a sales channel for a business
name (string): name of the sales channels; maximum 50 characters

Campaigns

<can be applied to one or multiple products of one branch or multiple branches, to one category or multiple categories of a branch, or to a sales channel>
id (integer): uniquely identifies a campaign

Discounts (Parent: Campaigns)

id (int): identifies a discount campaign
discount_name (string): name of the discount

WorkTime Sessions

id (datetime string): identifies worktime sessions uniquely
start_time (datetime string): start time of the worktime session
end_time (datetime string): end time of the worktime session
orders (array of orders): list of all the orders
branch_id: reference to the branch this WorkTime Session belongs to

Orders

id (integer): identifies an order uniquely
products_id (array): array of reference of products in the order
order_status : reference to the order status id
campaigns_id (array): campaigns if they are applicable for an order
sales_channel_id: reference to the sales channel of the order
worktime_session_id: reference to the WorkTime session an order belongs to

Order Status

id (int): identifies an order status
order_status: a string describing the status of an order (e.g. Initiated, Confirmed, Processing, Completed, Completed But Payment Pending)

5.2 Entity-Relationship Diagram

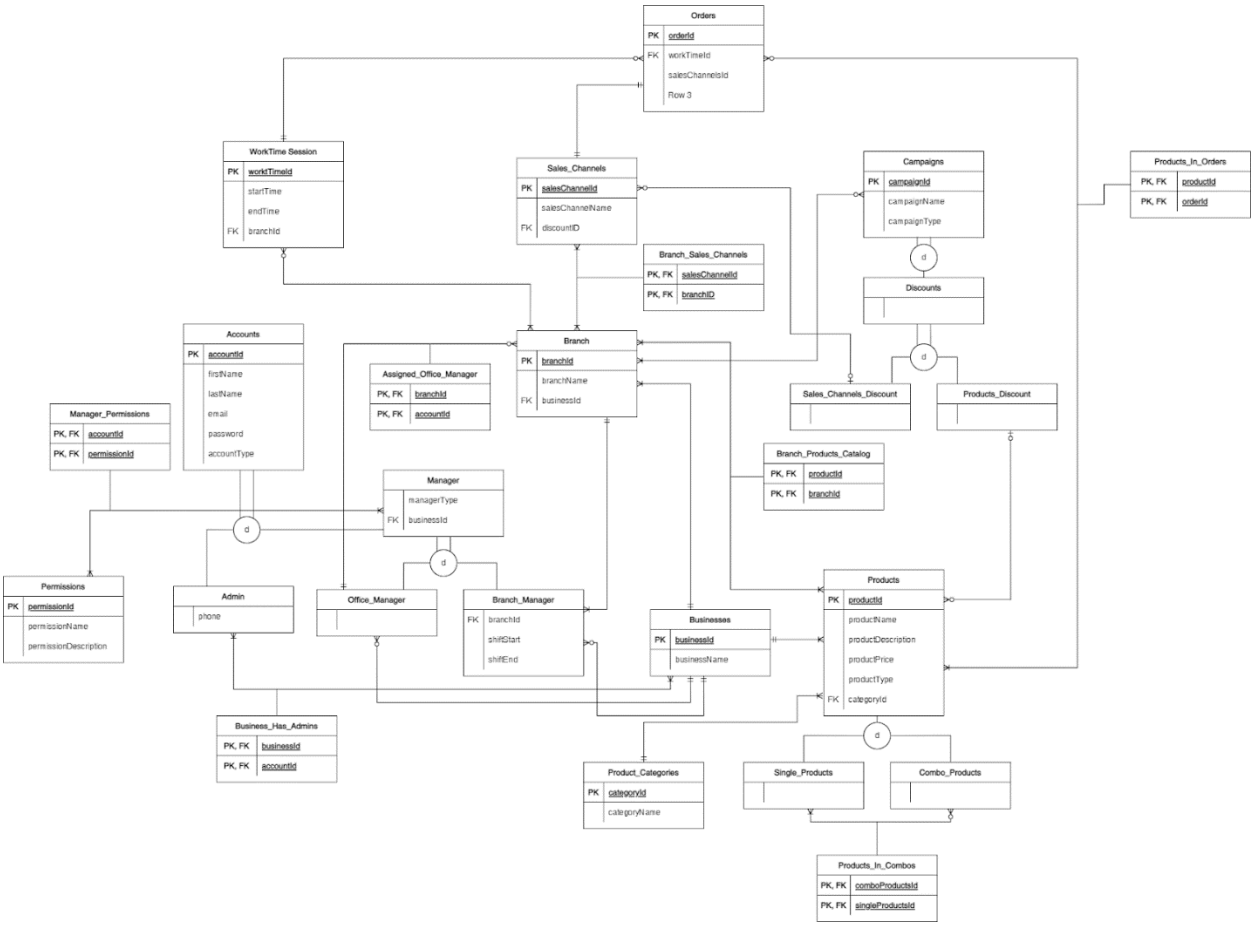

Figure 8 - ERD

5.3 System Analysis & Design

List of Features

Features for each business and its users

1. Open a business workplace account on our platform and allow other stakeholders to join the workplace
2. Edit, create, and delete branches of an organization
3. Add, remove, edit categories and items to product catalog and assign product catalog to one or multiple branches as branch catalog
4. Edit, create, and delete sales channels and assign sales channels to each branch
5. Edit, create, and delete campaigns (discounts and offers)
6. Take customer orders as they come and place orders from the point-of-sale

Features for software owners

1. Analyze data from users to improve user workflow
2. Analyze user sales, items, and volumes
3. Track accounts that are opened.
4. Track usage stats.
5. Setup pricing models, payment gateways, and track each business account information.

Feasibility Analysis

After conducting a feasibility analysis we have been able to show that this project is feasible for the organization. The initial version of the product can be feasible and provided as a SaaS product to a large number of SMEs, businesses and even large organizations at a low cost. We have conducted the following feasible analysis.

Economic Feasibility Analysis: The project is economically feasible as the tech stack used and the hardware cost to run the application is fairly low cost or made with free to use and open source technology. As the customer base of our organization increases, adjustments can be made to scale our product.

Technical Feasibility Analysis: The project also passes this feasibility test as the software will be made with open source technologies such as ReactJS, NodeJS, and MySQL. We are also using Git for version control. In addition, our users will only require a browser to operate our web application, there is no need for installation, setup, and updates and both PC and mobile users will be able to use our software.

Operational Feasibility Analysis: Our software also passes this feasibility analysis as our database will provide a secured and permanent place to record daily operational business data for our users' point-of-sale businesses. Each business and its corresponding employees will also see their own dashboard where they will be able to perform their business related tasks and processes which will help increase productivity for our users. Business admins will also be able to set which of their employees are able to do what in the software.

Problem Solution Analysis

During the development phase of the project, we faced various problems. We tried to solve them by using the following steps. They are:

- Identification: Identifying the problem as precisely as possible.
- Requirements that are affected: The project requirements that are affected by
- the problem.
- Solution: Finding the most accurate and precise solution.
- Implementation: Implementing the particular solution.
- Testing: Testing if the solution solved the problem. If not then the previous steps are followed again.

5.4 Software Development & Design

Use Case Diagrams

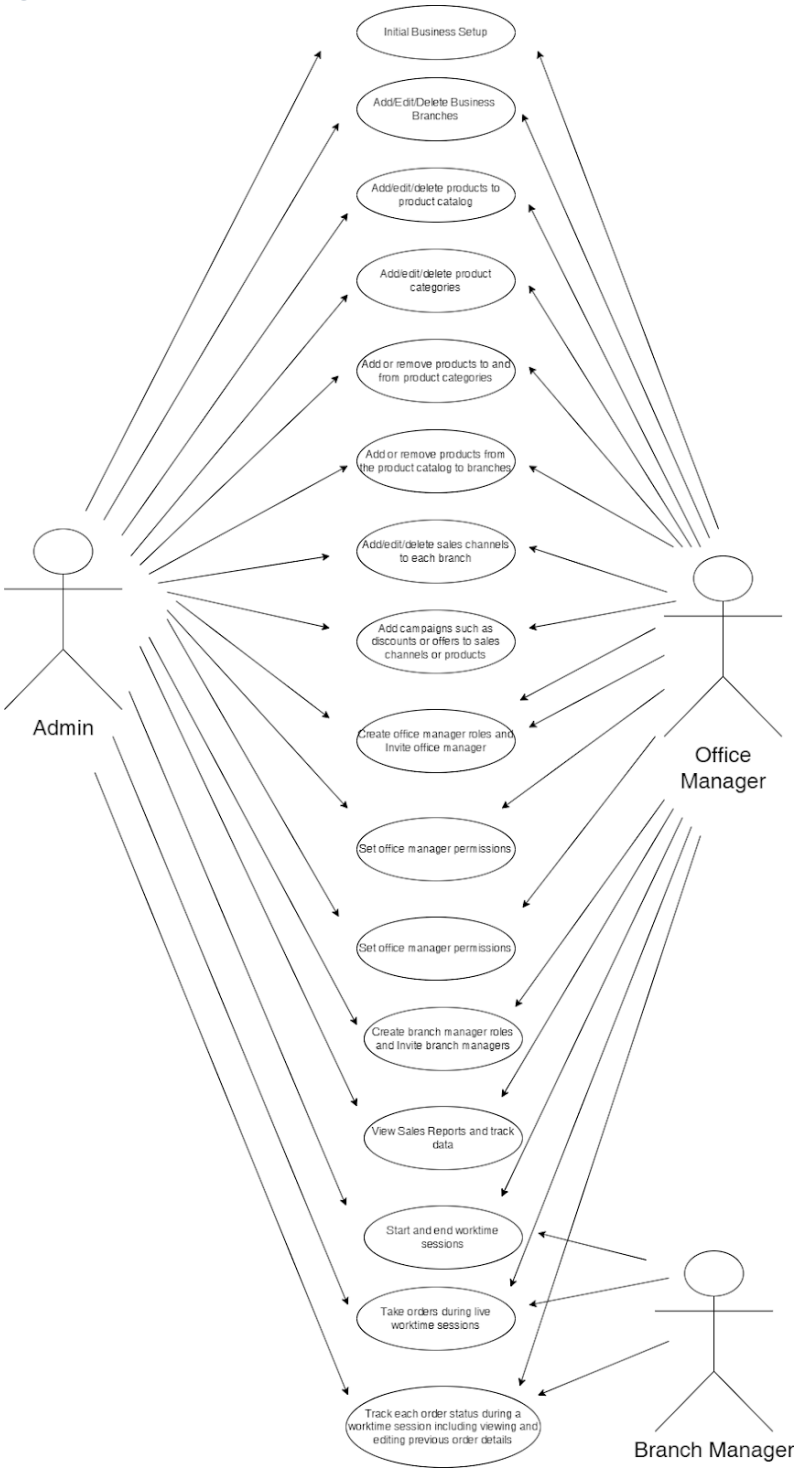

Figure 9 Use Case Diagram for all stakeholders

5.4 Non-functional Requirements

Non-functional requirements are the requirements that act as constraints and say how the system must conduct business processes. Our software will have the following non-functional requirements.

Security

- Each user that can log in must be the owner of a business or an employee of a business to log in to the dashboard. The business account must be opened beforehand.
- Users except the admin must be given the proper permissions by the admin of a business before they are able to carry out business processes through the dashboard or view sensitive data.
- Business information and data of one business must be kept secured and hidden away from other businesses as well as employees in the business who don't require to see it as per their role in the organization.
- The database must be able to backup data regularly, prevent loss of business data, and keep it safe from unwanted parties as such our web application must be secure to use and transfer data between business employees and to and from the client and server.

Performance

- The dashboard must load within 2 seconds.
- It should not take more than 3 seconds for any data that is entered through the dashboard/admin panel to be saved in the database and return acknowledgment or failure of data being saved.
- If there is a network error relevant information must be shown to the user.
- If there is a delay for one operation to be completed in the panel other operations must not stop.

Usability

- The web application must be usable on and backward compatible with Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera Browser.
- It must be usable through multiple-sized devices i.e computers, laptops, tablets, phones.
- It must be easy to use, navigate, and understand.

Maintainability & Manageability

This section also implicitly contains reliability and serviceability.

- The software must be easy to update, change, and modify.
- It must be customizable for different types of businesses.
- It must be able to recover the state of a user session in case of power failure, network failure or abrupt closure of the application.
- It must be able to recover the data of the business in the case of data loss or failure.

Availability

- The application must be available through any browser on any device at all times. It will be since it is a web application.

Scalability

- Application should be able to include more features without affecting other features.
- Application should be able to scale up as more business accounts are opened, as activity of an organization using the software increases, or if more employees/branches are added to an organization.

5.5 Product Features

Input

Input of POS Software	
Process	**Field Types**
Login	Email - string Password - string
Signup	Business name - string Email - string Password - string Retype password - string
Input Branches	Branch name - string
Input Branch Manager	Manager name - string Manager Email - string
Input Office Manager	Manager name - string Manager Email - string
Input Products	Product name - string Description - string Price - float
Input Categories	Category name - string
Input sales channels	Name - string
Input discounts	Discount name - string
Input orders	Products Sales Channels Discounts

Output

Output of POS Software	
Process	**Field Types**
Login	On success - redirect to business dashboard On failure - show notification and remain in login
Signup	On success - redirect to business dashboard On failure - show notification and remain in login

Input Branches	On success - add to database and show output On failure - show notification and don't add to database
Input Branch Manager	On success - add to database and show output, send email to manager On failure - show notification and don't add to database
Input Office Manager	On success - add to database and show output, send email to manager On failure - show notification and don't add to database
Input Products	On success - add to database and show output On failure - show notification and don't add to database
Input Categories	On success - add to database and show output under products On failure - show notification and don't add to database
Input sales channels	On success - add to database and show output in worktime session for each order On failure - show notification and don't add to database
Input discounts	On success - add to database and show output in worktime session for each order On failure - show notification and don't add to database
Input orders	On success - add to database and show output in worktime session for each order. Also print out order receipt. On failure - show notification and don't add to database

5.6 Architecture

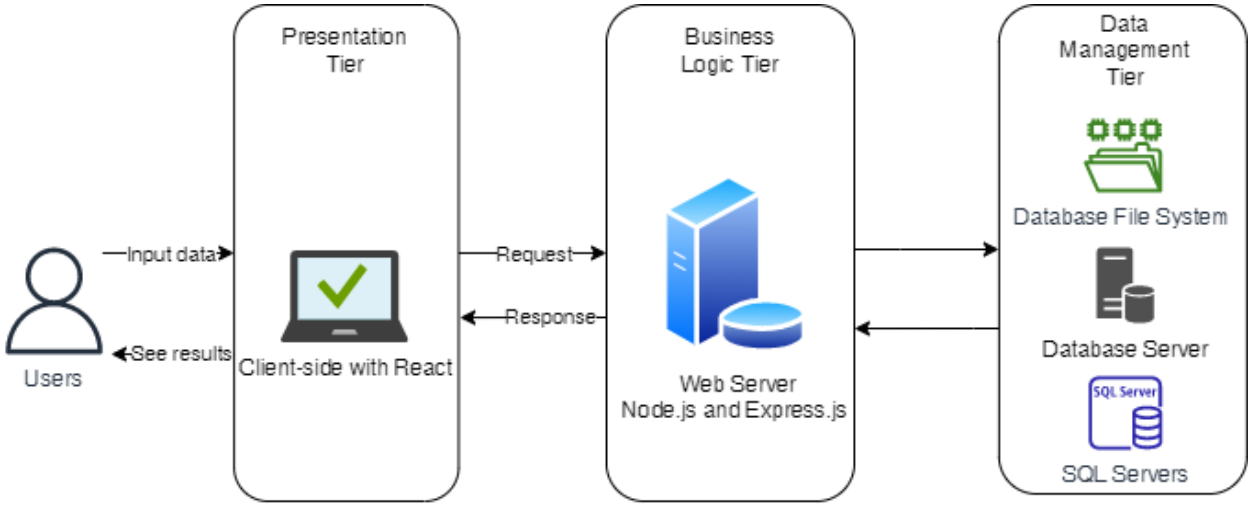

Figure 10 Architecture

Chapter 6

Results & Analysis

This part of the report shows the outcomes and the analysis of the software project throughout the time it was developed. This type of analysis is extremely important for all stakeholders involved as well as to gauge how much the software has fulfilled the requirements provided at the beginning of the project.

The analysis had started from the beginning of the project and was carried out throughout by the development team. Our strategy for the analysis was:

1. Did we obtain the specific goals of the project set out at the beginning of the project?
2. How will we determine that our specific goals have been accomplished?
3. Who will be the one to accomplish a specific goal?
4. When will each of the goals be accomplished?
5. How will the outcomes be applied/implemented?

We used the final software to answer all of our questions such as what the main goal of the software is, who the users are, the kind of data we wanted to store and process. The final software was then used to accomplish most of the requirements provided to us by the software owners.

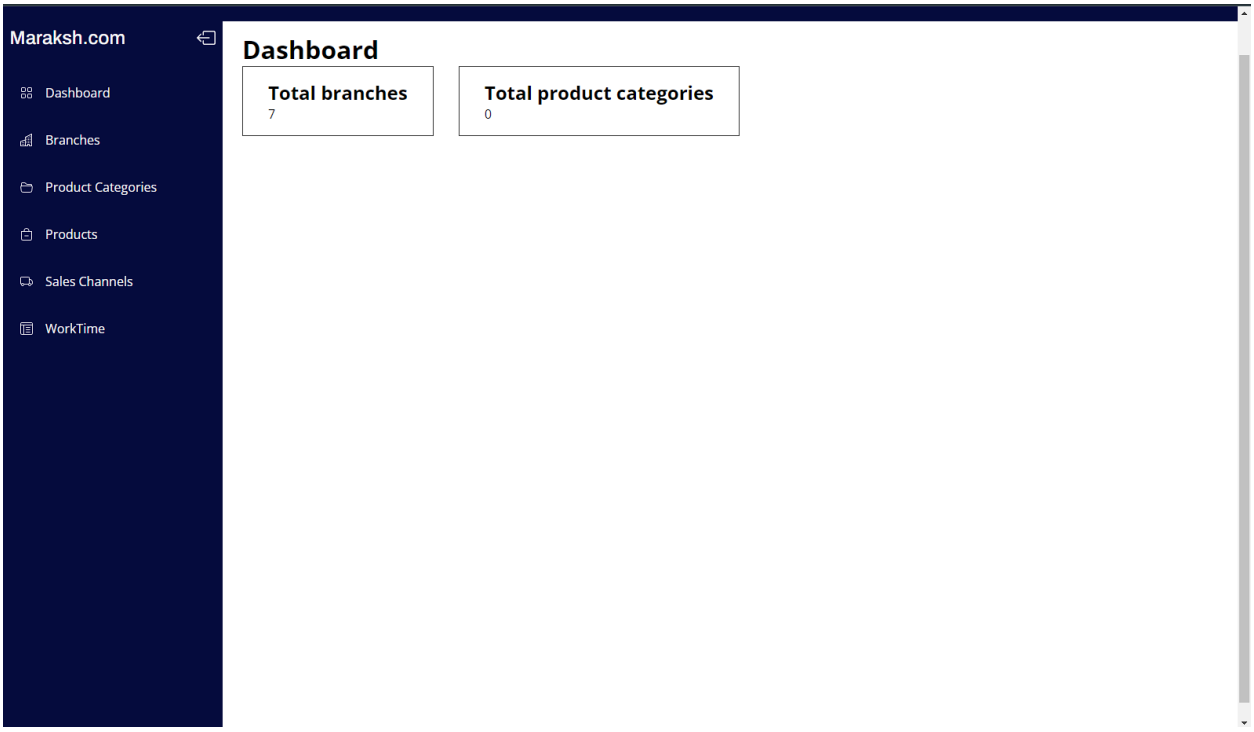

Figure 11 - Dashboard Home

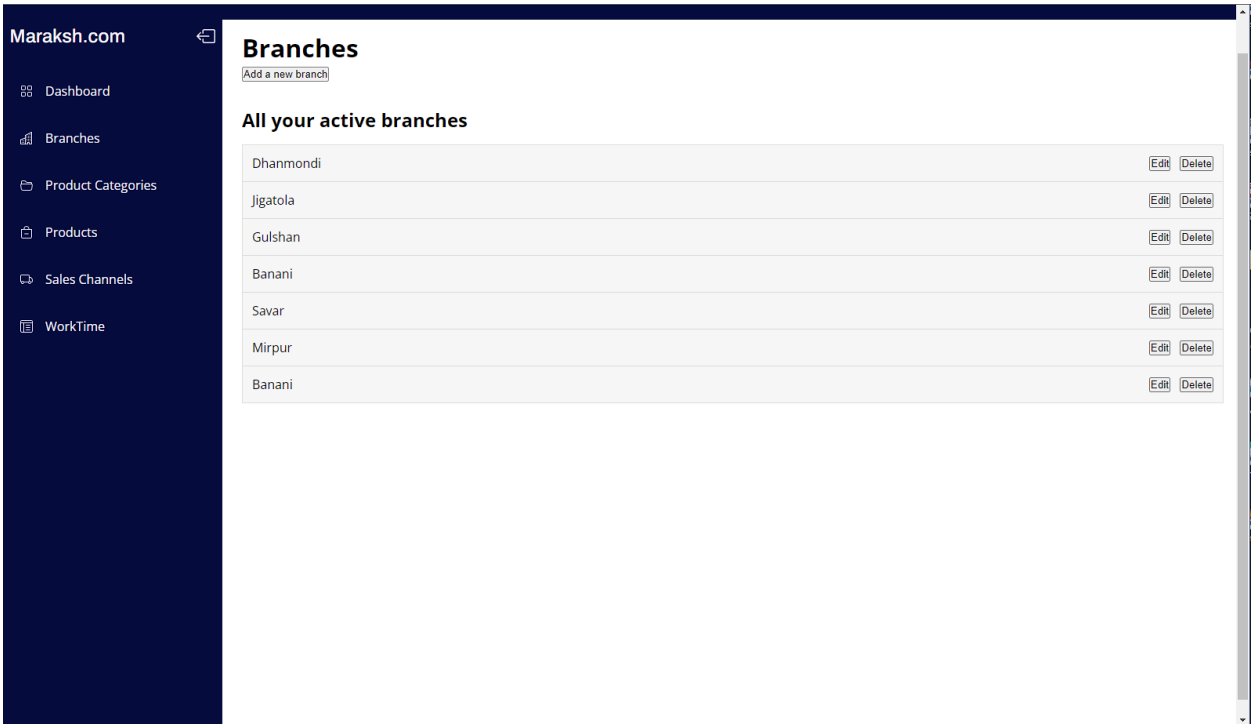

Figure 12 Branches

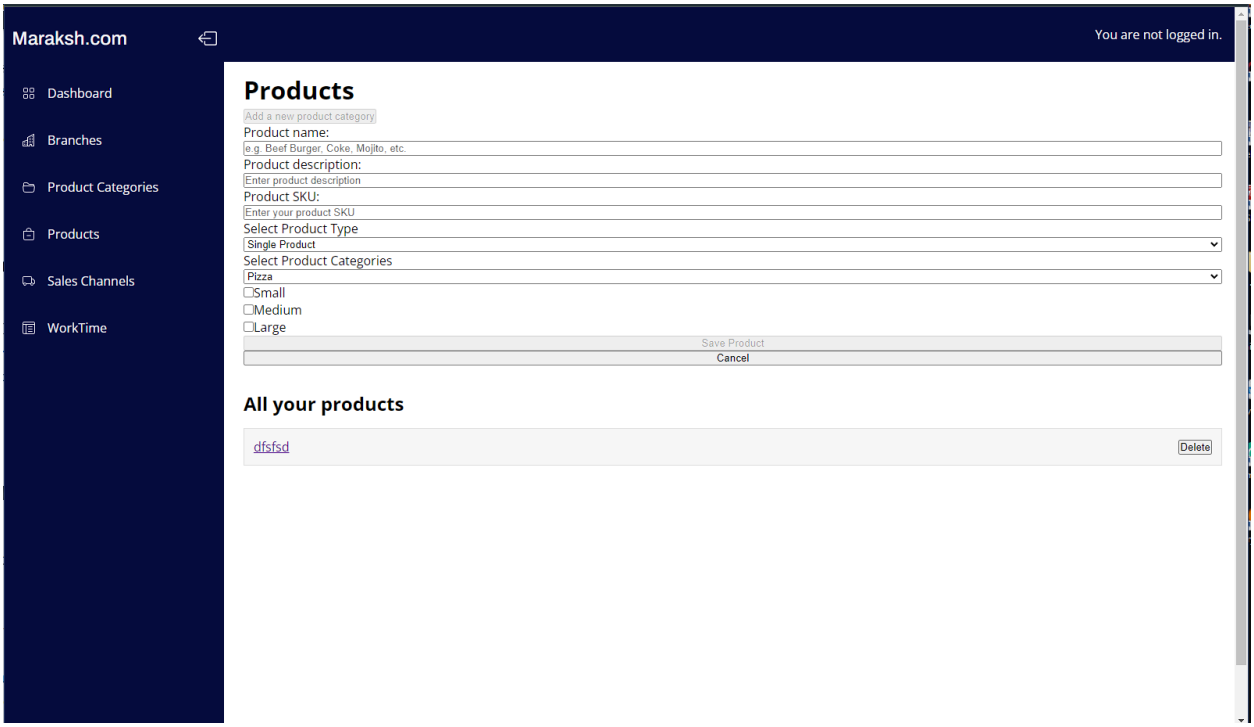

Figure 13 Products

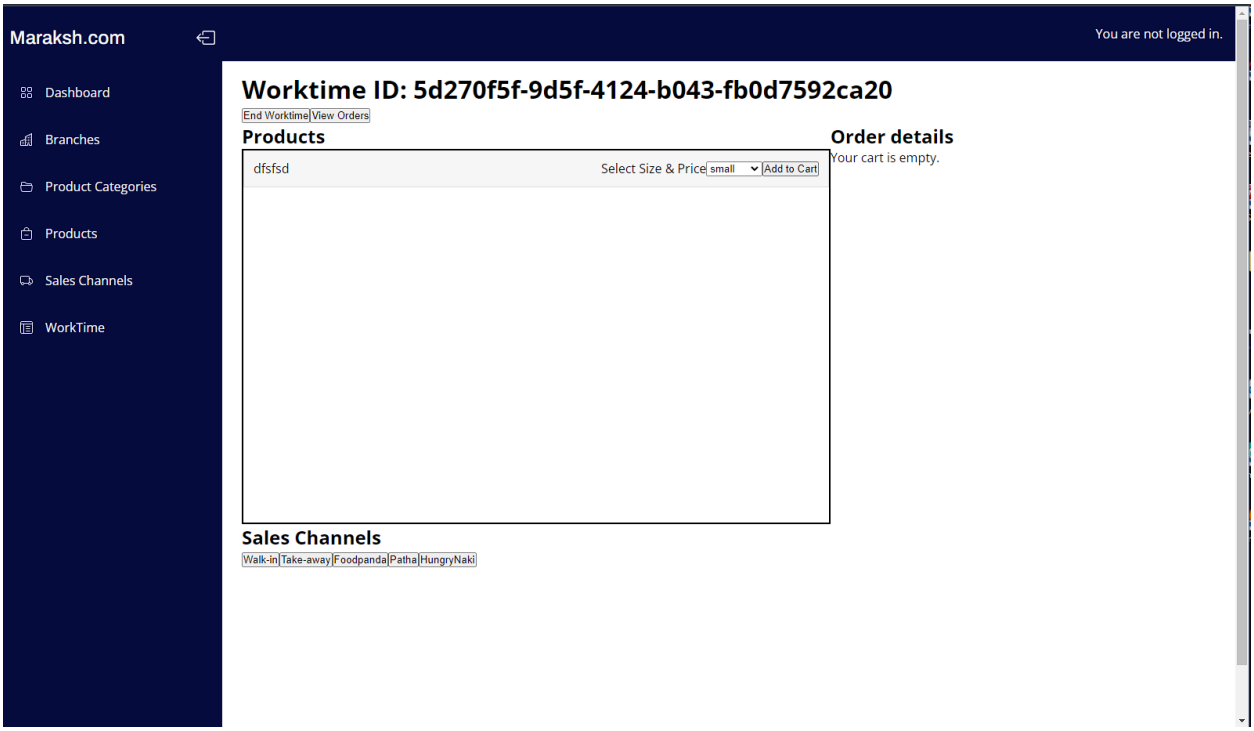

Figure 14 Cart

The developers went on to accomplish most of the requirements (both functional and non-functional) within the due time and are currently in the process of writing a user manual as well to

use the product. We also used the functional and non-functional requirements to determine how much of the project was complete.

We did the following to analyze the project:

1. Used various research papers to understand and compare our project.
2. Understand and verify the data that was being collected and stored/processed.
3. Evaluate the project against itself as time passes.
4. Compared the work done versus the planned schedule.

After that, we conducted each activity in our software to see if the desired outcome occurred or not. With that in mind the objectives or the requirements provided to us by the software owners were achieved.

We determined that there may be three reasons why a specific activity may or may not show the desired outcome.

1. The logic was right, but the execution was poor.
2. The logic was solid, and the activity was carried out perfectly, but the lacking was in the observation.
3. The reasoning was faulty.

After deliberate testing of each activity, all the activities gave perfect results and the outcomes hoped were also achieved.
This was communicated to my organization supervisor with details included as well as necessary reports. We also gathered more requirements for the next phase of the project.

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

Sustainability is the ability to maintain something at a certain level for the long run. For our project as well, sustainability of the software was thought of from the beginning of the project. As requirements change, more will be added, the current code will be refactored and improved, and the quality of the software will get better as well.

We built this software with ReactJS, NodeJS, ExpressJS, and MySQL. As JavaScript is one of the most popular languages out there and Express is one of the most popular frameworks for server-side programming, as well as ReactJS, is one of the most popular front-end libraries out there consistently evolving and getting better, we can rest assured that the project can be supported for the long term.

7.2 Social and Environmental Effects and Analysis

The social and environmental effects of the software are provided below.

Social Effects Analysis

The software will be provided to clients of various organizations and businesses who will sign up as monthly customers. It will be used to measure each business's various point-of-sale data including important sales-related data.

It will allow various employees in each organization to fulfill their specific roles through the software and gather, view, and process important business and sales data.

Environmental Effects Analysis

As our software is completely online so there is no environmental impact other than the electricity used to run the servers where the code and data will be stored. Other than that, the software will have specific user roles for each organization using it so there is no chance for data to be forged or stolen or get corrupted.

7.3 Addressing Ethics and Ethical Issues

Since our software will be mostly managed by the admin or manager of each organization there will not be an ethical issue here. While sensitive data that is not meant for anyone else in the organization will also not be shown to anyone other than the manager or the admin. However, the managers must be loyal and sincere about their work and not leak out sensitive financial data of each organization.

Chapter 6

Lessons Learned

6.1 Problems Faced During this Period

During the period within which we tried to develop this software we faced a multitude of problems.

1. Planning phase: During the planning of the project we had to figure out how exactly to implement the project and which methodologies we had to follow.
2. Designing the database and ERD: During the designing of the database and the ERD we had a lot of problems with which data we need to collect from users based on their requirements.
3. Understanding NodeJS, ExpressJS, and ReactJS: Even though most of our team was good JS programmers, for me it took some time to really understand how to build a server and front-end with JavaScript as well as connect it to our MySQL database as we had to decide between using ORM, a No-SQL database, as well as directly integrating Node with MySQL.

6.2 Solution to those Problems

My teammates were professionals and helped me along to learn the concepts and everything necessary to build the product. We also did a weekly meeting where we would discuss what is wrong and how to overcome the problems of the previous week in the following week. Online documentation of the technologies used also helped us a lot. Also since we followed the agile methodology, after each sprint we were able to figure out whether we need to change something about the database design, which eventually helped us build a good ERD.

Chapter 7

Future Work & Conclusion

7.1 Future Work

Our plans for the future are as follows:

1. Include a proper password protected system that distinguishes each user and their roles in the organization in the software.
2. Include a better user interface with better user experience.
3. Integrate the software with a printer so each order can be printed out.
4. Make the project available in production mode.
5. Include an inventory management system.
6. Find out if there are any errors in the software that we haven't noticed and solve them.
7. Consistently provide support and improve the existing features in the system.

7.2 Conclusion

This internship opportunity was my first professional experience with trained experts and gave me the chance to understand how software is built in the real world. I was able to implement the things I have learned so far through my university courses so far and gave me the practical experience I needed. Now, my own contributions will be used by real customers in the world.
Even though, due to time constraints the product I developed is not perfect, I faced many challenges and eventually overcame them which gave me a great learning opportunity that I so much needed. I learned JavaScript, ReactJS, Node.js and Express.js and much more as well such as leadership, management, team building and social and professional skills as well. I am very grateful for this opportunity to have this experience from my internship.

Bibliography

[1] A I Sourav, A. E. (2020). Feasibility Study of Web-Based POS System Implementation. *IOPScience*, 9.

An Undergraduate Internship/Project on yourTopic

By

Tashfeen Mustafa Choudhury

Student ID: **1821069**

Summer, 2022

Consent Form

The student modified the internship final report as per the recommendations made by his/her academic supervisor and/or panel members during final viva, and the department can use this version for archiving.

(Signature of the Supervisor)

Dr. Sadiaque A. Sazid

Name of the Supervisor

Department of Computer Science & Engineering

Independent University, Bangladesh