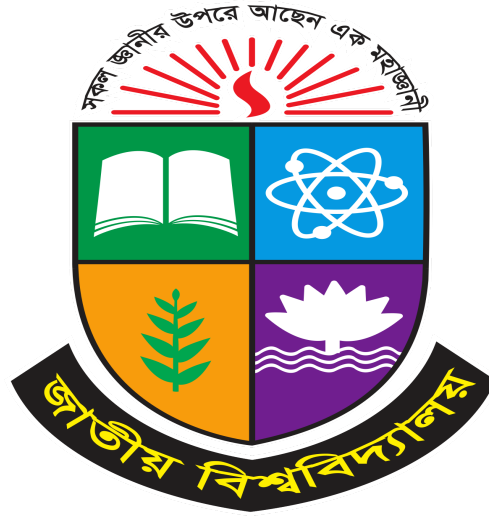


Business ERP for Entrepreneur



A Project presented to the National University in partial fulfillment of the requirement for the degree of Bachelor of Science (Hon's) in Computer Science & Engineering

Submitted By

Sk Md Hasib Unnoor

Registration No: 17502004982

Session: 2017-2018

Supervised By

Md. Imran Hossain

Head of CSE Program

Daffodil Institute of IT (DIIT)



Department of Computer Science & Engineering

Daffodil Institute of IT, Dhaka

Under National University, Bangladesh

August 29, 2023

APPROVAL

The Project “**Business ERP for Entrepreneur**” is submitted to the Department of Computer Science & Engineering, Daffodil Institute of IT (DIIT), under National University of Bangladesh in partial fulfillment of the requirements for the degree of Bachelor of Science (Hon’s) in Computer Science and Engineering and approved as to its style and content.

Md. Imran Hossain
Head and Supervisor
Department of CSE, DIIT

DECLARATION

We affirm that the project work titled “**Business ERP for Entrepreneur**” is being submitted in partial fulfillment for the degree of B.Sc. (Hon’s) in Computer Science & Engineering is the original work carried out by me. It has not formed the part of any other project work submitted for any degree or diploma, either in this or any other University.

Submitted By

Sk Md Hasib Unnoor

Registration no: 17502004982

Session: 2017-2018

ACKNOWLEDGEMENTS

Despite our efforts, the success of this project depends largely on the encouragement and guidance of our mentors. We would like to take this opportunity to express our gratitude to the people who are playing a vital role in the successful completion of this project.

Our sincere thanks to **Prof. Dr. Mohammed Shakhawat Hossain, Principal, DIIT** who has allowed us to work on this project and showed encouragement.

Our cordial thanks to our Project Supervisor **Md. Imran Hossain, B.Sc, M.Sc., Lecturer, Department of Computer Science & Engineering, DIIT** for his valuable guidance and support to meet the successful completion of our project.

We extend our sincere thanks to our family & classmates for their constant support throughout this project.

Finally, we would be grateful to **National University, Bangladesh** and coordinators of the **Bachelor of Science in Computer Science and Engineering degree program** for giving us this opportunity to apply the knowledge that we have gained throughout the study of the degree program.

ABSTRACT

This project is aimed at developing a web based platform application named “Business ERP for Entrepreneur” for entrepreneurs or new organization. In the modern business landscape, entrepreneurs and new organizations face unique challenges as they strive to establish and grow their ventures. Managing various business functions efficiently, coordinating resources, and maintaining accurate records are essential for sustained success. To address these needs, we present "ERP," a specialized Enterprise Resource Planning (ERP) application tailored to the requirements of entrepreneurs and new organizations. ERP is designed to streamline and integrate critical business processes, providing a comprehensive suite of tools to manage finances, operations, human resources, customer relations, and more. This ERP application aims to empower entrepreneurs by offering them a centralized platform that enhances decision-making, optimizes resource allocation, and facilitates growth.

TABLE OF CONTENT

Approval	i
Declaration.....	ii
Acknowledgement.....	iii
Abstract.....	iv
CHAPTER 1: INTRODUCTION	1-8
1.1 Introduction.....	2
1.2 Background of Study.....	2
1.3 Objectives	3
1.3.1 Primary Objectives.....	3
1.3.2 Secondary Objectives.....	3
1.4 Limitations of Existing System.....	4
1.5 Proposed system.....	4
1.6 Methodology.....	5
1.6.1 Data source.....	5
1.7 Process model.....	5
1.7.1 Why Incremental Process Model.....	6
1.8 Feasibility Study.....	6
1.8.1 Economic Feasibility.....	7
1.8.2 Technical Feasibility.....	7
1.8.3 Operational Feasibility.....	7
1.8.4 Scheduled Feasibility.....	7
1.9 Conclusion.....	8
CHAPTER 2: REQUIREMENT ENGINEERING	9-12
2.1 Introduction to Requirement Engineering.....	10
2.2 Requirement Analysis.....	10
2.2.1 Functional Requirements.....	10
2.2.2 Non-Functional Requirements.....	11
2.3 Other Requirements.....	11
2.3.1 Hardware Requirement.....	11

2.3.2 Software requirements.....	11
2.4 Benefits of the system.....	12
2.5 Conclusion.....	12

CHAPTER 3: ANALYSIS & DESIGN 13-25

3.1 Introduction.....	14
3.2 Use Case Diagram.....	14
3.2.1 General Use Case Diagram	14
3.2.2 Purchasing info & Status use case diagram.....	15
3.2.3 Product info & Status use case diagram.....	15
3.2.4 Sales info use case diagram	16
3.3 Activity Diagram.....	16
3.3.1 Activity Diagram (Admin).....	19
3.4 Entity Relationship Diagram (ERD).....	19
3.5 Process Flow Diagram.....	21
3.6 Data Flow Diagram (DFD).....	22
3.6.1 DFD Level 0.....	23
3.6.2 DFD Level 1.....	23
3.7 Sequence Diagram.....	24
3.8 Conclusion.....	25

CHAPTER 4: FUTURE ENHANCEMENT AND 26-27

4.1 Introduction.....	27
4.2 Limitations of Proposed System.....	27
4.3 Future Enhancement.....	27

CHAPTER 5: IMPLEMENTATION 28-37

5.1 login.....	28
5.2 Dashboard	30
5.3 Product Details.....	31
5.4 Supplier Details.....	32
5.5 Customer Details.....	32
5.6 Product Sale.....	33

5.7 Account and Transaction Details.....	34
5.8 Bill preparation and Invoice receipt.....	35
5.9 Human resources (HR)	36
5.10 Setup Menu.....	37

CHAPTER 6: CONCLUSION 38-41

6.1 Conclusion.....	39
References.....	41
Appendix.....	42

LIST OF FIGURES

Figure 1.7: Incremental process model.....	6
Figure 3.2.1: General Use Case Diagram.....	14
Figure 3.2.2: Purchasing info & Status use case diagram.....	15
Figure 3.2.3: Product info & Status use case diagram.....	15
Figure 3.2.4: Sales info use case diagram.....	16
Figure 3.3.1: Activity Diagram (Admin).....	19
Figure 3.4: ER Diagram	21
Figure 3.5: Process Flow Diagram.....	22
Figure 3.6.1: DFD Level 0.....	23
Figure 3.6.2: DFD Level 1.....	23
Figure 3.7: Sequence Diagram.....	24
Figure 5.1: Login Interface.....	29
Figure 5.2: Dashboard Interface.....	30
Figure 5.3(a): Product Details.....	31
Figure 5.3(b): Product List.....	31
Figure 5.4: Supplier Details.....	32
Figure 5.5: Customer Details.....	32
Figure 5.6: Product sale.....	33
Figure 5.7(a): Balance Sheet.....	34
Figure 5.7(b): Revenue Sheet.....	34
Figure 5.8: Invoice receipt.....	35
Figure 5.9(a): HR Staff Details.....	36

Figure 5.9(b): HR Role Details.....	37
Figure 5.10: Setup Menu.....	37



Chapter 1: Introduction

1.1 Introduction

The project “Business ERP for Entrepreneur” is a complete web based application designed on JavaScript technology using Visual Studio Software. The main aim of the project is to develop Enterprise resource planning (ERP) software. In the dynamic landscape of modern business, entrepreneurs and new organizations face unique challenges as they strive to establish themselves and grow. To navigate these challenges effectively, a powerful and tailored business Enterprise Resource Planning (ERP) application can prove to be an invaluable asset. This ERP application is specifically designed to cater to the needs and aspirations of entrepreneurs and new businesses, providing them with a comprehensive and integrated suite of tools to streamline operations, enhance decision-making, and drive sustainable growth.

1.2 Background of Study

The rapid growth of entrepreneurship and new organizations has highlighted the need for efficient and comprehensive business management solutions. Entrepreneurs and emerging businesses often face numerous challenges in managing their operations, resources, and finances. In this context, the development of a specialized Enterprise Resource Planning (ERP) application tailored to the unique requirements of these entities becomes crucial. This study aims to explore the background and rationale behind the creation of a business ERP application specifically designed to address the needs of entrepreneurs and new organizations.

The application's design and features will be informed by the distinct characteristics of startups and small businesses, such as limited resources, dynamic operational needs, and a focus on scalability. By examining the challenges faced by these entities and the limitations of existing ERP solutions, this study will pave the way for the development of a cutting-edge business management tool that empowers entrepreneurs to streamline their processes, optimize resource allocation, and make informed decisions. The outcome of this research will contribute to enhancing the operational efficiency and overall success of entrepreneurs and new organizations in an increasingly competitive business landscape.

1.3 Objectives

An ERP (Enterprise Resource Planning) application focused on entrepreneurs or new organizations would need to address the specific needs and challenges that these types of businesses face.

1.3.1 Primary Objective

The essential destinations of the venture are referenced beneath:

- To satisfy the prerequisite for accomplishing a Bachelor's qualification in Computer Science and Engineering.
- To know the essentials of the JavaScript Technology and Sublime Text /Visual Studio with Express Js Framework.

1.3.2 Secondary Objective

- To maintain an organization's transaction in a computerized system.
- To give an organization's owner a better facility.
- To analyze several parts of business.
- To inform the owner about the storage of production.
- To manage sales, profits and losses.
- To give subtleties data about the stock parity.
- To make the stock sensible and improve the utilization of stock in the association.
- To manage role and permission

1.4 Limitations of Existing System

As we know manual systems are quite tedious, time consuming and less efficient and accurate in comparison to the computerized system.^[7]

So following are some disadvantage of the old system:

- Time consuming.
- Less accurate
- Less efficient
- Lot of paperwork
- Slow data processing
- Not user friendly environment
- Difficult to keep old records

1.5 Proposed System

This system is to provide a user efficient working environment and more output can be generated through this. This system provides a user-friendly interface resulting in knowing each and every usability features of the system.

This system helps in tracking records so that past records can be verified through and one can make decisions based on the past records. This system completes the work in very less time consumption and high level of efficiency.

This system is developed in such a way that even a naïve user can also operate the system easily. The calculations are made very quickly and the records are directly saved into databases and the databases can be maintained for a longer period of time. Each record can be retrieved and can be verified for future transactions.

Also the system provides a high level of security for data lacking as only admin people can access the database no changes can be made in it until it verifies the user login id and password.

We also have an operator through which the operator can take orders but can't make changes in the database. Limited access is available to the operator.

1.6 Methodology

The development process on this project will complete the following structure described later on Software Analysis & Design.

This study on this project is tentative in nature. It aims to develop any type of Enterprise resource planning. The variables identified to manipulate through a handy inspection and from primary and secondary data.

1.6.1 Data Sources

For this project in data collection phase we collected two types of data i.e.

Primary Data: Primary data are generated by consulting with the faculty. Faculty practical experience, observation, and face-to-face interview with Faculty helped us to generate the primary data.

Secondary Data: Primary data are generated by consulting with the faculty. Faculty practical experience, observation, and face-to-face interview with Faculty helped us to generate the primary data.

Secondary data is generated by studying different articles, newspapers, research papers and of course information collected via the Internet. Data, facts and statistics collected from different websites and sources made us understand the project better.

1.7 Process model

In the incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle amulet. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.^[2]

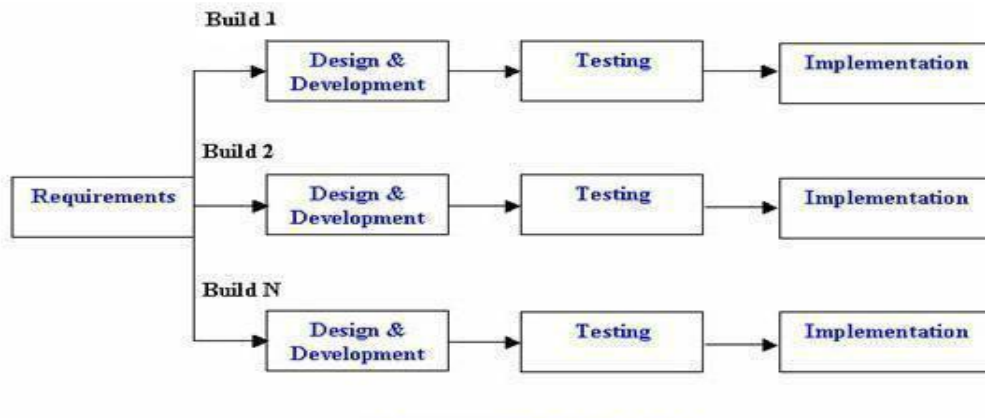


Figure 1.7: Incremental process model

1.7.1 Why Incremental Process Model

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customers can respond to each build.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during its iteration.

1.8 Feasibility Study

This software has been tested for various feasibility criteria from various points of views.

1.8.1 Economic Feasibility

The system is estimated to be economically affordable. The system is a medium scale web application and has an affordable price. The benefits include increased efficiency, effectiveness, and better performance. Comparing the cost and benefits the system is found to be economically feasible.

1.8.2 Technical Feasibility

Development of the system requires tools like:

- Visual Studio 2022
- pgAdmin 4
- Postman
- Node js Runtime

1.8.3 Operational Feasibility:

Programs that reduce costs without reducing the quality of a product are an example of operational feasibility. Studies are performed to make sure that programs can be initiated in the current production facility, without the need for more equipment or staff. If there is a need for more room, machinery or personnel, the system needs to make an improvement in the way the product is perceived by consumers. This allows the added expense to manufacture the product because of the anticipated increase in sales and revenue.

1.8.4 Schedule Feasibility

The organized schedule for the development of the system is presented in the schedule sub-section. The reasonable timeline reveals that the system development can be finished in the desired time frame.

1.9 Conclusion:

Above discussion shows that inventory management is essentially a variety of techniques, tools and technologies that a business uses to manage and control their inventory. The way that it's utilized and implemented ranges from simple right through to complex. It depends on the needs and scope of the business and the capabilities and functionality of the management software used. For all inventory-centric businesses, inventory management software is an essential and valuable tool. It is what controls the flow of stock in and out, it maintains the correct inventory levels for all items and stock, it allows access to sales data and analytics, and is how businesses set individual safety-stock requirements.

Chapter 2: Requirement Engineering

2.1 Introduction to Requirement Engineering

Requirement engineering encompasses the tasks that lead to an understanding of what the business impact of the software will be, what the user wants, and how end-users will interact with the software. Requirements engineering is defined in terms of its major activities.

- Understanding problems
- Solution determination
- Specification of a solution that is testable, understandable, maintainable and that satisfies project quality guidelines

Designing and building an elegant computer program that solves the wrong problem serves no one's needs. That's why it is important to understand what the customer wants before we begin to design and build a computer-based system.

2.2 Requirement Analysis

Requirement analysis provides the software designer with a representation of information, function and behavior that can be translated to data, architectural, interface and component-level designs.

2.2.1 Functional Requirements

Functional requirements of this project are given below:

- System shall have a form to accept the product details
- System shall display transaction details
- System shall provide a search facility on customer name, Order placed, date of order, date of order dispatch, date of transaction, transaction amount etc.
- System should inform the pending order and make changes if the order is dispatched
- System should report any out of range values on numeric field
- System should report on any Invalid dates

2.2.2 Non-functional Requirements

- All user manual should be provided in necessary format
- Application should support 5 simultaneous user
- Transaction should be completed within 0.5 sec

2.3 Other Requirements

2.3.1 Hardware Requirements:

The hardware listed by no means a minimum requirement to run the system, but rather a base limit for running the system smoothly and comfortably. This is also considering the potential amount of traffic that may go through the server.

- Intel Dual-core 2.0 GHz processor
- Minimum 2 GB of RAM
- 80 GB of disk space
- Monitor
- & other basic computer components

2.3.2 Software Requirements:

- Web Server
 - Node JS
- Database Engine
 - PostgresSql
- Database Tools
 - pgAdmin 4
- Text Editor
 - Visual Studio Code

- Development tools
 - Visual studio (IDE)
 - Java Script (Programming language)
 - Express (Backend Java Script framework)
 - React Js (Frontend Java Script framework)
- Documentation
 - Microsoft Word 2019

2.4 Benefits of the System

- Administrators will be able to maintain this system more accurately.
- All the information will be stored on to the database with its formatted screens and built in databases.
- All the information can be carried out more easily or quickly than any other manual process.
- User can easily take all information any time when he needs that stored by himself previously.
- Users can manage the dashboard easily.
- The security of this software is high. Without login details no one can enter the dashboard area

2.5 Conclusion:

In conclusion, system requirements are one of the foundations to building a successful project. System requirements are a broad and also narrow detailed statement that the customer makes in order to achieve their requirements. The customer should have a list of exact requirements they specify.

Chapter 3: Analysis & Design

3.1 Introduction:

In system analysis a study of the system as detailed as possible will occur with the help of some diagrams i.e. Use Case Diagram, Workflow Diagram, Data flow Diagram and Entity Relational Diagram etc.

3.2 Use Case Diagram

A use case diagram is the simplest representation of a user's interaction with the system that shows the relationship between the user and the different Use Cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. ^[4]

3.2.1. General Use Case Diagram:

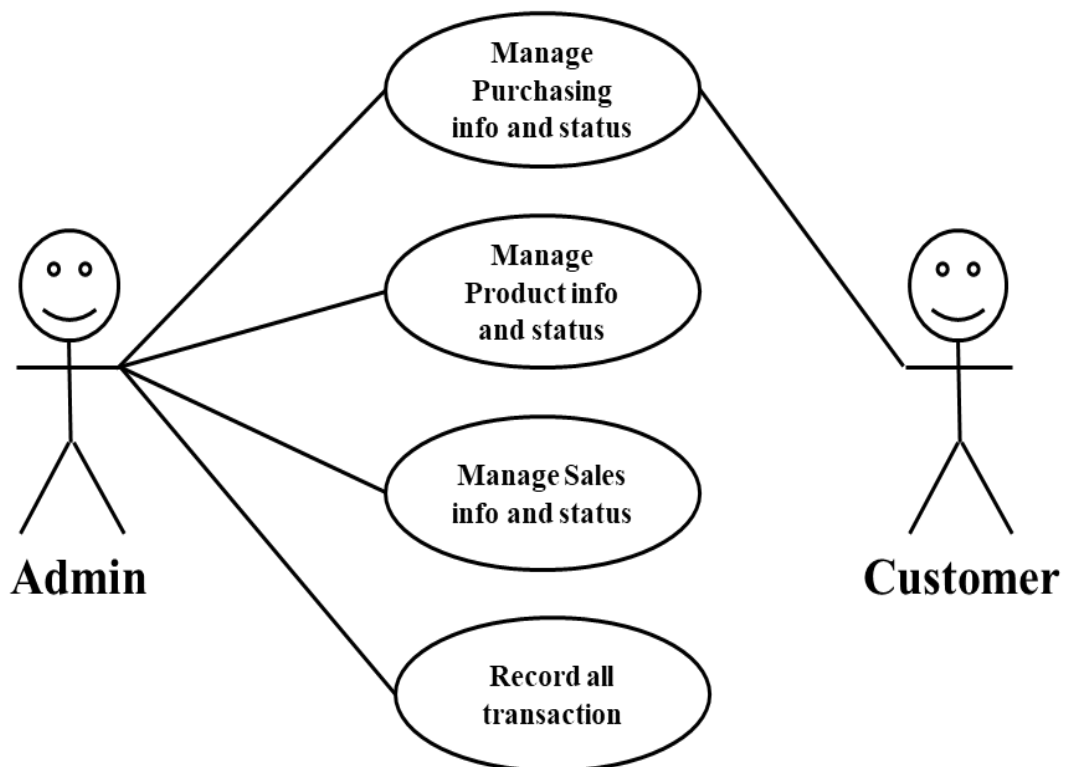


Figure 3.2.1: General Use Case Diagram

3.2.2. Purchasing info & status use case diagram:

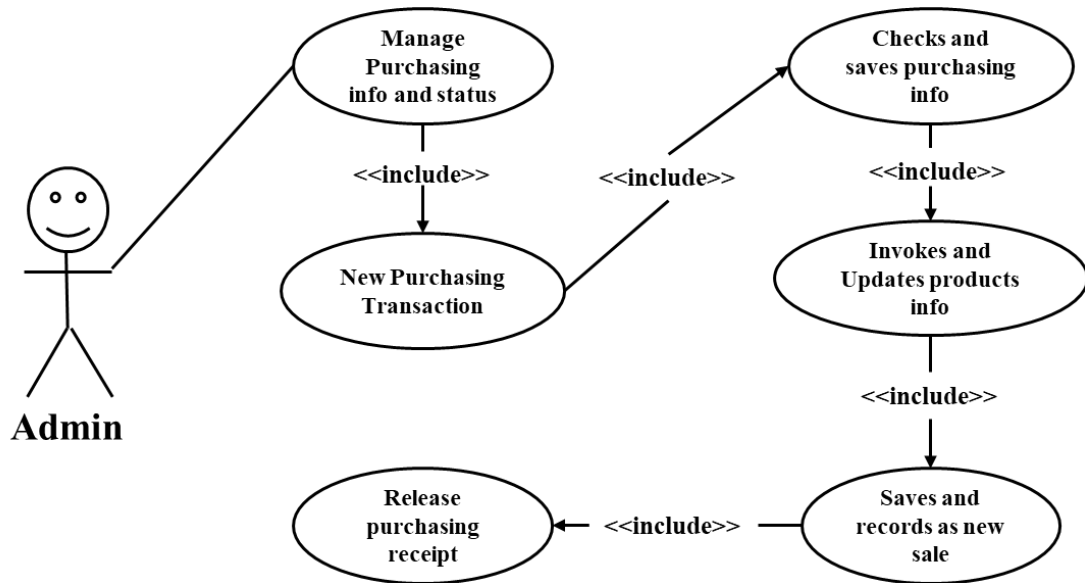


Figure 3.2.2: Purchasing info & status use case diagram.

3.2.3. Product info & status use case diagram:

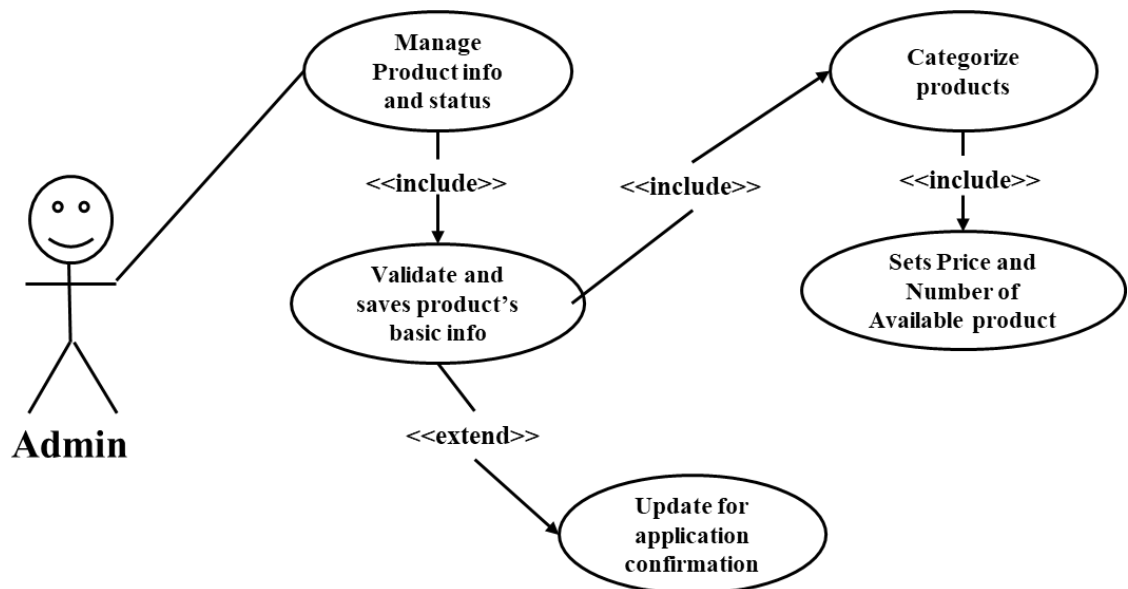


Figure 3.2.3: Manage Product Info and Status

3.2.4. Sales info use case diagram:

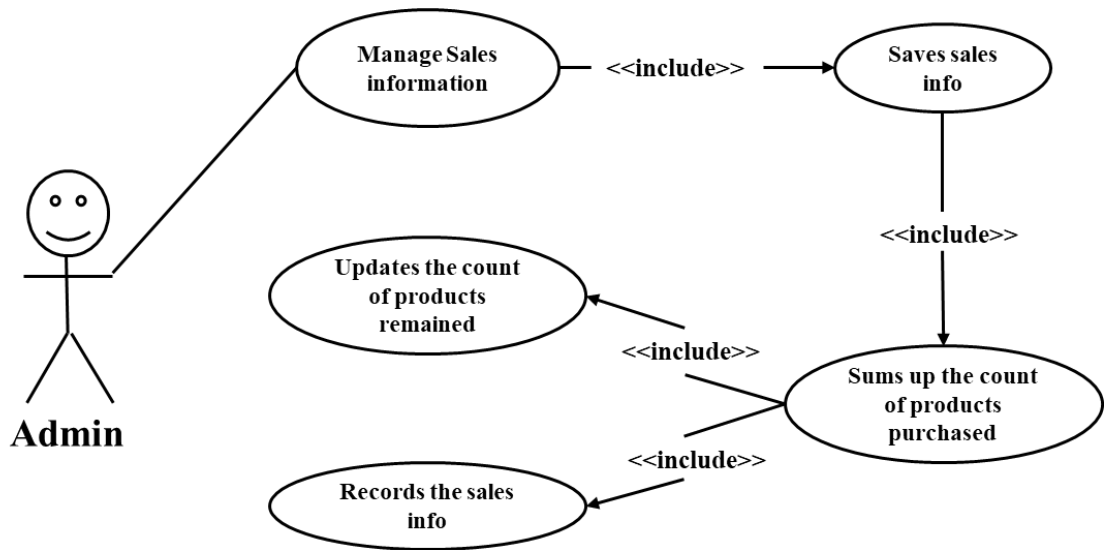


Figure 3.2.4: Manage Sales Information

3.3 Activity Diagram

Activity diagrams, which are related to program flow plans (flowcharts), are used to illustrate activities. In the external view, I use activity diagrams for the description of those business processes that describe the functionality of the business system. Contrary to use case diagrams, in activity diagrams it is obvious whether actors can perform business use cases together or independently from one another.

Activity

An activity diagram illustrates one individual activity. In my context, an activity represents a business process. Fundamental elements of the activity are actions and control elements (decision, division, merge, initiation, end, etc.).

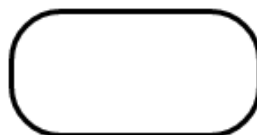


Figure 3.3(a): Activity

Elements are connected by so-called "activity edges" and form the "control flow", which can also be casually called 'flow'.

Action

An action is an individual step within an activity, for example, a calculation step that is not deconstructed any further. That does not necessarily mean that the action cannot be subdivided in the real world, but in this diagram will not be refined any further.



Figure 3.3(b): Action

The action can possess input and output information. The output of one action can be the input of a subsequent action within an activity. Specific actions are calling other actions, receiving an event, and sending signals.

Edge (Control Flow)

Edges, represented by arrows, connect the individual components of activity diagrams and illustrate the control flow of the activity:



Figure 3.3(c): Control flow

Within the control flow an incoming arrow starts a single step of an activity; after the step is completed the flow continues along the outgoing arrow. A name can be attached to an edge (close to the arrow).

Initial Node

The initial node is the starting point of an activity. An activity can have more than one initial node; in this case several flows start at the beginning of an activity:



Figure 3.3(d): Initial Node

It is also possible that an activity has no initial node, but is initiated by an event (action: accepting an event).

Activity Final Node

The activity final node indicates that an activity is completed. An activity diagram can have more than one exit in the form of activity final nodes:



Figure 3.3(e): Final Node

3.3.1 Activity Diagram (Admin)

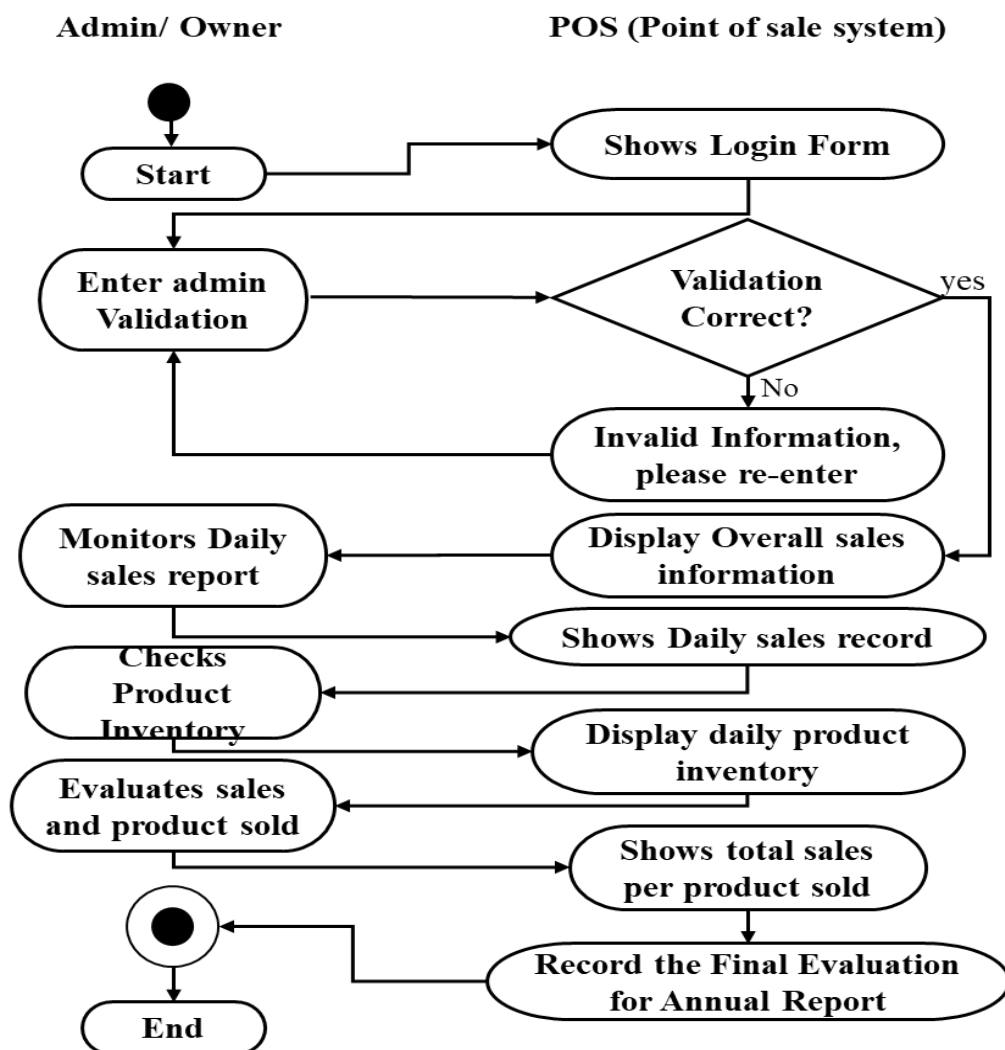


Figure 3.3.1: Activity Diagram (Admin)

3.4 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram (ERD) enables a software engineer to specify the data objects that are input and output from a system, the attributes that define the properties of these objects and their relationship.^[6] It provides an excellent graphical representation of the data structures and relationship. They provide a clear view of the logical structure of data within the boundary of interest and allow the engineer to model the data without considering the physical form. Some of the basic terms used in ERD described below:

Entity: An entity is an object with physical existence or may be an object with conceptual existence. For example, a car, a student, an employee, an applicant.



Figure 3.4(a): Entity

An entity represented by a rectangle.

Relationship: A relationship is a logical linkage between two or more entities which describes how the entities are associated with each other. A relationship described by a diamond.

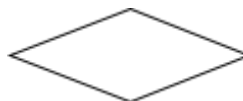


Figure 3.4(b): Relationship

Attribute: Attribute is a piece of information that describes a particular entity.

Figure: Attribute

Primary Key: A primary key is an attribute or collection of attributes that allow us to identify an entity uniquely.



Figure 3.4(c): Primary Key

Foreign key: A foreign key is an attribute of a relation, which refers to an existing attribute of another relationship.

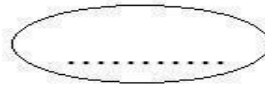


Figure 3.4(d): Foreign Key

Relationship Cardinality

Relationship cardinality refers to the number of entity instances involved in the relationship. The cardinality ratios are:

1:1 (One to One) 1: n (One to Many) n: n (Many to Many)

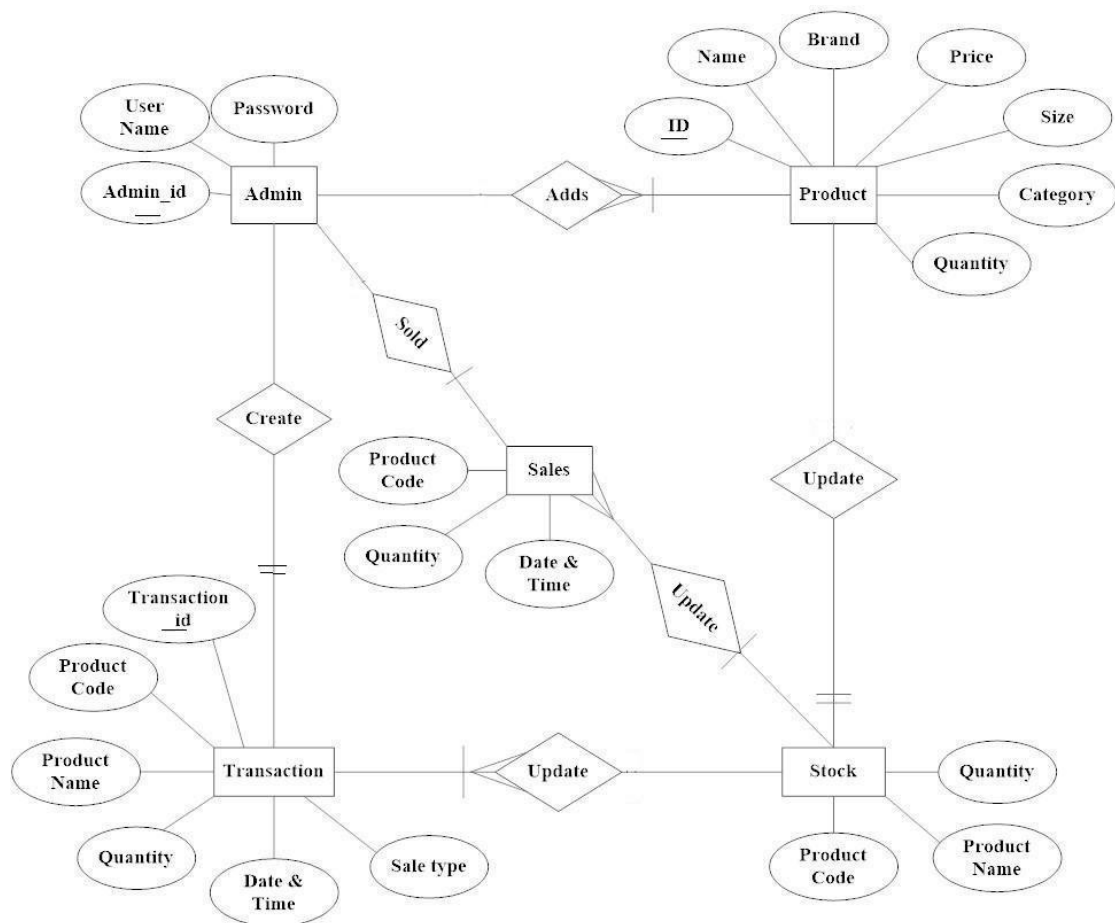


Figure 3.4: ER Diagram

3.5 Process Flow Diagram

A process flow diagram is a diagram commonly used in chemical and process engineering to indicate the general flow of plant processes and equipment.

The PFD displays the relationship between major equipment of a plant facility and does not show minor details such as piping details and designations.

A process flow diagram of the project is given below:

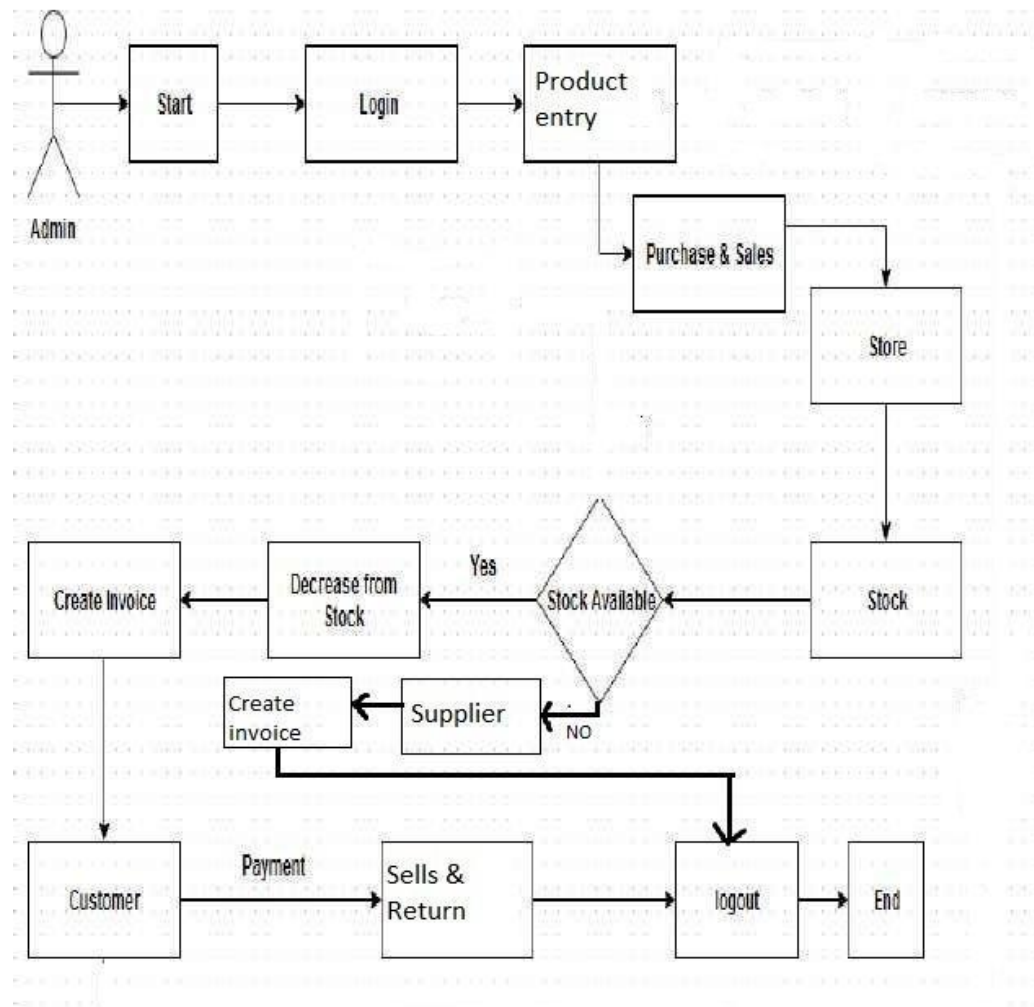


Figure 3.5: Process flow Diagram

3.6 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).^[3]

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

3.6.1 Level 0 DFD:

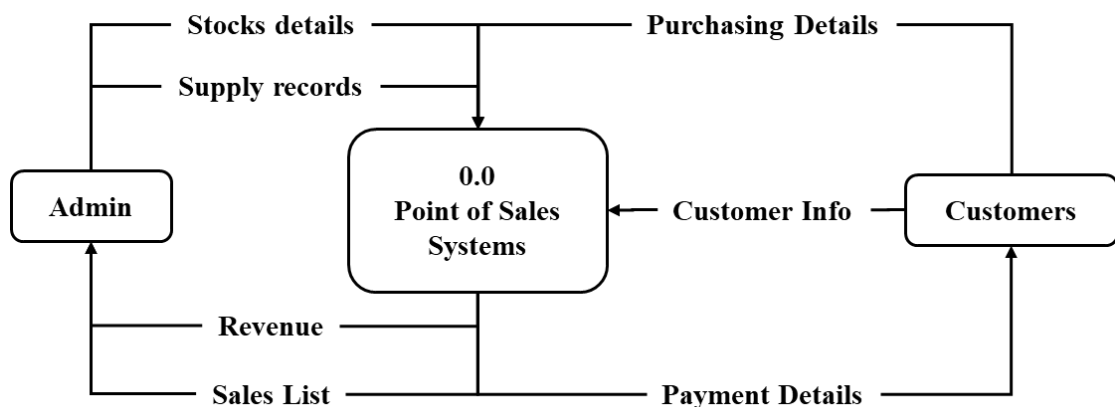


Figure 3.6.1: Level 0 DFD

3.6.2 Level 1 DFD:

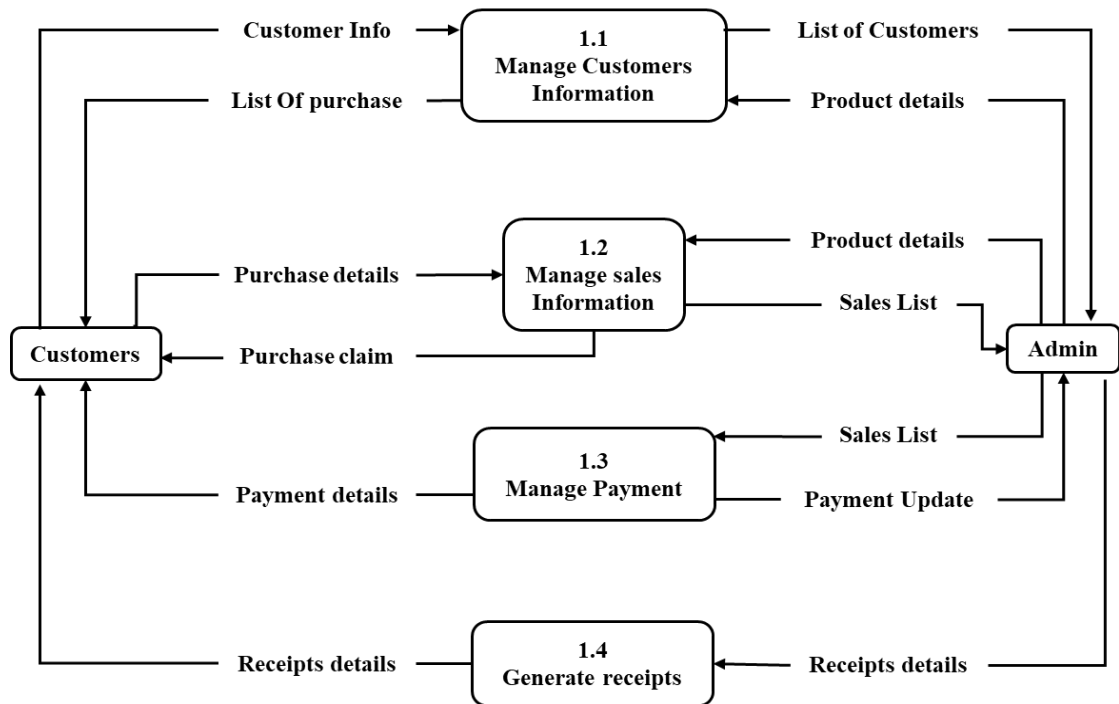
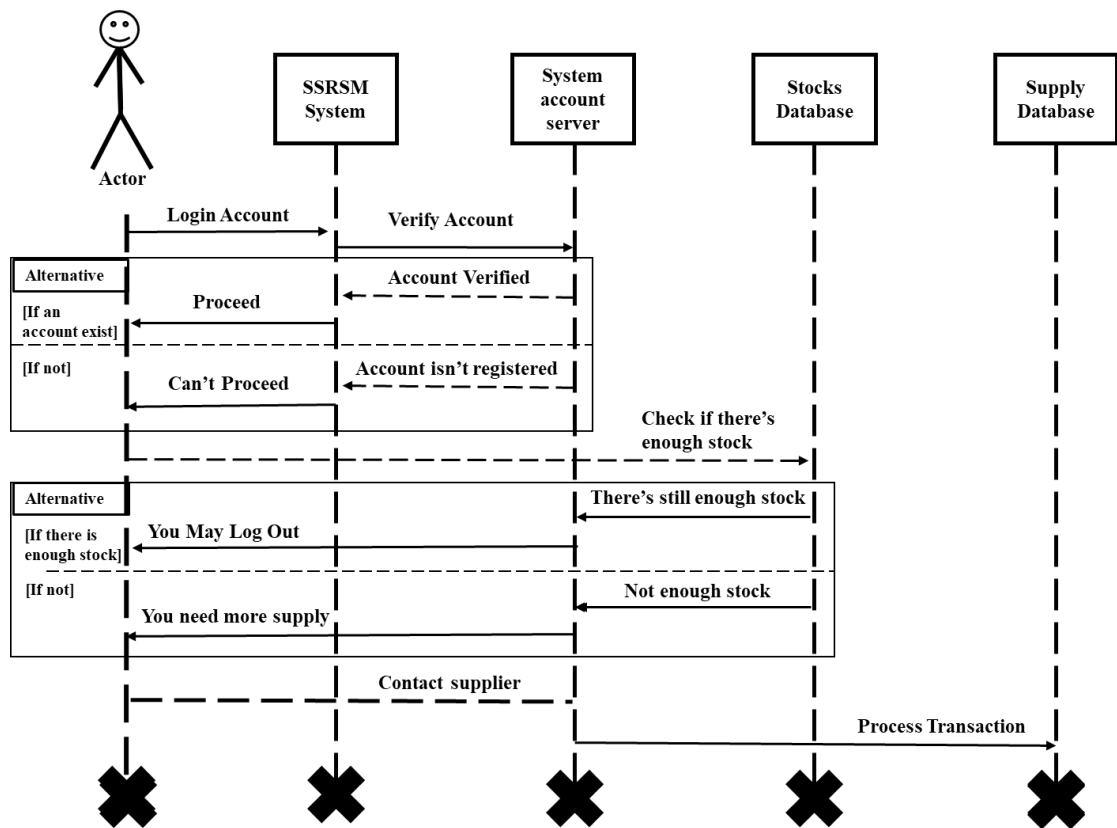


Figure 3.6.2: DFD 1 Level

3.7 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

Sequence Diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically).



1

Figure 3.7: Sequence Diagram

1

3.8 Conclusion:

So, in conclusion, we can say that system analysis is a problem solving strategy that includes glimpsing at the more extensive system, breaking the separated parts, and sorting out how it works to accomplish a specific object.



Chapter 4: Future Enhancement

4.1 Introduction:

Development of Business ERP for Entrepreneurs is on-going, and there is still much to do. The project team currently has a list of some enhancements planned for the future, varying from trivial to very complex. In order to get a computer base sale system to the customer as soon as possible, the entire system could not be delivered at once. A basic system to control sales and inventory will be delivered first. This section contains information on current limitations and upcoming enhancements over the capabilities of the basic system.

4.2 Limitations of Proposed Project:

This project has some limitations due to some technological unavailability and limited time frame. Some of the project limitations are:

- Only single admin panel is developed
- This application is not suitable for those organizations where there is a multi-level warehouse system available.

4.3 Future Enhancement

In this project we want to add some features in future. Some are given below – Making the system flexible in any type. .

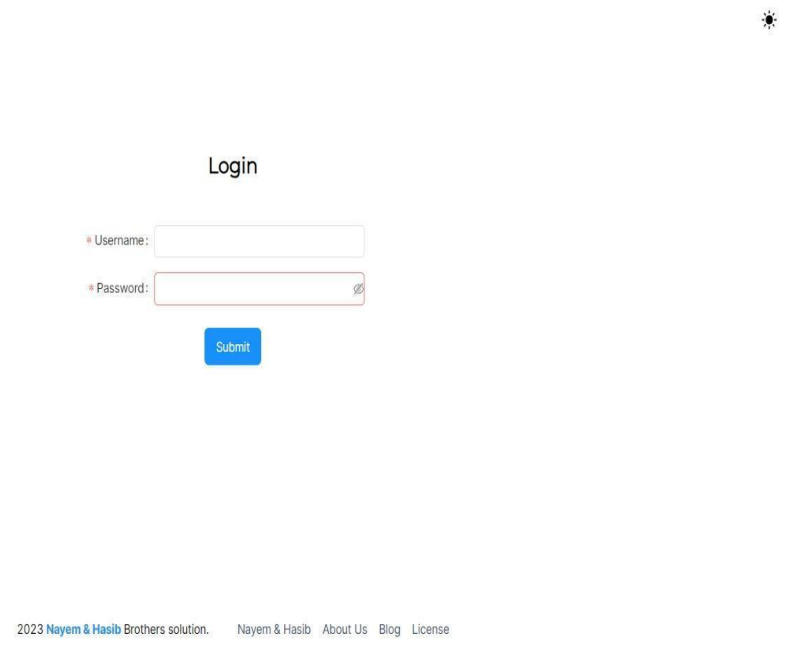
- Lost and breakage
- Interactive user interface design
- Showing important messages
- Showing Software's current Status
- A Timer



Chapter 5: Implementation

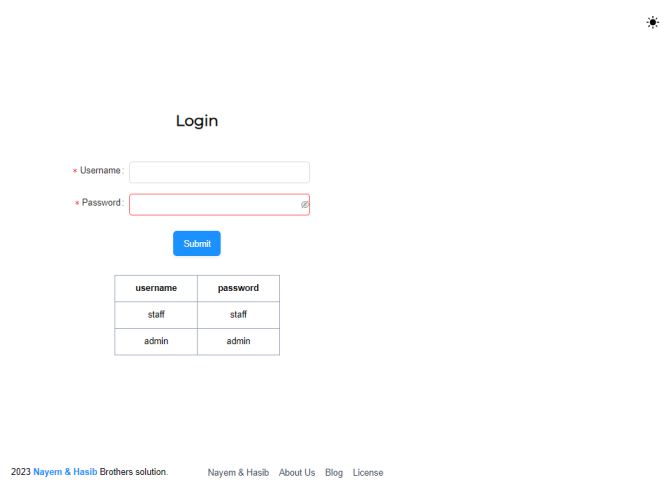
5.1 login:

Login Page is used to allow the user to login on the System.



The screenshot shows a login page with the title "Login" centered at the top. Below the title are two input fields: "Username:" and "Password:". The "Password:" field has a small eye icon on the right side. Below the input fields is a blue "Submit" button. At the bottom of the page, there is a footer with the text "2023 Nayem & Hasib Brothers solution." followed by links for "Nayem & Hasib", "About Us", "Blog", and "License".

Figure 5.1(a): Login Interface



The screenshot shows a login page similar to Figure 5.1(a), but with an additional table below the "Submit" button. The table has two columns: "username" and "password". The rows contain the following data:

username	password
staff	staff
admin	admin

The footer text is the same as in Figure 5.1(a): "2023 Nayem & Hasib Brothers solution." followed by links for "Nayem & Hasib", "About Us", "Blog", and "License".

Figure 5.1(b): Login Interface

5.2 Dashboard:

This is the main landing page of this application where a user can see some necessary incites over data that are created and collected throughout the use of this application. It will help the user to keep track of transaction analytics.

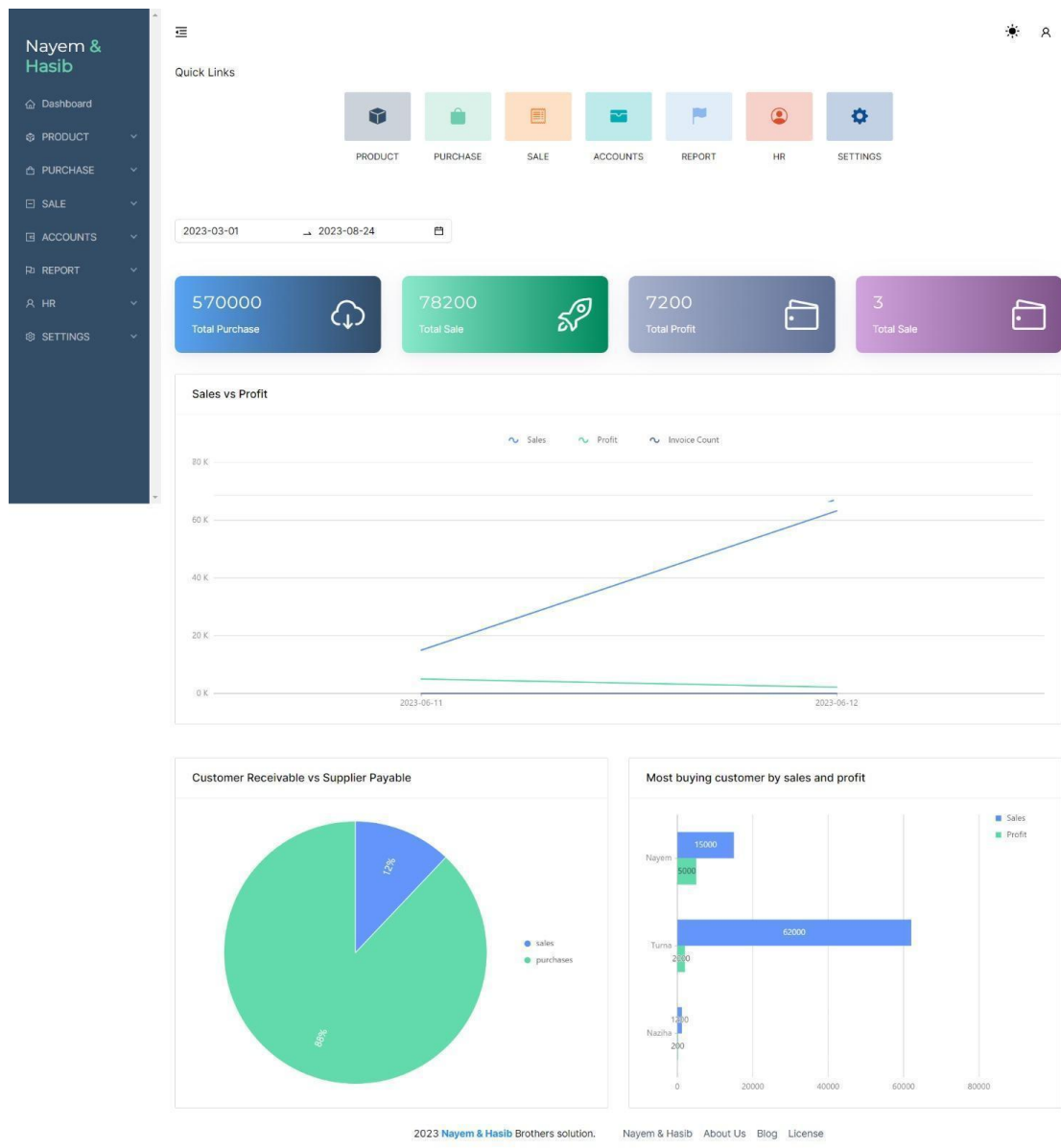


Figure 5.2: Dashboard Interface

5.3 Product Details:

Each individual product has its own individual detail log. They can be identified through many criteria such as name, product code, category, sub-category, unit, pricing etc.

The screenshot shows the 'Add Product' and 'Import From CSV' interface. The 'Add Product' form includes fields for Name, Select Subcategory, Select Brand, Select Unit Type, Unit Measurement, Quantity, Purchase Price, Sale Price, and Reorder Quantity. The 'Import From CSV' section has a 'Choose File' button and an 'Import From CSV' button.

Figure 5.3(a): Product Details

The screenshot shows the 'Products List' interface. It includes a 'Column Visibility' button, a 'DOWNLOAD CSV' button, and 'ACTIVE' and 'INACTIVE' status buttons. The table below lists product details:

Image	ID	SKU	Name	Unit Measurement	Quantity	Purchase price	Sale price	Sub Category	Brand	Unit Type	Reorder QTY	Action
	4	6780	No Longer Human	1	9	1000	1200	Books	English books	pc	1	Barcode
	3	4200	iphone 12 pro	1	5	60000	62000	Phone	apple	pc	2	Barcode
	2	12567	T-shirt	1	10	1000	1500	clothes	Gucci	pc	2	Barcode
	1	ABC123	Hp245	1	29	10000	15000	Laptop	HP	pc	5	Barcode

Figure 5.3(b): Product List

5.4 Supplier Details:

User can create supplier by giving name, phone number and address of supplier

The screenshot shows the 'Supplier Details' page. On the left is a dark sidebar with the logo 'Nayem & Hasib' and a menu with categories: Dashboard, PRODUCT (Products, Product Categ..., Product Subcat..., Product Brand), PURCHASE (Suppliers, New Purchase, Purchase List), SALE, and ACCOUNTS. The main content area has a 'Back' button at the top left. Below it are two panels: 'Add Supplier' with input fields for Name, Phone, and Address, and an 'Add Supplier' button; and 'Import From CSV' with a 'Choose File' button, 'No file chosen' text, and an 'Import From CSV' button. At the bottom is a 'Suppliers List' section with a 'Column Visibility' button and a 'DOWNLOAD CSV' button.

Figure 5.4: Supplier Details

5.5 Customer Details:

Users can accumulate client data or customer data in order to make sales or to verify if necessity arises. User can also see the list of customers. Users also can track transactions with each individual client using this information.

The screenshot shows the 'Customer Details' page. On the left is a dark sidebar with the logo 'Nayem & Hasib' and a menu with categories: Dashboard, PRODUCT, PURCHASE, SALE, and ACCOUNTS. The main content area has an 'Add Customer' form with input fields for Name, Phone, and Address, and an 'Add Customer' button. To the right is an 'Import From CSV' section with a 'Choose File' button, 'No file chosen' text, and an 'Import From CSV' button. Below these is a 'Customer List' section with a 'Column Visibility' button and a 'Download CSV' button. The list contains three entries:

ID	Name	Phone	Address
1	Nayem	01744195964	250/a,mirpur,shawrapara,dhaka
2	Turna	0997547	sony
3	Naziha	7754579	puran dhaka

Figure 5.5: Customer Details

5.6 Product Sale:

Tracking product sale details in an inventory system involves recording key information like product sold, quantity, price, customer data, and transaction date, enabling businesses to manage stock, assess sales trends, and make informed restocking decisions.

The screenshot shows a web-based interface for creating a new sale. On the left is a dark sidebar menu with options like 'Products', 'PURCHASE', 'SALE', 'ACCOUNTS', 'REPORT', and 'HR'. The main content area is titled 'Sale New Products'. It contains several input fields: 'Customer' (a dropdown menu), 'Date' (set to 2023-08-24), and 'Sales Person' (another dropdown). Below these is a table with columns: 'SL', 'Product', 'U.M', 'Quantity', 'Unit Price', and 'Total'. A '+ Add Product' button is positioned below the table. To the right of the table are summary fields: 'Total: 0 tk', 'Discount:', 'After Discount: 0 tk', 'Paid Amount:', and 'Due Amount: 0 tk'. At the bottom right is a blue 'Sale Product' button. The footer of the page reads '2023 Nayem & Hasib Brothers solution. Nayem & Hasib About Us Blog License'.

Figure 5.6: Product sale

5.7 Account and Transaction Details:

Important incites and reports are available through interactive database linking. Necessary data will be readily available for audit and balance sheet management.

The screenshot displays a 'Balance Sheet' report. On the left is the same sidebar menu as in Figure 5.6. The main content area shows the report title 'Balance Sheet' and a table with two columns: 'Account' and 'Amount'. The table is organized into three sections: 'Assets', 'Liabilities', and 'Equity'.
Assets:
Cash: -184346
Bank: 10000
Inventory: 609000
Accounts Receivable: 200
TOTAL: 434854
Liabilities:
Accounts Payable: 317654
TOTAL: 317654
Equity:
Capital: 110000
At the bottom of the table, there is a 'Settings' button.

Figure 5.7(a): Balance Sheet

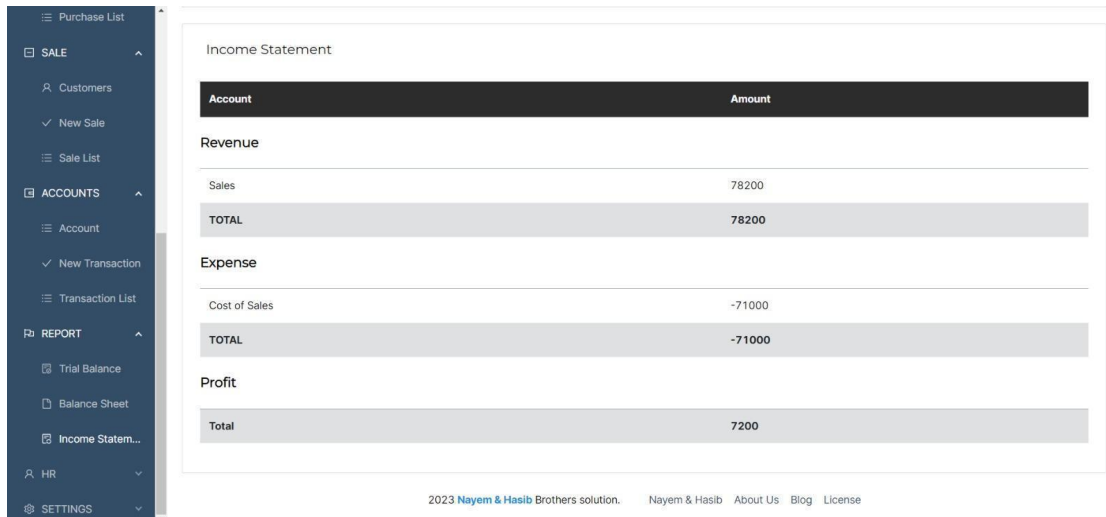


Figure 5.7(b): Revenue Sheet

5.8 Bill preparation and Invoice receipt:

A print dialogue box allows the user to print a receipt for the customer which contains purchase details with invoice id and client information. It allows both client and user to track the purchase information.



Figure 5.8: Invoice receipt

5.9 Human resources (HR):

HR creates staff profiles, specifying roles and permissions in the inventory system. This controls staff access to inventory functions like adding, editing, and viewing items, maintaining data integrity and security. Clear role assignment enhances efficiency and accountability in inventory management.

The screenshot shows the HR management interface. On the left is a dark sidebar with a menu including Dashboard, PRODUCT, PURCHASE, SALE, ACCOUNTS, REPORT, HR, and SETTINGS. The main content area has a 'Back' button at the top left. The 'Add New Staff' form contains the following fields:

- User Name:
- Password:
- Email:
- Joining Date:
- Leave Date:
- Role:
- Id No:
- Phone:
- Address:
- Salary:
- Blood Group:
- Designation:

Below the form is a blue 'Add New Staff' button. The 'Staff List' section below features a 'Download CSV' button, 'Active' and 'Inactive' status filters, and a 'Column Visibility' dropdown. The table below lists the staff members:

ID	Name	Role	Created at	Action
1	admin	admin	2023-06-10	View
2	staff	staff	2023-06-10	View
3	nayem	Staff_1	2023-06-11	View

At the bottom of the page, there is a footer with the text: '2023 Nayem & Hasib Brothers solution. Nayem & Hasib About Us Blog License'.

Figure 5.9(a): HR Staff Details

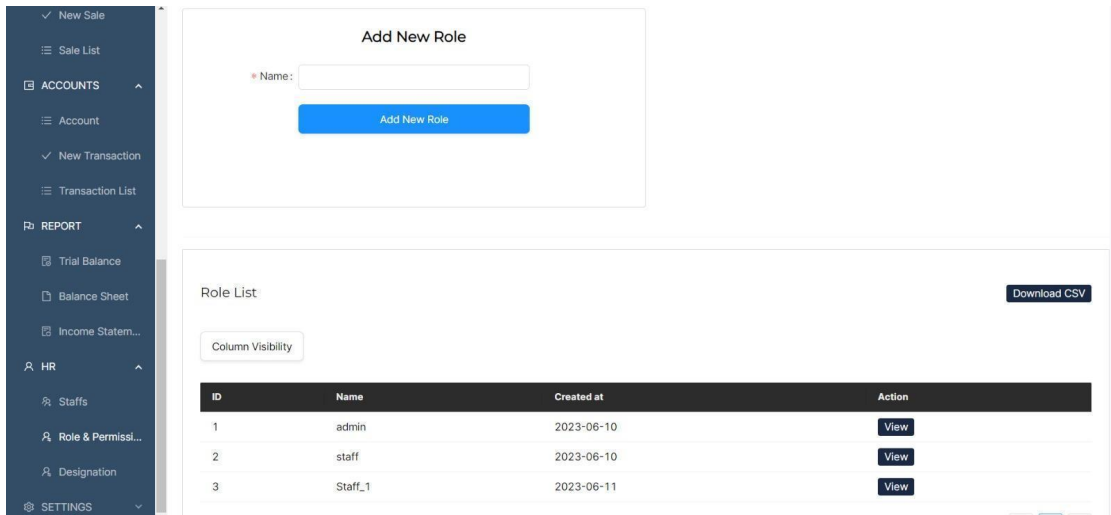


Figure 5.9(b): HR Role Details

5.10 Setup Menu:

This Menu allows a user to manage company/ organization/ shop information along with website details and Footer details.

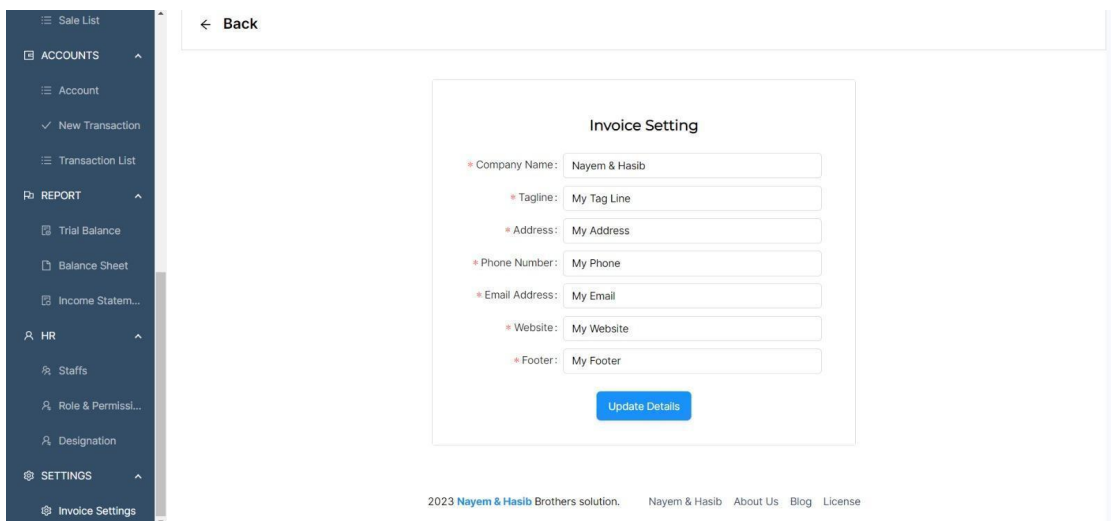


Figure 5.10: Setup Menu



Chapter 6: Conclusion

6.1 Conclusion:

In conclusion, the development and implementation of a Business ERP application tailored for entrepreneurs and new organizations present a pivotal solution in fostering their growth and success. This specialized ERP system addresses the unique challenges faced by startups, providing an integrated platform to streamline operations, enhance decision-making, and promote scalability. By offering modules for finance, inventory, sales, HR, and customer relationship management, this ERP application empowers entrepreneurs to efficiently manage their resources and processes.

The application's user-friendly interface and customizable features cater to the specific needs and limited resources of emerging businesses. Automation of routine tasks, real-time analytics, and data-driven insights enable entrepreneurs to allocate their time and energy towards strategic innovation rather than administrative burdens.

Through centralized data management and improved collaboration, this ERP application cultivates a holistic organizational view, which is crucial for prudent decision-making. As new businesses evolve rapidly, the scalability of the ERP system accommodates growing demands seamlessly. In a technology-driven landscape, embracing this ERP solution can provide entrepreneurs with a competitive edge and a solid foundation for sustainable success. Ultimately, by embracing this ERP application tailored to their unique needs, entrepreneurs and new organizations can embark on their journey with heightened efficiency, strategic foresight, and a trajectory towards long-term prosperity.



References

References:

- [1] Park, Jonghyun, and Jin Park. "Enterprise resource planning and efficiency." *Managerial Finance* 41, no. 4 (April 13, 2015): 405–15. <http://dx.doi.org/10.1108/mf-07-2014-0182>.
- [2] Sundtoft Hald, Kim, and Jan Mouritsen. "Enterprise resource planning, operations and management." *International Journal of Operations & Production Management* 33, no. 8 (July 12, 2013): 1075–104. <http://dx.doi.org/10.1108/ijopm-11-2011-0430>.
- [3] Greaves, Ian. "An Investigation of Enterprise Resource Planning (ERP) and Supporting Technologies." NSUWorks, 2011. http://nsuworks.nova.edu/gscis_etd/168.
- [4] Fakhroutdinov, K., 2023. *Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors)..* [online] Uml-diagrams.org. Available at: <<https://www.uml-diagrams.org/use-case-diagrams.html>> [Accessed 28 March 2023].
- [5] Albers, Jason. "Bill of material testing for enterprise resource planning (ERP) implementation." Menomonie, WI : University of Wisconsin--Stout, 2004. <http://www.uwstout.edu/lib/thesis/2004/2004albersj.pdf>
- [6] Smartdraw.com. 2023. *Entity Relationship Diagram (ERD) - What is an ER Diagram?.* [online] Available at: <<https://www.smartdraw.com/entity-relationship-diagram/>> [Accessed 28 March 2023].
- [7] Loss Prevention Media. 2023. *Understanding Essential POS Features and Limitations.* [online] Available at: <<https://losspreventionmedia.com/understand-your-pos-features-and-limitations/>> [Accessed 28 March 2022].



Appendix

Dashboard:

```
import { Affix, Drawer, Layout } from "antd";

import React, { useEffect, useState } from "react";

import { useLocation } from "react-router-dom";

import Sidenav from "../Sidenav/Sidenav";

import Footer from "./Footer";

import Header from "./Header";

import styles from "./Main.module.css";

import QuickLinks from "./QuickLinks";

const { Header: AntHeader, Content, Sider } = Layout;

function Main({ children }) {

  const [collapsed, setCollapsed] = useState(false);

  const [sideNavOpenKeys, setSideNavOpenKeys] = useState("");

  const sideNavOpenKeysHandler = (val) => {

    setSideNavOpenKeys(val);

  };

  const handleCollapsed = (val) => {

    setCollapsed(val);

  };

  const [visible, setVisible] = useState(false);

  const [placement, setPlacement] = useState("right");

  const [sidenavColor, setSidenavColor] = useState("#1890ff");

  const [sidenavType, setSidenavType] = useState("transparent");
```

```

const [fixed, setFixed] = useState(false);
const openDrawer = () => setVisible(!visible);
const handleSidenavType = (type) => setSidenavType(type);
const handleSidenavColor = (color) => setSidenavColor(color);
const handleFixedNavbar = (type) => setFixed(type);
let { pathname } = useLocation();
pathname = pathname.replace("/", " ");
useEffect(() => {
  if (pathname === "rtl") {
    setPlacement("left");
  } else {
    setPlacement("right");
  }
}, [pathname]);
const isLoggedIn = Boolean(localStorage.getItem("isLoggedIn"));
return (
  <Layout className={styles.mainLayout}>
    {isLoggedIn && (
      <Drawer
        title={false}
        placement={placement === "right" ? "left" : "right"}
        closable={false}
        onClose={() => setVisible(false)}
        visible={visible}
        key={placement === "right" ? "left" : "right"}
        width={220}>

```

```

<Layout>
  <Sider
    trigger={null}
    width={220}
    theme="light"
    className={styles.siderDrawer}>
    <Sidenav color={sidenavColor} sideNavOpenKeys={sideNavOpenKeys} />
  </Sider>
</Layout>
</Drawer>
)}
{isLoggedIn && (
  <Sider
    breakpoint="lg"
    trigger={null}
    collapsible
    collapsed={collapsed}
    width={220}
    theme="light"
    className={styles.siderMain}>
    {collapsed ? (
      ""
    ) : (
      <div>
        <h3 className="text-white ms-4 mt-2 mb-1 ">
          Nayem{" "}

```

```

        <strong style={{ color: "#6ECCAF", fontWeight: "bold" }}>
            & Hasib
        </strong>
    </h3>
</div>
))
{isLoggedIn && (
    <Sidenav color={sidenavColor} sideNavOpenKeys={sideNavOpenKeys} />
)}
</Sider>
))
<Layout
    className={
        isLoggedIn
            ? collapsed
            ? styles.mainLayoutUncollapse
            : styles.mainLayoutCollapse
            : styles.mainLayoutMarginLeftZero
    }>
    {fixed ? (
        <Affix>
            <AntHeader>
                <Header
                    onPress={openDrawer}
                    name={pathname}
                    subName={pathname}

```

```

        handleSidenavColor={handleSidenavColor}
        handleSidenavType={handleSidenavType}
        handleFixedNavbar={handleFixedNavbar}
        collapsed={collapsed}
        handleCollapsed={handleCollapsed}
        isLoggedIn={isLoggedIn}
    />
</AntHeader>
</Affix>
):(
<AntHeader>
<Header
    onPress={openDrawer}
    name={pathname}
    subName={pathname}
    handleSidenavColor={handleSidenavColor}
    handleSidenavType={handleSidenavType}
    handleFixedNavbar={handleFixedNavbar}
    collapsed={collapsed}
    handleCollapsed={handleCollapsed}
/>
</AntHeader>
)}
{isLoggedIn &&
(pathname.trim() === "dashboard" || pathname.trim() === "") && (
<QuickLinks

```



```

        sideNavOpenKeys={sideNavOpenKeys}
        sideNavOpenKeysHandler={sideNavOpenKeysHandler}
      />
    )}
    <Content>{children}</Content>
    <Footer />
  </Layout>
</Layout>
);
}
export default Main;

```

Product:

```

import "bootstrap-icons/font/bootstrap-icons.css";
import { useState } from "react";
import { Link } from "react-router-dom";
import "./product.css";
import { Button, Dropdown, Menu, Segmented, Table } from "antd";
import { useEffect } from "react";
import { CSVLink } from "react-csv";
import { useDispatch, useSelector } from "react-redux";
import getTotalProduct from "../../api/getAllApis/getTotalProduct";
import { loadProduct } from "../../redux/actions/product/getAllProductAction";
import GenerateBarcodePopUp from "./generateBarcodePopUp";
function CustomTable({ list, total, status }) {
  const dispatch = useDispatch();

```

```

const [columnItems, setColumnItems] = useState([]);
const [columnsToShow, setColumnsToShow] = useState([]);
const columns = [
  {
    title: "Image",
    dataIndex: "thumbnail_image_url",
    render: (thumbnail_image_url) => (
      <img
        style={{ maxWidth: "40px" }}
        alt='product'
        src={thumbnail_image_url}
      />
    ),
  },
  {
    title: "ID",
    dataIndex: "id",
    key: "id",
  },
  {
    title: "SKU",
    dataIndex: "sku",
    key: "sku",
  },
  {
    title: "Name",

```

```

        dataIndex: "name",
        key: "name",
        render: (name, { id }) => <Link
to={`/product/${id}`}>{name}</Link>,
    },
    {
        title: "Unit Measurement",
        dataIndex: "unit_measurement",
        key: "unit_measurement",
    },
    {
        title: "Quantity",
        dataIndex: "quantity",
        key: "quantity",
    },
    {
        title: "Purchase price",
        dataIndex: "purchase_price",
        key: "purchase_price",
        responsive: ["md"],
    },
    {
        title: "Sale price",
        dataIndex: "sale_price",
        key: "sale_price",
        responsive: ["md"],
    }

```

```

    },
    {
        title: "Sub Category",
        dataIndex: "product_sub_category",
        key: "product_sub_category",
        render: (product_sub_category) =>
product_sub_category?.name,
    },
    {
        title: "Brand",
        dataIndex: "product_brand",
        key: "product_brand",
        render: (product_brand) => product_brand?.name,
    },
    {
        title: "Unit Type",
        dataIndex: "unit_type",
        key: "unit_type",
    },
    {
        title: "Reorder QTY",
        dataIndex: "reorder_quantity",
        key: "reorder_quantity",
    },
    {

```

```

        title: "Action",
        dataIndex: "sku",
        key: "sku",
        render: (sku, quantity) => <GenerateBarcodePopUp sku={sku ?
sku : 0} />,
    },
];

```

```

useEffect(() => {
    setColumnItems(menuItems);
    setColumnsToShow(columns);
}, []);

```

```

const colVisibilityClickHandler = (col) => {
    const ifColFound = columnsToShow.find((item) => item.key ===
col.key);
    if (ifColFound) {
        const filteredColumnsToShow = columnsToShow.filter(
            (item) => item.key !== col.key
        );
        setColumnsToShow(filteredColumnsToShow);
    } else {
        const foundIndex = columns.findIndex((item) => item.key ===
col.key);
        const foundCol = columns.find((item) => item.key ===
col.key);
        let updatedColumnsToShow = [...columnsToShow];

```

```

        updatedColumnsToShow.splice(foundIndex, 0, foundCol);
        setColumnsToShow(updatedColumnsToShow);
    }
};

const menuItems = columns.map((item) => {
    return {
        key: item.key,
        label: <span>{item.title}</span>,
    };
});

const addKeys = (arr) => arr.map((i) => ({ ...i, key: i.id }));

return (
    <div>
        <div>
            {list && (
                <div style={{ marginBottom: "30px" }}>
                    <Dropdown
                        overlay={
                            <Menu
                                onClick={colVisibilityClickHandler} items={columnItems} />
                            >
                        }
                        placement='bottomLeft'>
                            <Button>Column
                                Visibility</Button>
                    </Dropdown>
                </div>
            )}
        </div>
    </div>
);

```

```

        </div>
    })
</div>
<div>
    <Table
        scroll={{ x: true }}
        loading={!list}
        pagination={{
            defaultPageSize: 10,
            pageSizeOptions: [10, 20, 50, 100, 200],
            showSizeChanger: true,
            total: total,

            onChange: (page, limit) => {
                dispatch(loadProduct({ page,
limit, status }));
            },
        }}
        columns={columnsToShow}
        dataSource={list ? addKeys(list) : []}
    />
</div>
</div>
);
}

```

```

const GetAllProd = (props) => {
  const dispatch = useDispatch();
  const list = useSelector((state) => state.products.list);
  const [total, setTotal] = useState(0);

  useEffect(() => {
    dispatch(loadProduct({ status: "true", page: 1, limit: 10 }));
  }, []);

  //TODO :IMPLEMENT TOTAL PROD info
  useEffect(() => {
    getTotalProduct().then((res) => setTotal(res));
  }, [list]);

  const [status, setStatus] = useState("true");
  const onChange = (value) => {
    setStatus(value);
    dispatch(loadProduct({ status: value, page: 1, limit: 10 }));
  };

  const CSVlist = list?.map((i) => ({
    ...i,
    product_category: i?.product_category?.name,
  }));

  return (

```



```

<div className='card column-design'>
  <div className='card-body'>
    <h5>Products List</h5>
    {list && (
      <div className='card-title d-flex
justify-content-end'>
        <div className='me-2' style={{
marginTop: "4px" }}>
          <CSVLink
            data={CSVlist}
            className='btn btn-dark
btn-sm mb-1'
            filename='products'>
            Download CSV
          </CSVLink>
        </div>
        <div>
          <Segmented
            className='text-center
rounded danger'
            size='middle'
            options={[
              {
                label: (
<span>
          <i className='bi bi-person-dash-fill'></i> Inactive

```

```
                <CustomTable list={list} total={total} status={status}
/>
            </div>
        </div>
    );
};
export default GetAllProd;
```