**CHAPTER 5**

**DESIGN**

**5.1 SYSTEM DESIGN**

System design is a reduction of an entire system by studying the various operations performed and their relationships within the system and the requirements of its success. One aspect of design is defining the boundaries of the system and determining whether or not the candidate system should consider other related system. System can be defined, as an orderly grouping of interdependent components can be simple or complex.

The idea of the systems has been most practical and necessary in computerizing the interrelationships and integration of operations, especially when using computers. Thus it’s a way of thinking organizations and their problems. An organization consists of several interrelated and interlocking components.

The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the candidate system .It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the output and the inputs are also presented. In the second step, input data and master files are to be designed to meet requirement of the proposed output .The processing phase’s system’s objectives and complete documentation.

Finally details related to justification of the system and an estimate of the impact of the candidate system on the user and organization are documented and evaluated by management as a step towards implementation. The final report prior to the implementation phase includes procedure flow chart, record lay outs, and a workable plan for implementing the KDMS system.

System design has two phases:

* Logical
* Physical

The logical design reviews the present physical system, prepares the input and output and also prepares a logical design walk- through. We have to deal with how to take entries required and whether and how to process the user data. Also we have to deal with how to present the data in an informative and appealing format. This design also involves the methodology to store, modify and retrieve data from the data base as per the requirement.

Physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation plan and new hardware and software. We have to decide how and where to store the input data and how to process it so as to present it to the user in an easy, informative and attractive manner. A major step in the design is the preparation of input and output report in a form acceptable to the user .In this a data entry operator can feed the relevant details asked by the system for a particular task as input.

**Modules:**

1. Smart Checkout System using AI
2. Customer Classification using AI
3. Manager Module
4. Product Locator Module

**Module Description**

* **Smart Checkout System using AI**

Smart checkout system is a main part in this project where it will scan image using camera and identify product using google vision API. This module will detects and classifies multiple products including the location of each product within the camera position ie within images captured by the camera.Other than this it will load price and assign discount price applicable if any from server API and display it automatically on the screen.Load total bill on the screen after getting confirmation from user .The final bill will get displayed and the customer can now pay bill using payment gateway/card reader.After getting confirmation from payment gateway outpass will get generated and the customer can now leave the shop.And at last,it will automatically send a copy of invoice in checking area to cross check the bill (this feature is applicable in trial run period and will remove based on accuracy).

* **Customer Classification using AI**

This module can add new customer/register with auto face image capture and register and will save the data on to the database for future purpose.Here for registering customers face detection is done using Open CV with the help of a machine learning algorithm called CNN (Convolutional Neural Network).A stored database of the customers is manipulated using image processing techniques to accomplish this task.This module will identify the fraud customers and send alerts to security personnels.

* **Manager Module**

Manager module is the module which can access all other modules.It can add customers,edit customer details when needed and can also view the details of customers.Thet can access not only customer details but also employee details ie;they can add,edit and view the details of employees.Almost all power is vested with manager module which include managers can block/unblock customers and also employees.When looking into products section there also this module can add products,edit product details,remove the products from ERP to smart checkout module.Another speciality with this module is queue management -handles the customers queue section ordering also and also can view fraud detection alerts from customer classification module.

* **Product Module**

This module handles all about productsin that particular shop.This can add products,edit and view them.This is the module which locate the misplaced products/search for the products that are missing using camera images.Locate the misplaced products using object detection API Google vision,and shows the correct product location.

* **Convolutional Neural Network (CNN)**

CNN’s were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

**5.2 Data Flow Diagram**

A data flow diagram is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level 0 DFD also called fundamental system model represents the entire software elements as a single bible with input and output indicated by incoming and outgoing arrows respectively.

The DFD is one of the most important tools used by the system analyst. It can be used for the visualization of data processing. DFD provides a graphical representation of the flow of data through an information system. DFD illustrates how data is processed by a system in terms of inputs and outputs. It is a picture of system processing and flow without excessive concern for details.

The DFD showing the top level of the system is called “Context Diagram”. It should be overview including basic inputs, processes and outputs. Then it is exploded in to more detailed lower level diagram that shows additional features of the system.

The purpose of DFD is to provide a semantic bridge between users and system developers. The diagrams are graphical, eliminating thousands of words, logical representations, modeling what system does; hierarchical, showing system at any level of details; and Jargon less, allowing user interaction and reviewing.

Data Flow Diagrams Notations

**Process** : 

**Input/output** : 

**Flow of direction** : 

**Database/File** :

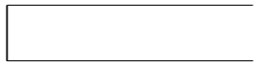


Figure 5.1 DFD Components

**5.2.1 Project DFD**

**Level 0 (Context Level):**

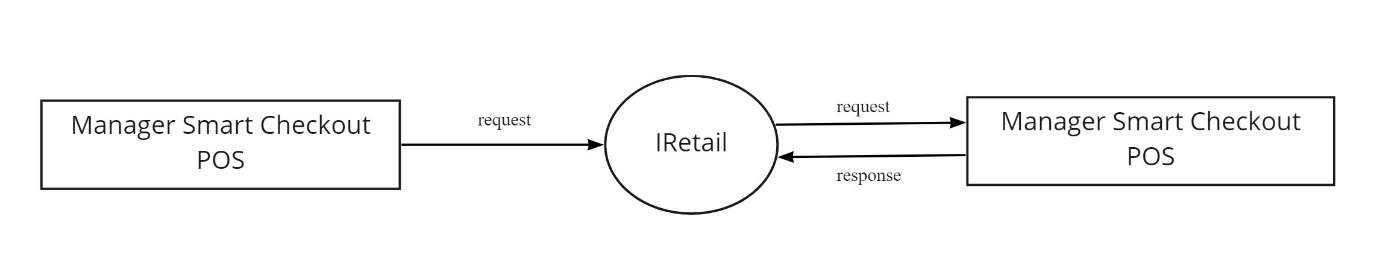


Figure 5.2 Context Diagram (Level 0)

**Level 1 :**

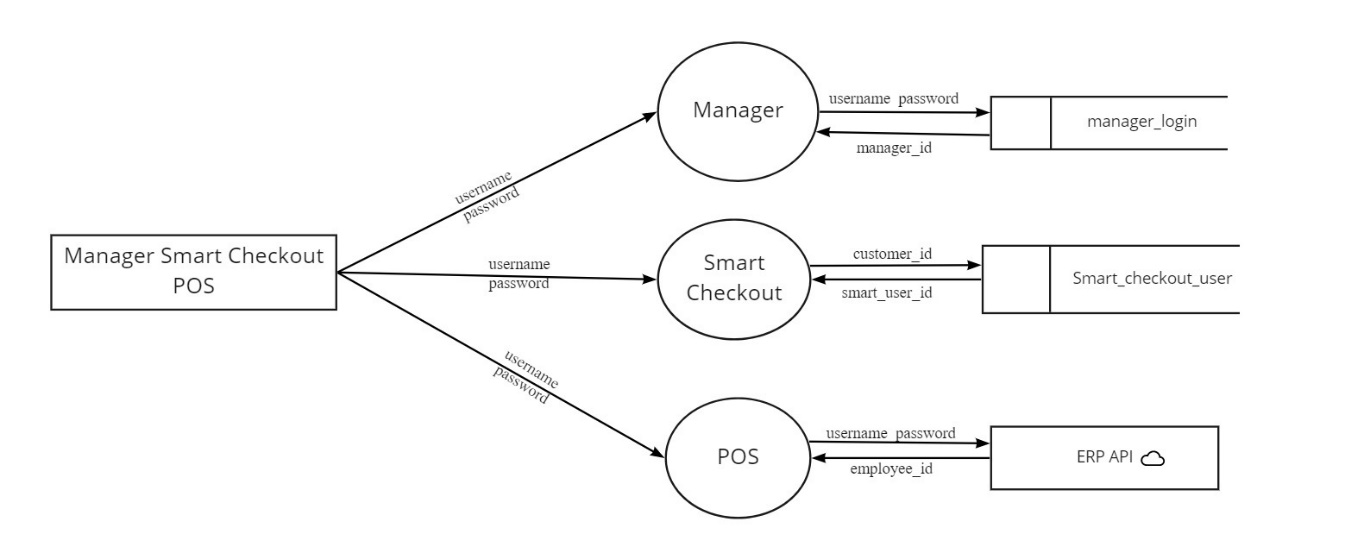
****

Figure 5.3 Module Description (Level 1)

**Level 1.0: Manager Module**

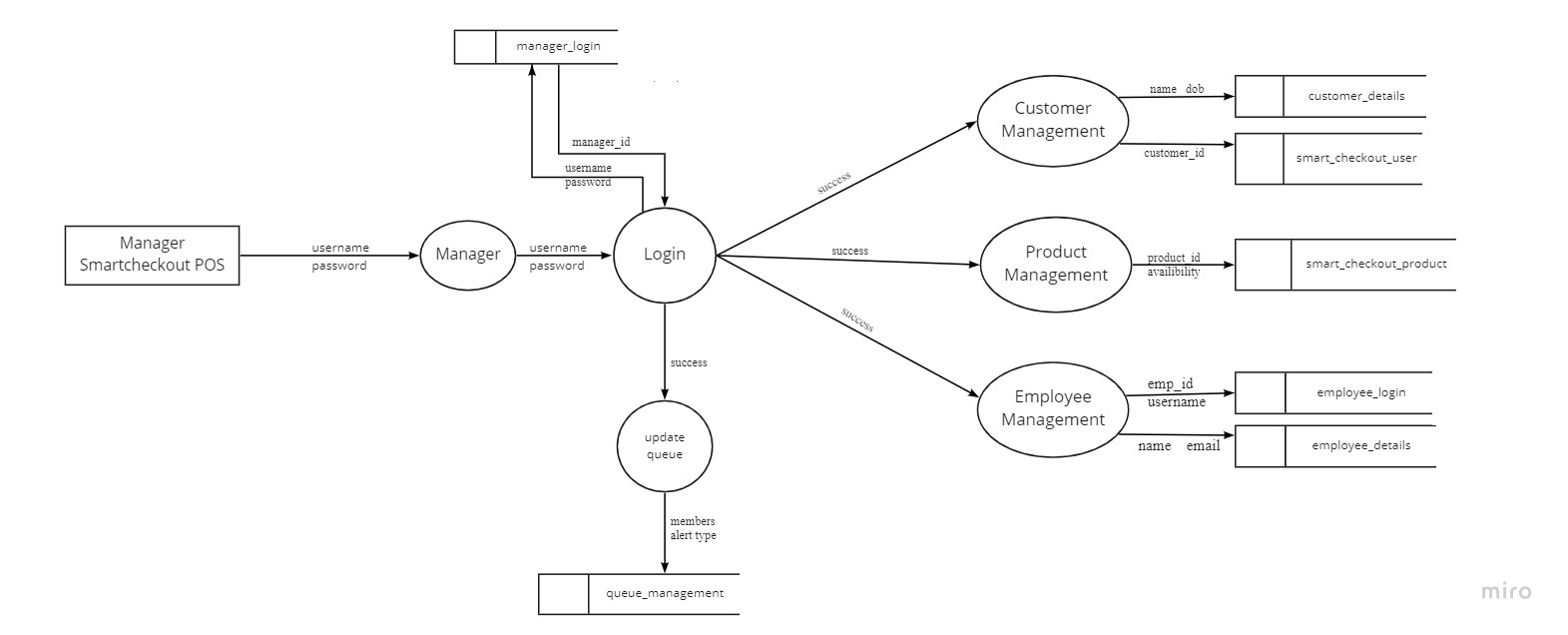


Figure 5.4 Manager Login (Level 1.0)

**Level 1.0.1: Customer Management**

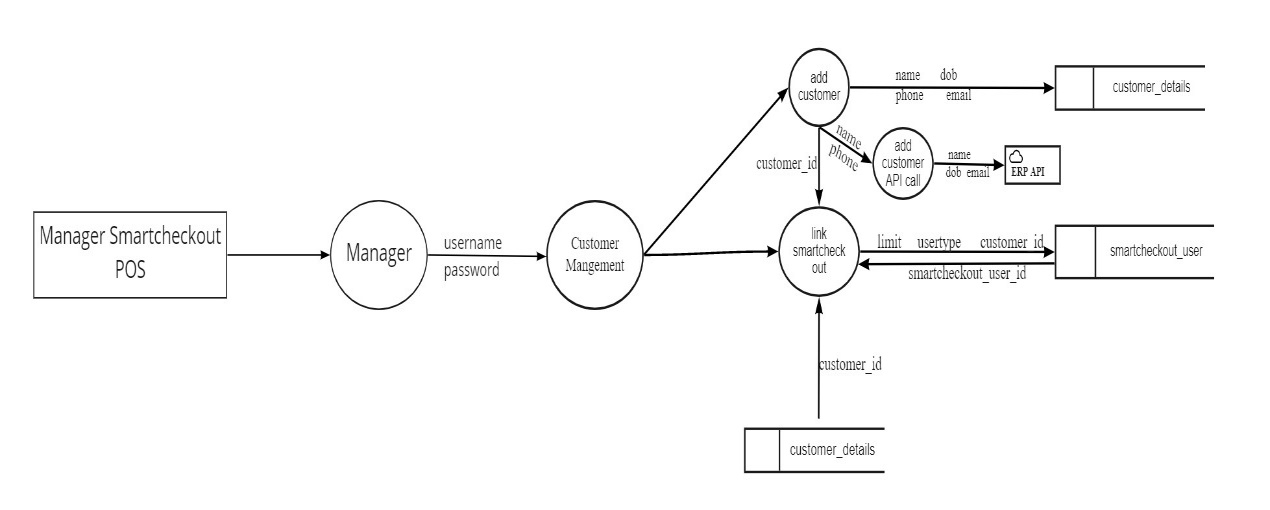


Figure 5.5 Customer Management (Level 1.0.1)

**Level 1.0.2 : Product Management**

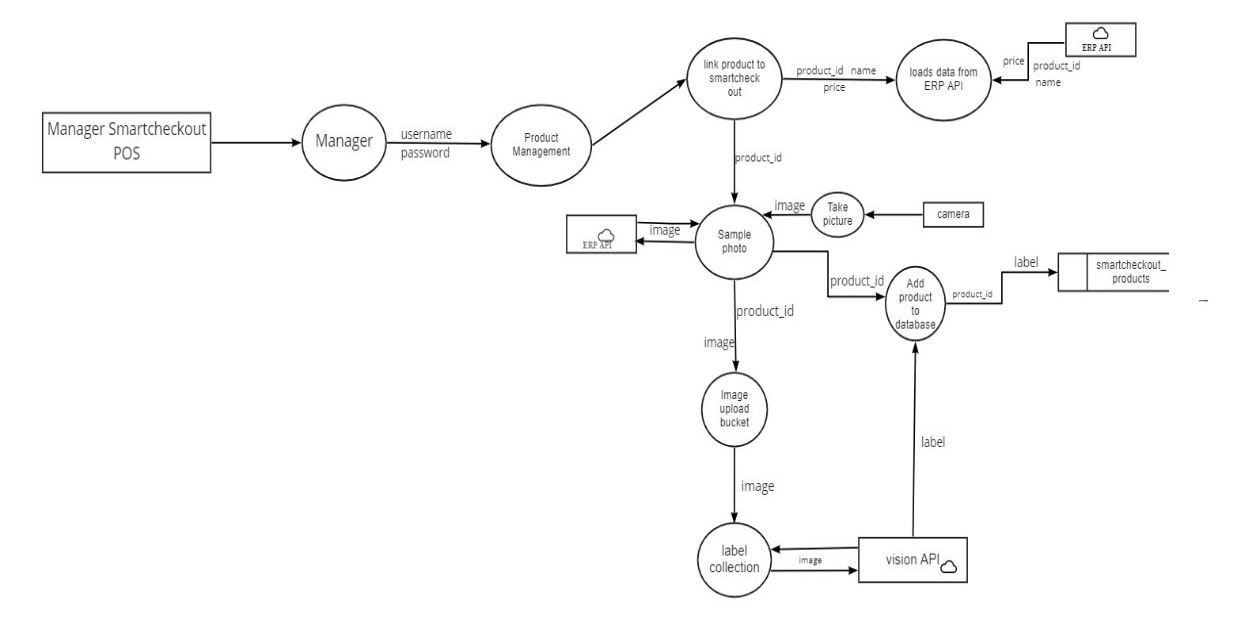


Figure 5.6 Product Management (Level 1.0.2)

**Level 1.0.3: Employee Management**

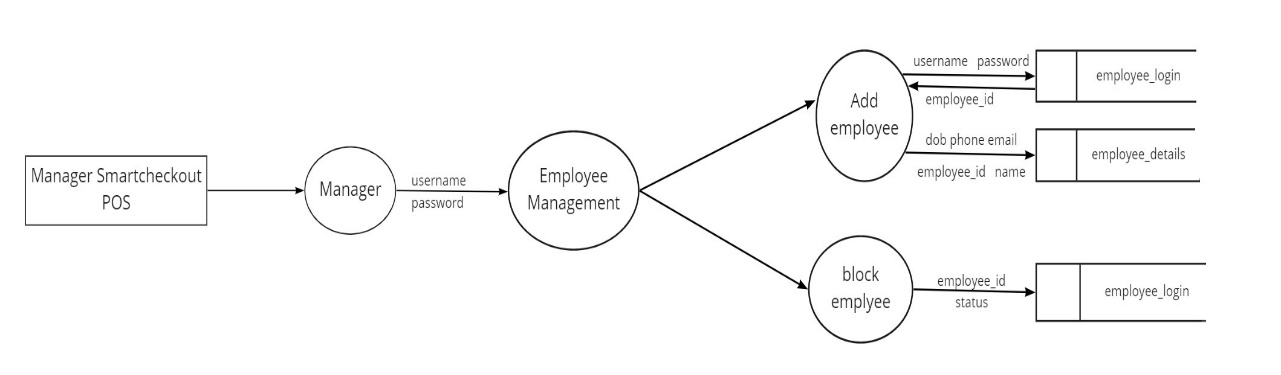


Figure 5.6 Employee Management (Level 1.0.3)

**Level 1.1: Smart Checkout**

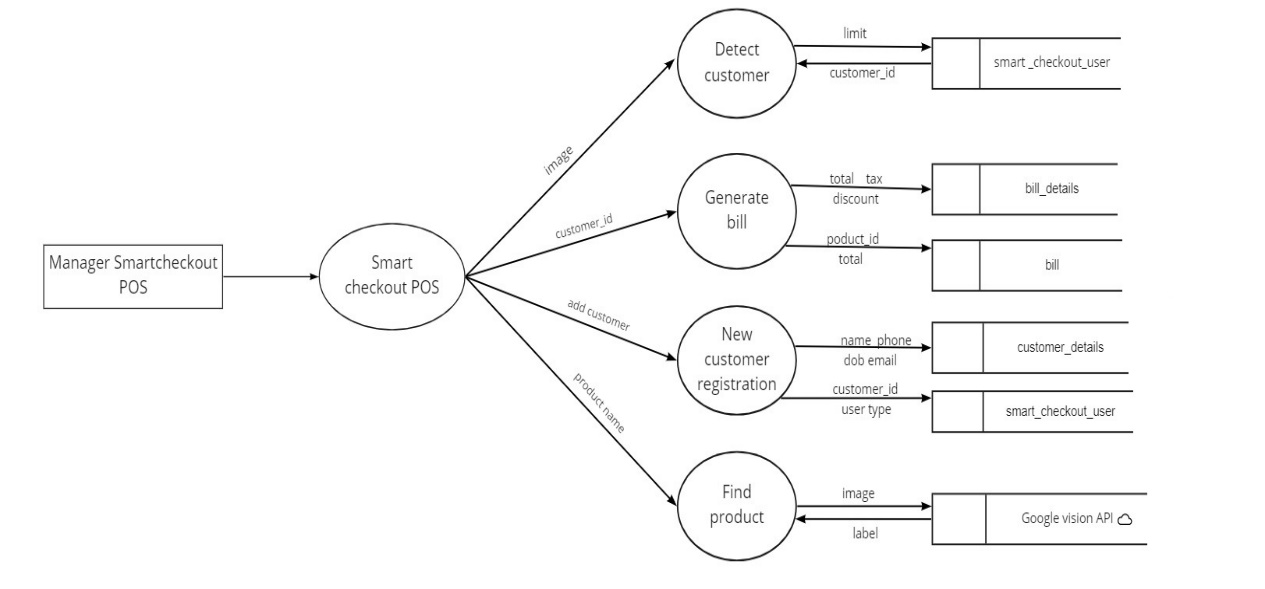
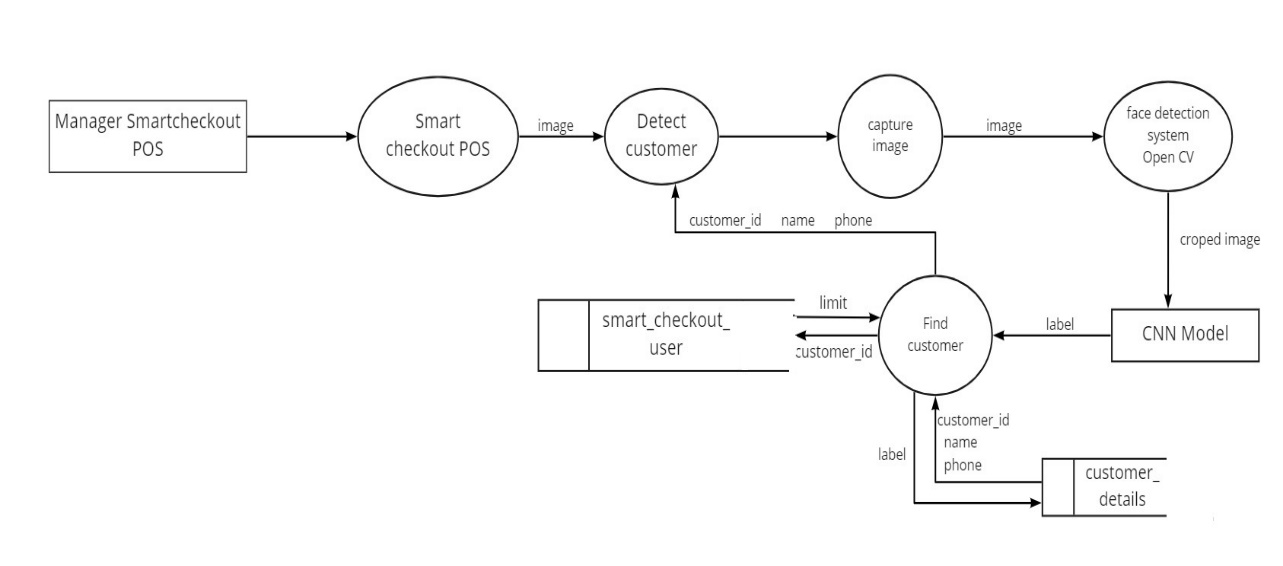
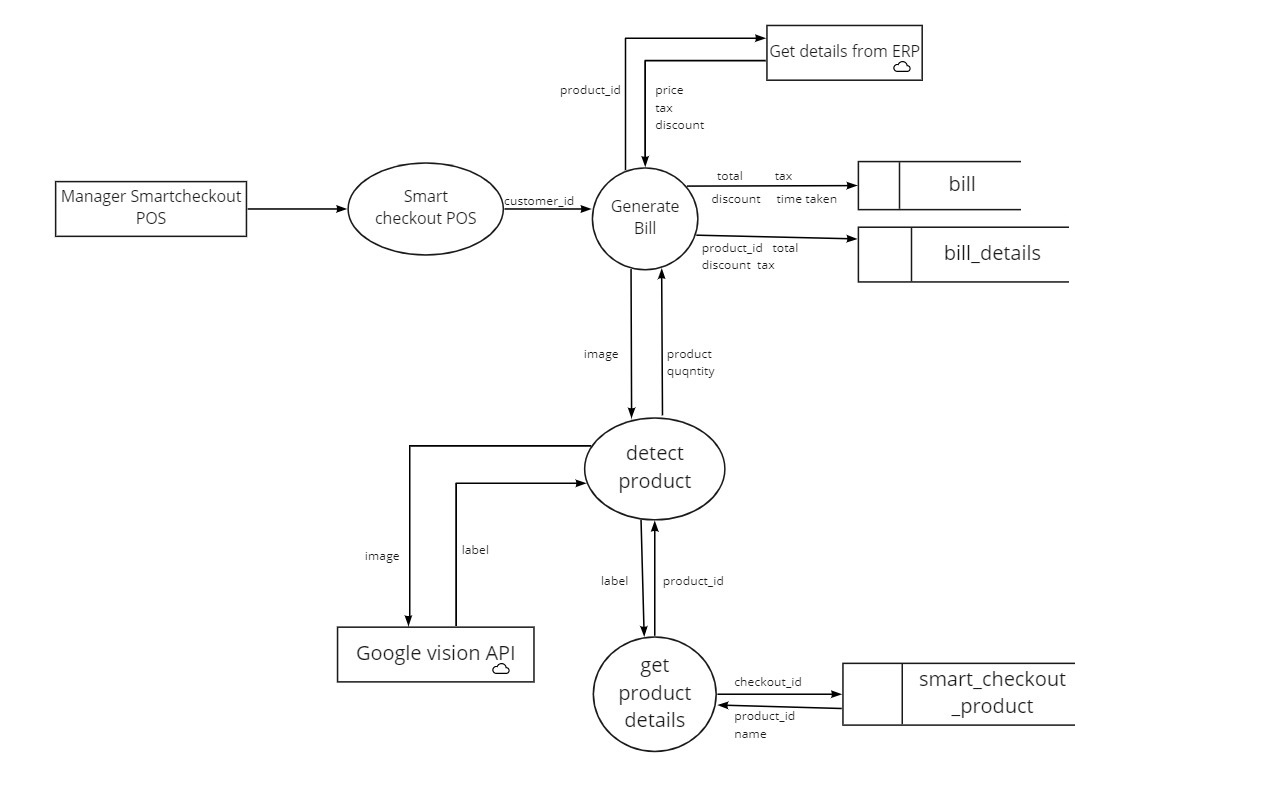
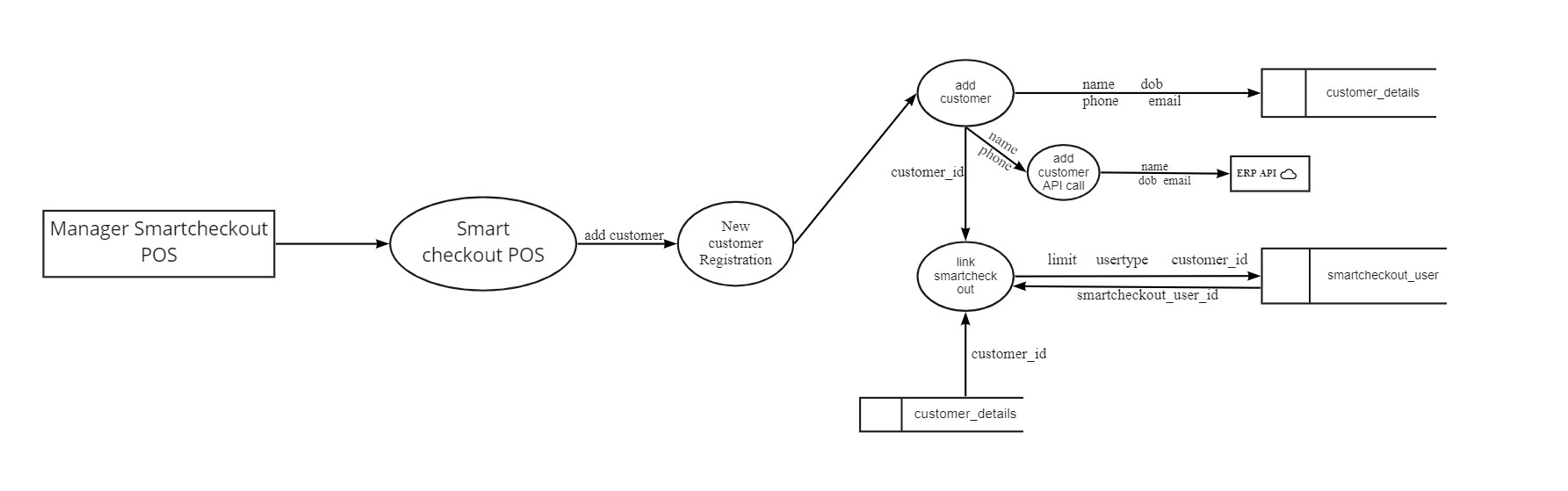


Figure 5.7 Smart Checkout (Level 1.1)



****

****

**Level 1.1.1: Find Product (Google Vision API)**

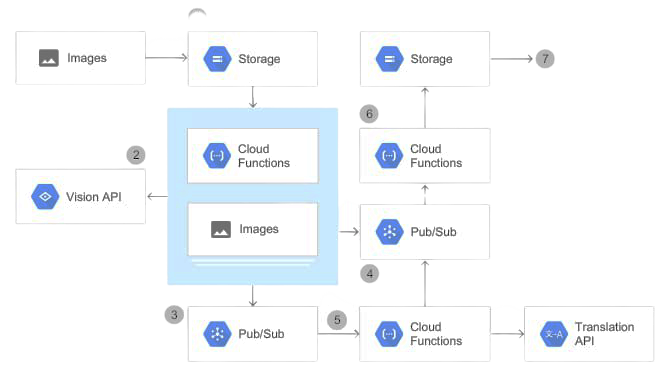
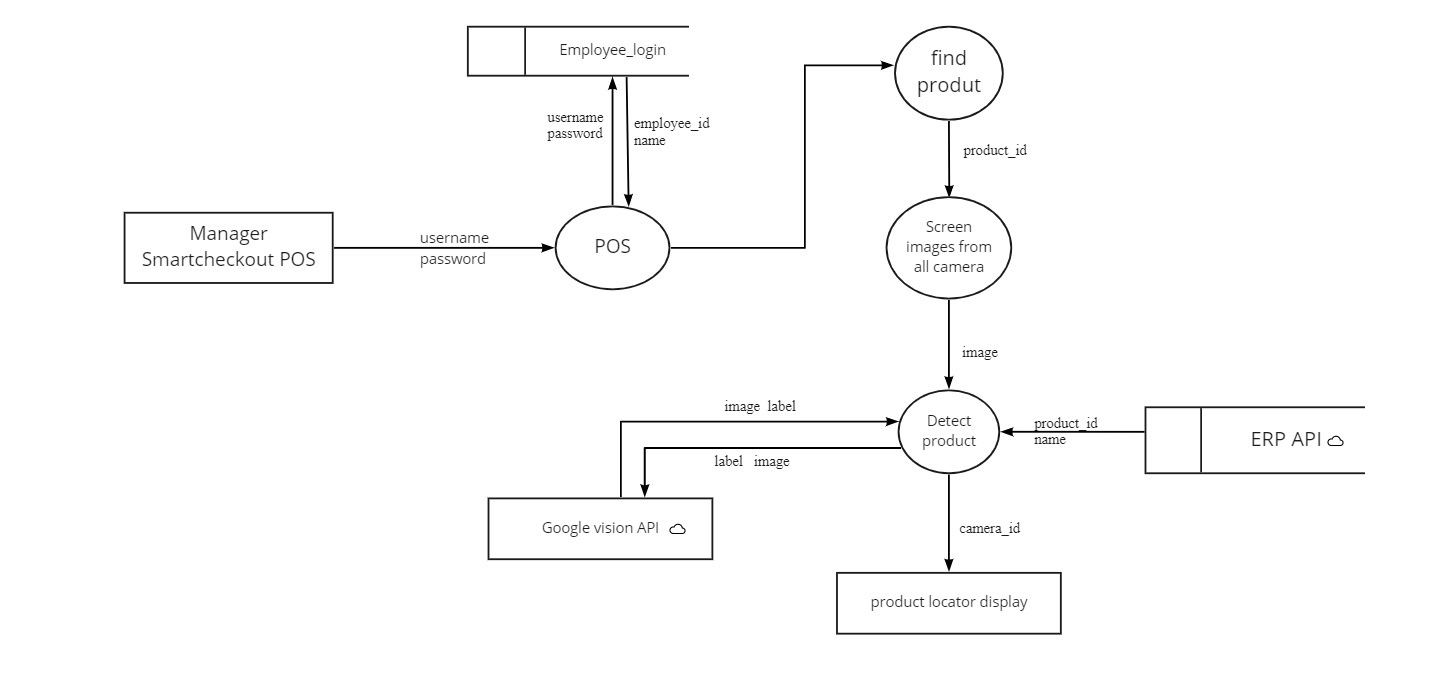


Figure 5.8 Find Product (Level 1.1.2)

**Level 1.2: POS**



**5.3 DATABASE DESIGN**

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. The main objectives of databases are data integrity and data independence. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users Database Management System (DBMS) allow the data to be protected and organized separately from other resources like hardware, software, and programs. DBMS is a software package, which contains components that are not found other data management packages. The significant of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. That is the difference between the logical and physical data.

The main objectives covered in database design are:

• Controlled redundancy

• Data independence

• Accuracy and integrity

• Privacy and security

• Performance.

**Manager Login**

Table Number: 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Manager\_login Primary Key: manager\_id** | | | |
| **Description:** Details of Manager login | | | |
| **Field** | **Data Type** | **Constraint** | **Description** |
| manager\_id | Integer | Primary Key | Id of manager |
| manager\_username | String |  | User name of manager |
| manager\_password | String |  | Password of manager |

**Queue Management**

Table Number: 5.2

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: queue\_management Primary Key: queue\_id** | | | |
| **Description:** Details of Manager\_login | | | |
| **Field** | **Data Type** | **Constraint** | **Description** |
| queue\_id | Integer | Primary Key | Id of queue |
| queue\_number | Integer |  | Number of queue |
| queue\_alerttype | String |  | Alert type of queue |

**Customer Details**

Table Number: 5.3

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Customer\_details Primary Key: customer\_id** | | | |
| **Description:** Details of Customer details | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| customer\_id | Integer | Primary Key | Id of customer |
| customer\_name | String |  | Name of customer |
| customer\_dob | String |  | Date of birth of customer |
| customer\_phone | String |  | Phone number of customer |
| customer\_email | String |  | Email id of customer |
| customer\_city | String |  | City of customer |
| customer\_status | Boolean |  | Status of customer |

**Smart Checkout Products**

Table Number: 5.4

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Smartcheckout\_products Primary Key: smart\_product\_id** | | | |
| **Description:** Details of Smart checkout products | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| smart\_product\_id | Integer | Primary Key | Id of smart checkout product |
| product\_id | Integer |  | Id of product |
| product\_displayname | String |  | Display name of product |
| product\_availability | Boolean |  | Availability of product |
| product\_identifier | String |  | Identifier of product |

**Smart Checkout User**

Table Number: 5.5

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Smartcheckout\_User Primary Key: smart\_user** | | | |
| **Description:** Details of Smart checkout user | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| smart\_user\_id | Integer | Primary Key | Id of smart checkout user |
| smart\_user\_limit | Double |  | Limit of smart checkout user |
| smart\_user\_usertype | String |  | User type of smart checkout user |
| smart\_user\_customer\_id | Integer | Foreign Key | Customer id of smart checkout user |
| smart\_user\_no\_of\_bills | Integer |  | Number of bills of smart checkout user |

**Employee Login**

Table Number: 5.6

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Employee\_login Primary Key: emp\_id** | | | |
| **Description:** Details of Employee login | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| emp\_id | Integer | Primary Key | Id of employee |
| emp\_username | String |  | Name of employee |
| emp\_password | String |  | Password of employee |
| emp\_status | Boolean |  | Status of employee |

**Employee Details**

Table Number: 5.7

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: Employee\_details Primary Key: emp\_details\_id** | | | |
| **Description:** Details of Employee | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| emp\_details\_id | Integer | Primary Key | Id of employee details |
| emp\_id | Integer | Foreign Key | Id of employee |
| emp \_name | String |  | Name of employee |
| emp \_address | String |  | Address of employee |
| emp\_email | String |  | Email id of employee |
| emp\_phone | String |  | Phone number of employee |
| emp\_dob | String |  | Date of birth of employee |

**Bill**

Table Number: 5.8

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: bill Primary Key: bill\_id** | | | |
| **Description:** Details of Bill | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| bill\_id | Integer | Primary Key | Id of bill |
| smart\_user\_id | Integer | Foreign Key | User id of smart checkout |
| bill\_total | Double |  | Total for the purchase |
| bill\_tax | Double |  | Tax for the purchase |
| bill\_discount | Double |  | Discount for the purchase |
| time\_taken | Double |  | Time taken for the purchase |

**Bill Details**

Table Number: 5.9

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name: bill\_details Primary Key: bill\_details\_id** | | | |
| **Description:** Details of Bill | | | |
| **Field** | **Data Type** | C**onstraint** | **Description** |
| bill\_details\_id | Integer | Primary Key | Id of bill details |
| bill\_id | Integer | Foreign Key | Id of bill |
| smart\_product\_id | Double | Foreign Key | Id of smart checkout product |
| bill\_total |  |  | Total for the purchase |
| bill\_discount | Double |  | Discount for the purchase |
| bill\_tax | Double |  | Tax for the purchase |
| Bill\_amount | Double |  | Bill amount in the bill |

**5.4 Input Design**

Input design is the process of converting user inputs into computer-based format. The goal of designing input data is to make the data entry, logical and error free. Input data are collected and organized into groups of similar data. The project requires a set of information from the user for processing. The various factors to be considered during input designs are input data to the system, methods for performing input validation and steps to follow when error occurs, medium to use, arrangement of data etc. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for the right message and help for the user at the right time are also considered for the development of this project.

In this project all the fields are validated. If any field then error message will be displayed, so as to help the user while giving inputs. The drop-down lists are used to reduce the user inputs and to select a preferred item from the list easily. Check boxes are used for user’s category selection. User just needs to click the preferred category from the checkbox list. User will select one of the items in list boxes. The following design guidelines will result in a friendly and deficient interface. Minimize the number of input actions required from user. This can be accomplished by using the mouse to select from the predefined set of inputs. In application user can select the options.

In the project “IRetail -Smart Store with enhanced security powered by AI” the input design is done in such a way that the users of the system will never get confused or enter wrong data. The simplicity and ease of use lies in the act that the desired objectives can be accomplished with a few mouse clicks.The main objectives that are guiding as in the input stages are:

* Controlling the amount of inputs
* Avoiding inordinate delay
* Controlling errors 50
* Avoiding extra steps
* Keeping the process simple
* To achieve highest level accuracy.

**Output Design**

Output refers to the results and information that are generated by the system. Here determine information to be present, decide layout and select output medium, arrange presentation of information in accepted format of column headings and pagination are specified. Output design plays a major role in providing the user with the required format. The major function of the output is to convey information and so its layout and design are careful considerations. Information must be carefully considered to the needs of the user. Standards for printed output suggest giving each output a name or title, providing a sample of the output layout, and specifying the procedure for providing the accuracy of the output data.

The output devices to consider depends on the compatibility of the devices with the system, response time requirement and printed quality required. 27 The design output form, attention is given to proper identification and wording, readability and use, composition and layout, order of data items and clarity of instructions. A well-designed form with clarity stated captions should be self-instructing. An organizations form must be centrally controlled for efficient handling. Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the systems relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user/ so while designing output the following things are to be considered.

* Determine what information to present.
* Arrange the presentation of information in an acceptable format.
* Decide how to distribute the output to intend receipts.
* Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy.
* Efficient and intelligent output design should improve system relation with the user and help in decision making that is, this makes system user friendly to be displayed or printed as per the user’s choice.

##### 5.4 Program Design

###### i. Manager

Step 1: Start

Step 2: Once logged in the manager has the privilege to manage customer, queue management, employee management, product management.

Step 3: In manager function mangers can login,update queue,monitor customers,can identify fraud customers ,details about employees like login,edit their details,delete,view.

Step 4: In customer function manager can add customers,link smart checkout system,add customer API call,capture image of customers and store their details in a dataset.

Step 5: In product function manager can link products to smart checkout,detect products,loads data from ERP-API ,add products to API.

Step 6: In employee function manager can add,update,delete,block and view the employee details.

Step 7: Stop

###### ii. Smart Checkout

Step 1: Start

Step 2: Smart checkout can capture images,face detection of the customers,bill generation,find product details,detect the products at the time of billing .

Step 3: Stop

###### iii. POS

Step 1: Start

Step 2: In this function POS will find the products that are misplaced

Step 3: Stop

**User**

Step 1: Start

Step 2: Once logged in the user can add their details and this details will be stored in the database of the shop for further classifications.

Step 3: User get their allotted queue

Step 4: User can the collect the products needed,and can generate bill by showing products to the camera.

Step 5: If any products are not clearly captured by the camera will be notified by the software.

Step 6: If step 4 repeats again and again the software will classify that customer as fraud customer.And the user can no longer use this facility.

Step 7: After users can checkout after paying the bill

Step 8: Stop

### CHAPTER 6

**FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

#### 6.1 FUNCTIONAL REQUIREMENTS

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used. Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform. Functional Requirements should include the following things:

* Details of operations conducted in every screen
* Data handling logic should be entered into the system
* It should have descriptions of system reports or other outputs
* Complete information about the workflows performed by the system
* It should clearly define who will be allowed to create/modify/delete the data in the system

#### 6.2 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements are “system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”, “quality goals", "quality of service requirements" and "non-behavioural requirements. Some of the nonfunctional requirements are mentioned below i. Usability: The system shall have a clean interface with only needed features, clear terminology and tool tips wherever necessary. Warnings or alerts shall be specified in clear way. ii. Efficiency: The system shall respond to different searches being conducted like searching particular product, search quantity, etc. in a very fast way. iii. Portability: The system shall be independent of the specific technological platform used to implement it. iv. Reliability: Reliability defined as a measure of the time between failures occurring in a system (measure show frequently the system fails), so that the system shall operate without any failure for a particular period of time v. Availability: Availability measures the percentage of time the system is in its operational state so that the system shall be available for use 24 hours per day and 365days per year.

CHAPTER 7

TESTING

**7.1.Testing Objectives**

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences.

Testing is the major quality measure employed during the software development. After the coding phase, computer programs available are executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

Testing is a process of executing a program with the intension of finding an error. A good test case is one that has a highest probability of finding an as yet undiscovered error. A Successful test is one that uncovers an as yet undiscovered error. Our objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort.

Testing demonstrate that software functions appear to be working according to specification, that performance requirements appear to have been met. Data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But there is one thing that testing cannot do:

Testing cannot show the absence of defects it can only shoe that software defects as present.

**7.2. Testing Methods**

**7.2.1. Unit Testing**

In this testing, each module is tested individually and integrated the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as ‘module’ testing. The modules of the system are testing separately.

This testing is carried out during programming stages itself. In this testing step each module is found to be validation checks for verifying the data input given the user which both the formal and validity of the entered. It is very easy to find error and debug the system.

In the project “**IRetail-Smart Store enhanced with Security powered by AI**” each module is tested individually and is found to be an error free one

**7.2.2 Integration Testing**

Data can be lost across an interface; one module can have an adverse effort on the other sub functions, when combined may not produces the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The developed system has run successful for this sample data.The need for integrated test is to find the overall system performance.

In the project “**IRetail-Smart Store enhanced with Security powered by AI**” each module is tested individually and all the modules are integrated together and the integration testing is carried out for the whole system. The whole system is working accurately without any errors.

**7.2.3 User Acceptance Testing**

Acceptance testing is a key factor to the success of any system. The system, under the consideration was treated for user acceptance by constantly keeping in touch with the prospective system user at the time of developing and making changes where ever & whenever required.

In the project “**IRetail-Smart Store enhanced with Security powered by AI**” User acceptance testing is carried out and proper changes are made whenever required.

**7.2.4 Data Validation Testing**

Data validation is the process of testing the accuracy of data; a set of rules you can apply to a control to specify the type and range of data that users can enter. It can be used to display error alerts when users enter incorrect values in the form. Rather than checking for errors after a form is completed, data validation verifies values as the form is being filled out.

A strategy for software testing integrates software test case design method in to a well- planned series of steps that result in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when these steps are planned and then undertaken, and how much effort, time and resources will be required.

Therefore any testing strategy must incorporate test planning, test case, design, test execution and resultant data collection and evaluation. A software testing strategy should be flexible enough to promote a customized testing approach. At the same time, it must be rigid enough to promote reasonable planning and management tracking as the project progresses. The project manager, software engineers and testing specialists develop a strategy for software testing.

In the project“**IRetail-Smart Store enhanced with Security powered by AI**” data validation testing is carried out and proper error messages are displayed when users enter incorrect values into a form.

**7.2.5 Whitebox Testing**

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc. In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

**7.2.6 Blackbox Testing**

Black box testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name “black box” suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. Black box testing is sometimes also called as “Opaque Testing”, “Functional/ Behavioral Testing” and “Closed Box Testing”.

The base of the black box testing strategy lies in the selection of appropriate data as per functionality and testing it again the functional specifications in order to check for normal and abnormal behavior of the system. Now a day, it is becoming common to route the testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system, which makes it unfit to test the application by the developer.

In order to implement Black Box Testing strategy, the tester is needed to be through with the requirement specifications of the system and as a user, should know, how the system should behave in response to the particular action. In our project white box as well as black box testing is done.