**5. Detail Design**

**5.1 Introduction**

During detailed design, the internal logic of each module specified in system design is decided. During this phase further details of the modules are decided. Design of each of the modules usually specified in a high-level description language which is independent of the language in which software eventually be implemented.

**5.2 Structure of the system**

Login

Customer Registration

Registration

Dealer Registration

Product management

Admin

Add

Update

Active/Inactive

SellGross

Dealer authorization

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

**5.3 Module Description**

**Structure Chart**

Structure chart is a top-down modular design, consist of squares representing different models in a systems and lines**.** Structure chart shows how program has been partitioned into manageable modules hierarchy and organization of those modules and communicational interface.

|  |  |  |
| --- | --- | --- |
| Symbol | Name | Process |
|  | Data Flow | Shows the direction flow of data |
|  | Control Flow | Shows the direction flow of control |
|  | Processing | Shows manipulation, calculation and processing |
|  | Module invocation | It represents subordinate module invoked by super ordinate module |
| Main  A  B  C | Conditional invocation | It indicates the invocation of sub-ordinates. Module depends on the evaluation of condition |
| Main  A  B | Invocation | It represents the repetition |

**Flow Chart**

|  |  |  |
| --- | --- | --- |
| Symbol | Name | Purpose |
|  | Terminator | It indicates the start and end of the process |
|  | Input/Output | Input/Output data |
|  | Decision | It represents the comparison or  question that determines an alternate path to be followed |
|  | Flow direction | Shows the direction of data flow |
|  | Processing | It represents manipulation, calculation or information processing |
|  | Direction action storage | File storage |
|  | Preparation (Looping) | An instruction or looping |
|  | In page |  |
|  | Off page |  |
|  | Delay |  |

A flowchart is a graphically representation of the structure of process or system, algorithm or the step-by-step solution of the problem. Flowchart describes the flow of data through an information processing system and the parts of the flows. The flow is a set of the logic operations that meet the certain requirements.

**5.3.1 Login**

**5.3.1.1 Input**

Username and password

**5.3.1.2 Procedural Details**

Login

Display respective page

Authenticate

Input details

Read username and password

Display error message

Validate

Admin/Dealer

Customer

Load page

Load page

Order table

Product table

**5.3.1.3 File I/O Interfaces**

Admin/Dealer/Staff table

**5.3.1.4 Output**

Login and display respective pages

**5.3.1.5 Implementation aspects (if any)**

Textboxes, Buttons and Tabs

**5.3.2 Registration**

**5.3.2.1 Customer**

**5.3.2.1.1 Input**

Customer registration details

**5.3.2.1.2 Procedural Details**

Customer registration

Input customer details

Validation

Display successful message and load next page

Store to customer table

Display Error message

**5.3.2.1.3 File I/O Interfaces**

Customer table

**5.3.2.1.4 Output**

Add details to customer table and show message

**5.3.2.1.5 Implementation aspects (if any)**

Textboxes, Dropdown list, Buttons

**5.3.2.2 Dealer**

**5.3.2.1.1 Input**

Dealer registration details

**5.3.2.1.2 Procedural Details**

Dealer registration

Input dealer details

Validation

Display successful message and load next page

Store to dealer table

Display Error message

**5.3.2.1.3 File I/O Interfaces**

Dealer table

**5.3.2.1.4 Output**

Details will be stored and successful message will be displayed

**5.3.2.1.5 Implementation aspects (if any)**

Textboxes, Buttons and Dropdown list

**5.3.3 Admin**

**5.3.3.1 Product Management**

**5.3.3.1.1 Add**

**5.3.3.1.1.1 Input**

Product details

**5.3.3.1.1.2 Procedural Details**

Start

Product Details

Validation

Product

Display error message

If exists

Store to database

Product added successfully message

Stop

True

False

**5.3.2.1.1.3 File I/O Interfaces**

Product table

**5.3.2.1.1.4 Output**

Store the product details and show successful message

**5.3.2.1.1.5 Implementation aspects (if any)**

Textboxes, and Buttons

**5.3.3.1.2 Update**

**5.3.3.1.2.1 Input**

Updated product details

**5.3.3.1.2.2 Procedural Details**

Start

Product

Click on update

Make changes

Click on update product

Update database

Stop

Show successful message

**5.3.3.1.2.3 File I/O Interfaces**

Product table

**5.3.3.1.2.4 Output**

Changes will be updated

**5.3.3.1.2.5 Implementation aspects (if any)**

Textboxes, and Buttons

**5.3.3.1.3 Active/Inactive**

**5.3.3.1.3.1 Input**

Active/Inactive text from buttons

**5.3.3.1.3.2 Procedural Details**

Step:1 Start

Step:2 CLICK on set as

active/inactive button

Step3: IF active THEN

UPDATE product table SET status as active

ELSE IF inactive THEN

UPDATE product table SET status as inactive

Step4: DISPLAY message as product

status has been changed

Step5: END

**5.3.3.1.3.3 File I/O Interfaces**

Product table

**5.3.3.1.3.4 Output**

Successful message

**5.3.3.1.3.5 Implementation aspects (if any)**

Buttons

**5.3.3.2 Dealer Authorization**

**5.3.3.2.1 Input**

Dealer details

**5.3.3.2.2 Procedural Details**

Step1: Start

Step2: retrieve dealer details where status is pending

Step3: CLICK on accept or reject

Step4: IF accept THEN

SET status as accept

END IF

IF reject THEN

SET status as reject

END IF

Step5: END

**5.3.3.2.3 File I/O Interfaces**

Dealer table

**5.3.3.2.4 Output**

Update dealer status and show message

**5.3.3.2.5 Implementation aspects (if any)**

Buttons

**5.3.3.3 Dealer Allotment**

**5.3.3.3.1 Input**

Dealer details and item order details

**5.3.3.3.2 Procedural Details**

Step1: Start

Step2: CLICK on DEALER ALLOTMENT

Step3: retrieve orders from order table where dealer\_id is NULL

Step4: retrieve order details from order details table

Step5: retrieve dealer\_id from dealer table where region\_id in dealer\_id

Step6: SHOW orders and dealer\_id

Step7: SELECT dealer\_id from dropdown list

Step8: CLICK on allot button

Step9: assign dealer\_id to orders and UPDATE

dealer\_id in order table

Step10: END

**5.3.3.3.3 File I/O Interfaces**

Dealer table and item order table

**5.3.3.3.4 Output**

Successful message

**5.3.3.3.5 Implementation aspects (if any)**

Dropdown list, buttons

**5.3.3.4 Region Management**

**5.3.3.4.1 Add**

**5.3.3.4.1.1 Input**

Region details

**5.3.3.4.1.2 Procedural Details**

Start

Region Details

Validation

Region

Display error message

If exists

Store to database

Region added successfully message

Stop

True

False

**5.3.3.4.1.3 File I/O Interfaces**

Region table

**5.3.3.4.1.4 Output**

Region will be stored and successful message will be displayed

**5.3.3.4.1.5 Implementation aspects (if any)**

Textboxes and Buttons

**5.3.3.4.2 Update**

**5.3.3.4.2.1 Input**

Updated region details

**5.3.3.4.2.2 Procedural Details**

Start

Region

Click on update

Make changes

Click on update region

Update database

Stop

Show successful message

**5.3.3.4.2.3 File I/O Interfaces**

Region table

**5.3.3.4.2.4 Output**

Region updated message

**5.3.3.4.2.5 Implementation aspects (if any)**

Textboxes and Buttons

**5.3.3.4.3 Active/Inactive**

**5.3.3.4.3.1 Input**

Region id

**5.3.3.4.3.2 Procedural Details**

Step1: Start

Step2: CLICK on active/inactive button

Step3: IF active THEN

UPDATE region table SET status as active

ELSE IF inactive THEN

UPDATE region table SET status as active

Step4: DISPLAY message as region status has been changed

Step5: END

**5.3.3.4.3.3 File I/O Interfaces**

Region table

**5.3.3.4.3.4 Output**

Successful message

**5.3.3.4.3.5 Implementation aspects (if any)**

Buttons

**5.3.3.5 View feedback**

**5.3.3.5.1 Input**

Button clicks

**5.3.3.5.2 Procedural Details**

View feedback

Retrieve feedback from database

Load feedback page

Display feedback

**5.3.3.5.3 File I/O Interfaces**

Feedback table

**5.3.3.5.4 Output**

Feedback will be displayed

**5.3.3.5.5 Implementation aspects (if any)**

Buttons

**5.3.4 Dealer**

**5.3.4.1 Pending orders**

**5.3.4.1.1 Input**

Button clicks

**5.3.4.1.2 Procedural Details**

Step1: start

Step2: CLICK on pending order

Step3: retrieve orders where dealer\_id in order table

Step4: DISPLAY orders with generate bill button

**5.3.4.1.3 File I/O Interfaces**

Order table

**5.3.4.1.4 Output**

Pending orders will be displayed

**5.3.4.1.5 Implementation aspects (if any)**

Buttons

**5.3.4.2 Previous orders**

**5.3.4.2.1 Input**

Button clicks

**5.3.4.2.2 Procedural Details**

Step1: Start

Step2: CLICK on previous order

Step3: retrieve dealer\_id where dealer\_id in order

table

Step4: retrieve order\_id where order\_id in bill table

Step5: DISPLAY orders with bill

Step6: END

**5.3.4.2.3 File I/O Interfaces**

Order table

**5.3.4.2.4 Output**

Previous order will be displayed

**5.3.4.2.5 Implementation aspects (if any)**

Buttons

**5.3.4.3 Bill Generation**

**5.3.4.3.1 Input**

Order details

**5.3.4.3.2 Procedural Details**

Bill generation

Input order details from database

Generate bill id

Display bill in pdf format

**5.3.4.3.3 File I/O Interfaces**

Cart table, Order table and Bill table

**5.3.4.3.4 Output**

Display bill

**5.3.4.3.5 Implementation aspects (if any)**

Buttons

**5.3.5 Customer**

**5.3.5.1 Cart**

**5.3.5.1.1 Modify cart**

**5.3.5.1.1.1 Input**

Items in the cart

**5.3.5.1.1.2 Procedural Details**

Step1: start

Step2: CLICK on add to cart button

Step3: retrieve product\_id from

product table

Step4: retrieve customer\_id from

customer table

Step5: store to cart table

Step6: CLICK on cart

Step7: retrieve details from cart table

Step8: SET quantity by increasing or\

decreasing

Step9: store quantity to cart table

**5.3.5.1.1.3 File I/O Interfaces**

Cart table

**5.3.5.1.1.4 Output**

Items will be added to cart with quantity

**5.3.5.1.1.5 Implementation aspects (if any)**

Buttons

**5.3.5.1.2 Place order**

**5.3.5.1.2.1 Input**

Details from cart

**5.3.5.1.2.2 Procedural Details**

Start

Click on Place order

Cart

Order

Order details

Retrieve details from cart

Generate order id

Store to database

Order placed successfully message

Stop

**5.3.5.1.2.3 File I/O Interfaces**

Order table, Cart table

**5.3.5.1.2.4 Output**

Order placed message

**5.3.5.1.2.5 Implementation aspects (if any)**

Buttons

**5.3.5.2 Previous order**

**5.3.5.2.1 View bill**

**5.3.5.2.1.1 Input**

Button clicks

**5.3.5.2.1.2 Procedural Details**

Start

Click on view bill

Retrieve bill from database

Bill

Show bill

Stop

**5.3.5.2.1.3 File I/O Interfaces**

Bill table

**5.3.5.2.1.4 Output**

Bill will be displayed

**5.3.5.2.1.5 Implementation aspects (if any)**

Buttons

**5.3.5.2.2 Feedback**

**5.3.5.2.2.1 Input**

Feedback details

**5.3.5.2.2.2 Procedural Details**

Feedback

Feedback details

Generate feedback id for feedback

Store to database

Display message

**5.3.5.2.2.3 File I/O Interfaces**

Feedback table

**5.3.5.2.2.4 Output**

Successful message

**5.3.5.2.2.5 Implementation aspects (if any)**

Textboxes, Buttons, Hyperlinks

**5.3.5.3 Pending orders**

**5.3.5.3.1 Input**

Click

**5.3.5.3.2 Procedural Details**

Step1: start

Step2: CLICK on active orders

Step3: retrieve dealer\_id not in order table

Step4: DISPLAY orders

**5.3.5.3.3 File I/O Interfaces**

Order table

**5.3.5.3.4 Output**

Active orders will be displayed

**5.3.5.3.5 Implementation aspects (if any)**

Buttons