**3 System Design**

* 1. **Introduction**
* System design is the process of defining the architecture, module interfaces and data for a system to satisfy specified requirements.
* The purpose of the design phase is to plan the solution of the problem specified by the requirement documents.
* This is the first step that moving from problem domain to the solution domain.
* The design of the system is essentially a blueprint or a plan for a solution for the system.

**3.2 Assumption and constraints**

* For windows 11 1 gigahertz (GHz) or faster with 2 or more cores on a compatible 64-bit processor or System on a Chip (SoC). 4 gigabytes (GB). 64 GB or larger storage device.
* The system is dependent on the availability of an Apache Tomcat Server to run.
* 3 months of time for completion.

**3.3 Functional Decomposition**

Functional decomposition is the process of taking a complex process and breaking it down into its smaller, simpler parts. Using functional decomposition large or complex functionalities are more easily understood. It is mainly used during project analysis phase, so each phase can be viewed as software. So, this has modular with some sub modules.

* + 1. **System Software Architecture**

Dealer authorization

Login

Customer Registration

Registration

Dealer Registration

Product management

Admin

Add

Update

Active/Inactive

SellGross

Dealer allotment

Add

Region management

Update

Active/Inactive

View Feedback

Pending order

Dealer

Previous order

Bill generation

Modify cart

Customer

Cart

Place order

Previous order

View bill

Feedback

Pending order

**3.3.2 System technical architecture**

Internet

Chrome client

Xampp server

MySQL Database server

Mozilla client

Presentation Tier

Application Tier

Data Management Tier

Printer

Computer

Hard disk or storage device

Keyboard and mouse

**3.3.3 System hardware architecture**

**3.3.4 External Interface**

Not applicable

* 1. **Description of programs**
     1. **Context Flow Diagram**

In CFD entire system is considered as a single process. Context flow diagram shows input and outputs of the system. It shows all the external entities that interact with the system and how the data flows between these external entities and system.

Order placement

Registration

Customer

Registration

Feedback

Bill view

Admin

Bill generates

Dealer

Products, Regions, Dealers

Messages

Active orders

Previous orders

* + 1. **Data Flow Diagram**

Data flow diagram shows the flow of data through system. Data flow diagrams also called the data flow graphs. It views a system as a function that transforms the inputs into desired outputs. It aims to capture the transformation that taken place within a system to the input data so that eventually the output data is produced.

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
|  | Process | It performs transformation of data from one state to another. |
|  | Sink/Source | It represents the external entity that may be either source or sink. |
|  | Flow of data | It represents the flow of data from source to destination. |
|  | Data Source/Data storage | It is the place where data is stored. |

**Top level DFD**

Users

* 1. **Description of the components**
     1. **Login module**
        1. **Input**

User name and password

* + - 1. **Process**

Read details and validation

* + - 1. **Output**

Logged in to their respective pages.

* + - 1. **Interface with another functional components**

Independent

* + - 1. **Resource allocation**

Admin/Dealer/Customer table

* + - 1. **User interface**

Textboxes are provided to enter the username and password. Login button is provided to move to next page.

Display message

Admin/Dealer/Customer

User name

Password

Admin/Dealer/Customer

Valid

Invalid

**3.5.2 Registration module**

Registration

Customer registration

Dealer registration

* + - 1. **Customer Registration**
         1. **Input**

Registration details of the customer.

**3.5.2.1.2 Process**

Read customer registration details

* + - * 1. **Output**

Registration successful message view be display and stored in database.

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Customer table

* + - * 1. **User interface**

Textboxes are provided to enter the details. Register button is provided to store the details.

Valid data

Customer

Customer

Invalid

Customer details

Show message

* + - 1. **Dealer Registration**
         1. **Input**

Registration details of the Dealer.

**3.5.2.1.2 Process**

Read Dealer registration details

* + - * 1. **Output**

Registration successful message view be display and stored in database.

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Dealer table

* + - * 1. **User interface**

Textboxes are provided to enter the details. Register button is provided to store the details.

Valid data

Dealer

Dealer

Invalid

Customer details

Show message

**3.5.3 Admin module**

Admin

* + - 1. **Product Management**

Product Management

* + - * 1. **Add**

**Input**

Product Details- name, image, price, description, stock, Unit.

**Process**

Validates the products and Store to database

**Output**

Show successful message

**Interface with another component**

Independent

**Resource allocation**

Product table

**User Interface**

Textboxes are provided for entering product details. Add products button is provided to add product. After add

button he will see back/return button to back to administrative page.

Invalid

Read details

Admin

Valid

Products

* + - * 1. **Update**

**Input**

Enter Product Details- name, image, price, description, stock, Unit to be updated.

**Process**

Validate input and update database

**Output**

Changes are updated in database and successful message will be displayed.

**Interface with another components**

Independent

**Resource allocation**

Product table

**User interface**

Update button will be provided for update details. Textboxes are provided for updating details. Update button will help to save details to database.

Message

Updated data

Admin

Product

Product selected

Display

Load

* + - * 1. **Active/Inactive**

**Input**

Product ID

**Process**

Product will not be available to use.

**Output**

product will be disabled and product status will be updated

**3.4.3.1.3.4 Interface with other components**

Independent

**Resource allocation**

Product table

**User Interface**

Item list will be displayed in the form of list. To active/inactive the product, set as active/inactive button is be provided.

Generate message

Click event

Admin

Product

Load

Product\_status

Display

* + - 1. **Dealer Authorization**
         1. **Input**

Dealer details.

* + - * 1. **Process**

Selection process and update table

* + - * 1. **Output**

Successful message.

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Dealer table

* + - * 1. **User interface**

Approve and reject button is provided to authenticate.

Dealer details

Admin

Dealer details

* + - 1. **Dealer Allotment**
         1. **Input**

Dealer details and item order details.

* + - * 1. **Process**

Dealer is allotted to orders and updated in database.

* + - * 1. **Output**

Successful message

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Dealer table and order table.

* + - * 1. **User interface**

Dropdown list for selecting dealer and allot button to allot dealer.

Admin

Item order

Dealer

Dealer id

Item order details

Item order details

Dealer id

Message

* + - 1. **Region Management**

Region management

* + - * 1. **Add**

**Input**

Region details- name, pin code

**3.5.3.4.1.2 Process**

Validation for existence

**Output**

Region will be added to database and success message will be shown.

**Interface with other components**

Independent

**3.5.3.4.1.5 Resource allocation**

Region table

**3.5.3.4.1.6** **User interface**

Textboxes are provided for adding region details. By clicking add button region will be added to table.

Invalid

Product

Admin

Details

Valid data

Region

* + - * 1. **Update**

**Input**

Region details- name, pin code

**Process**

Update operation and store to database

**Output**

Successful message

**Interface with other components**

Independent

**Resource allocation**

Region table

**User interface**

Textboxes are provided for entering details. update button will be provided to update database.

Updated

Database

Admin

Region table

Load regions

Modified

Show message

Region selected

* + - * 1. **Active/Inactive**

**Input**

Region id

**Process**

Region\_status update operation and update database

**Output**

Successful message

**Interface with another functional components**

Independent

**Resource allocation**

Region table

**User interface**

Region details will be displayed on the screen. and set as active/inactive button will be provided for active/inactive operation.

Updated

Database

Display

Update

Operation

Details

Load

Admin

Region table

Success message

* + - 1. **View feedback**

* + - * 1. **Input**

Feedback\_id

* + - * 1. **Process**

Feedback of the order will be displayed which is retrieved from feedback table.

* + - * 1. **Output**

Successful message

* + - * 1. **Interface with another functional components**

Independent

**3.4.2.4.5 Resource allocation**

Feedback table

**3.4.2.4.6 User interface**

Feedback from the user will be display for respective orders

Admin

Feedback

Feedback details

* + 1. **Dealer**

Dealer

* + - 1. **Pending orders**
         1. **Input**

Order id

* + - * 1. **Process**

Retrieve pending order details

* + - * 1. **Output**

Pending orders will be displayed

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Item order table

* + - * 1. **User interface**

Shows all pending order.

Pending orders

Button click

Dealer

Order

Load

* + - 1. **Previous orders**
         1. **Input**

click

* + - * 1. **Process**

Retrieving order details

* + - * 1. **Output**

Orders will be displayed

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Item order table

* + - * 1. **User interface**

Shows all proceeded order.

Button click

Dealer

Order

Load

* + - 1. **Bill generation**
         1. **Input**

Order details

* + - * 1. **Process**

Bill generation

* + - * 1. **Output**

Bill will be displayed

* + - * 1. **Interface with another functional components**

Independent

* + - * 1. **Resource allocation**

Cart table, Item order table, Bill table.

* + - * 1. **User interface**

After clicking on generate bill button the bill will be generated and displayed on the screen.

Display

Generated bill

Dealer

Item order

Bill

Cart

* + 1. **Customer**

Customer

Pending order

Previous order

Cart

* + - 1. **Cart**

Cart

* + - * 1. **Modify cart**

**Input**

Button clicks

**Process**

Store details to cart

**Output**

Successful message

**Interface with another functional component**

Independent

**User interface**

It shows all products which are in cart.

Product id

Customer

Cart

Product

quantity

* + - * 1. **Place order**

**Input**

Order details

**Process**

Generate order\_id and store it to order table

**Output**

Successful message

**Interface with another functional component**

Independent

**User interface**

Button is used to place order.

Show message

Order id

Cart details

Load

Customer

Cart

Order details

Order

* + - 1. **Previous order**

Previous order

* + - * 1. **View bill**

**Input**

Button clicks

**Process**

Retrieve bill from table

**Output**

Display bill

**Interface with another functional component**

Independent

**User Interface**

View button is given to view bill for previous order.

Button clicks

Customer

Bill

Bill details

* + - * 1. **Feedback**

**Input**

Entered feedback

**Process**

Store to feedback table

**Output**

Show successful message

**Interface with another functional component**

Independent

**User Interface**

Feedback button is given to write feedback.

Button clicks

Button clicks

Customer

**s**

Feedback

Store

Show message

* + - 1. **Pending order**
         1. **Input**

Button clicks

* + - * 1. **Process**

Retrieve previous orders

* + - * 1. **Output**

Display active orders

* + - * 1. **Interface with another functional component**Independent
        2. **User Interface**

Button is given to view pending order.

Load

Active order details

order

Customer