**CHAPTER 1**

# INTRODUCTION

## 1.1 GENERAL INTRODUCTION

In the retail industry intelligent operations like top customer recognition, minimising losses due to shoplifting, emergencies and fraud is key to success.This application integrates to the cctv camera and the deep learning algorithms used to process this images ,automated intrusion detection alerts, real-time fire alerts,smart checkout, blacklisted customer alerts, and potentially fraudulent point of sale (POS) transactions are main features.Dynamic Pricing with machine learning:the price of certain products change over time through an algorithm that considers a variety of pricing variables. These metrics could include the season of the year, as well as supply and demand. With this technology, retailers have more flexibility when generating the right price at the right time without losing sight of their main goals, including profit or revenue optimization. Product locator using image search : search and locate a particular product in the store using image search .the system will collect all available product locations from the live videos from multiple cameras.This is a useful feature for finding misplaced products POS user fraud detection : record POS transaction details on screen as an overlay on video images to reduce fraudulent transactions and employee fraud .To improve the ease and speed of transaction searches face of customer is used to search for customer dispute cases. Smart Checkout System:using self service kiosk users can generate bills by showing products to linked camera, using Google vision products will get identified . this will reduce the queue. blacklisted customers can't use this system. Count customers and Queue detection: Detect long queues at the checkout in real time and open new lanes to speed up and improve the customer experience, and reduce abandoned purchases. customer recognition:Premium customers who opt to participate in special marketing promotions or other incentives can be identified so staff can provide the right kind of service at the right time. blacklisted customer alerts can also provided at the same time to avoid loss.

## 1.2 GOAL OF THE PROJECT

The main goal of the project is improve the security and ease for the customers with the help of the operations like top customer recognition, minimizing losses due to hoplifting,checkout automation,trust,queue detection,reduce the use of manpower,increases efficiency . This application integrates to the cctv camera and the deep learning algorithms used to process this images ,automated intrusion detection alerts, real-time fire alerts,smart checkout, blacklisted customer alerts, and potentially fraudulent point of sale (POS) transactions are main features.Dynamic Pricing with machine learning, the price of certain products change over time through an algorithm that considers a variety of pricing variables.All these will automate the working inside a retail shop,Self checkout system ,detection of misplaced objects,dynamic pricing according to the market rate analysis etc are the min highlights thus by increasing the efficiency,decrease the manpower,more convinent to both customers and shop owners.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 STUDY OF SIMILAR WORK**

### 2.1.1 EXISTING SYSTEM

Currently,in the retail shops there is only POS systems,there is not an option for a customer to take part in the billing process and all.In the existing system they can order the list of products,customers can registration,products pricing list,billing,sending sms for the customers etc can be done.This system is automated,even though it needs more manpower than newly developing system. Avoid the need for paperwork and effectively manage your cash flow to ring up sales and keep track on the amount of ready cash available in your stores,automate sales, manage billing but this all needs manpower.Existing system is not upto the technology.

### 2.1.2 DRAWBACKS OF EXISTING SYSTEM

* Struggling without a contact less shopping solutions
* Queue detection
* Data privacy
* Security issues
* Over crowding and queuing in shops
* No time slot reservation
* Product detection-Misplaced products cannot be identified
* Genuine and fraud customers cannot be categorized automatically
* Needs manpower
* Self checkout system
* Lack of personal interaction

**CHAPTER 3**

OVERALL DESCRIPTION

## 3.1 PROPOSED SYSTEM

In the retail industry intelligent operations like top customer recognition, minimising losses due to shoplifting, emergencies and fraud is key to success.This application integrates to the cctv camera and the deep learning algorithms used to process this images , automated intrusion detection alerts, real-time fire alerts,smart checkout, blacklisted customer alerts, and potentially fraudulent point of sale (POS) transactions are main features.queue detection technology is built with purpose – to offer cutting-edge flow analysis to retail outlets, including supermarkets, exhibition halls, chain stores and any situation where waiting to pay is required. by deep learning algorithms, this easy-to-use technology is highly accurate at counting people, and can even detect their specific dwell time.Once finding a queue that is too long or a customer who has been waiting too long, the operating system can promptly notify the store manager to open a new checkout window, making the process quick and easy for the customer.Queue Detection Technology can enable managers to improve the operational efficiency of a store. Armed with new insights into the typical flow of customers during specified business hours, managers can scale staffing levels up and down accordingly.

Smart Checkout System is another important feature of this application.An alternative to checkout lanes staffed by cashiers, they are introduced to offer customers more control, convenience, and a speedier checkout option.Self service kiosk users can generate bills by showing products to linked camera, using Google vision products to get identified.This will reduce the queue. A single attendant can work the register for people who need assisted service while being available to help self-checkout customers.Increased convenience at checkout gets customers through the store faster, meaning the store can process more transactions with less staff.Self-checkout saves on labor cost, allowing more associates to support and serve shoppers by eliminating the need for onelane cashiers.can switch between cashier-assisted and self-service modes for optimal performance.

The price of certain products change over time through an algorithm that considers a variety of pricing variables. Dynamic Pricing enables businesses to use different pricing methods to calculate average rate / gross margin as well as implement limits on pricing. With end-to-end automation, managers can update pricing up to three times a day without the need for assistance from IT.Pricing solution allows businesses to automate pricing strategies and set flexible pricing rules to account for the pricing elasticity of every product in inventory. Managers can compare product market value with competitors, examine stock demands, and set up pricing rules at the category or commodity level. Fraud detection solution that works on a stream of POS data. The solution identifies normal behavior patterns, detects possible fraud and abuse. record POS transaction details on screen as an overlay on video images to reduce fraudulent transactions and employee fraud .To improve the ease and speed of transaction searches face of customer is used to search for customer dispute cases.

## 3.2 FEATURES OF PROPOSED SYSTEM

* Dynamic Price prediction
* Face recognition
* Customer identification
* Queue detection and management
* Fraud detection
* Identify products
* Smart checkout system
* Count customers
* Image based product search
* Billing
* Categorize Customers
* High speed processing of video
* Cloud integration

## 3.3 FUNCTIONS OF PROPOSED SYSTEM

1. **Customer recognition:**The camera detects and locates the image of a face, either alone or in a crowd.Identify the identity of existing customers or register if that customer is new. Premium customers who opt to participate in special marketing promotions or other incentives can be identified so staff can provide the right kind of service at the right time. Blacklisted customer alerts can also be provided at the same time to avoid loss.
2. **Dynamic Pricing with machine learning:**The price of certain products change over time through an algorithm that considers a variety of pricing variables. These metrics could include the season of the year, as well as supply and demand. With this technology, retailers have more flexibility when generating the right price at the right time without losing sight of their main goals, including profit or revenue optimization.
3. **Product locator using image search:**Search and locate a particular product in the store using image search .the system will collect all available product locations from the live videos from multiple cameras.This is a useful feature for finding misplaced products.
4. **POS user fraud detection :** Record POS transaction details on screen as an overlay on video images to reduce fraudulent transactions and employee fraud .To improve the ease and speed of transaction searches face of customer is used to search for customer dispute cases.
5. **Smart Checkout System:**using self service kiosk users can generate bills by showing products to linked camera, using Google vision products will get identified . this will reduce the queue. blacklisted customers can't use this system.
6. **Queue Detection Technology** uses cameras to count the number of people in a line, to a predetermined threshold. Once the threshold of customers has been reached, and if more customers are continuing to wait, the system alerts store management to open new checkout counters or cash registers.

## 3.4 REQUIREMENTS SPECIFICATION

System analyst tasks to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. These can be done through questionnaires, details investigation, observation, collection of samples etc. As the details are collected, the analyst study the requirements data to identify the features the new system should have, including both the information the system produce and operational features such as processing controls, response times, and input output methods.

Requirement specification simply means, “Figuring out what to make before you make it”. It determines what people need before you start developing a product for them. Requirement definition is the activity of translating the information gathered in to a document that defines a set of requirements. These should accurately reflect what consumer wants. It is an abstract description of the services that the system should provide and the constraints under the system must operate. This document must be written for that the end user and the stake holder can understand it.The notations used for requirements definition should be based on natural languages, forms and simple intuitive diagrams. The requirements fall into two categories: functional requirements and nonfunctional requirements.The requirements of specification of the proposed system are as follows:

* Minimum time needed for various processing
* Better service
* Faster response time
* User friendly

## 3.5 FEASIBILITY ANALYSIS / STUDY

The main aim of the feasibility study activity is to determine. Whether it would be financially and technically feasible to develop the product. The feasibility study activity involves analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system the processing required to be carried out of these data, the output data required to be carried out of these data, the output data required to be produced by the system, as well as various constraints on the behaviour of the system.

Feasibility studies aim to objectively and rationally uncover the strengths and weakness of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained As such, a welldesigned feasibility study should provide a historical background of the business or project, description of the product or vice, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, studies precede technical development and project implementation.

The feasibility study to be conducted for this project involves:

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility
* Behavioral Feasibility

### 3.5.1 TECHNICAL FEASIBILITY

The technical Feasibility depends on the technical aspects of the proposed system. The main consideration is to be given at the study of available resources of the organizations where the project is to be developed and implemented.

This " IRetail-Smart store with enhanced securiypowered by AI " is developed by using Python andMongoDB . So the technical part of this project is very secure. Maintainability and productivity is also high. So “ IRetail-Smart store with enhanced securiypowered by AI” is technically feasible.

### 3.5.2 OPERATIONAL FEASIBILITY

Operational analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and saving that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. Cost-based study: It is important to identify cost and benefit factors, which can be categorized as follows:

1. Development costs.
2. Operating costs.

This is an analysis of the costs to be incurred in the system and benefits derivable out of the system.

" IRetail-Smart store with enhanced securiypowered by AI " there is no difficulty in implementing the system. The proposed system is effective, user friendly and functionally. The user of the system must be completely aware of the internal working of the system so that the users will not face any problems running the system.

### 3.5.3 ECONOMICAL FEASIBILITY

Economic is used for evaluating the effectiveness of the system. This project is economically feasible because the project can be completed in few months. The cost and the benefits associated with the candidate system was considered. This feasibility checks whether the system can be developed with the available funds. The " IRetail-Smart store with enhanced securiypowered by AI " does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of requirements that are used required.

### 3.5.4 BEHAVIOURAL FEASIBILITY

The behavioral feasibility depends upon whether the system performed in the expected way or not.Feasibility study is a test of system proposal according to its workability, impact on organization ability to meet the user’s need and efficient use of resources. However, a feasibility study provides a useful starting point for full analysis. Our project checks whether the system is performed in the expected way or not. For this we have given inputs for checking whether the expected outputs where generated. Feasibility study is a test of system proposal according to its workability, impact on organization ability to meet the user’s need and efficient use of resources. However, a feasibility study provides a useful starting point for full analysis.

**CHAPTER 4**

# OPERATING ENVIRONMENT

## 4.1 HARDWARE REQUIREMENT

|  |  |
| --- | --- |
| Processor | : Intel i7 8th Gen |
| RAM | :8 GB DDR4 |
| Hard Disk | : 512GB SSD |
| Display Size | : 15’’LED Monitor |
| Screen Resolution | : 1920\*1080 Pixels |
| Keyboard | : Wireless Enabled Keyboard(Recommend :Logitech) |
| Keyboard Mouse | : Wireless Enabled Mouse (Recommend :Logitech) |

MONITOR : LED Monitor

Dedicated Graphics Card : Nvidia Geforce GTX 1050 4GB DDR5

## 4.2 SOFTWARE REQUIREMENT

Operating System : Windows (7/8/10)/Ubuntu (14/16/18/20)

Programming Language : Python

IDE : Open CV

Front-End : Python Django

Back-End : MongoDB

## 4.3 TOOLS AND PLATFORMS

### 4.3.1 Python

Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python is simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python is used in many applications:

* Web and Internet Development
* Scientific and Numeric
* Education
* Desktop GUIs
* Business Applications

Python is mostly preferred for image processing than MATLAB. The scientific Python ecosystem is maturing fast and Python is an appealing alternative, because it’s free, open source, and becoming ever more powerful. The common differences between the two tools are:

* Easy programming language: Python was created to be a generic language that is easy to read, while MATLAB started as a matrix manipulation package to which a programming language was added.
* Powerful: It’s easier than other languages to transform the ideas into code. Further, Python comes with extensive standard libraries, and has a powerful data types such as lists, sets and dictionaries. These really help to organize the data.
* Portability: Because Python is for free, the code can run everywhere. Further, it works on Windows, Linux, and OSX.
* Class and function definitions: Functions and classes can be defined anywhere. In one file as many functions and classes can be designed.
* Great GUI toolkits: With Python a front-end for the application that looks good and works well can be created. Any of the major GUI toolkits like Wxor Qt. Pyzo comes with PySide (a wrapper for Qt) can be chosen.

**What can Python do?**

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development **Why Python?**
* Python works on different platforms (Windows, Mac, Linux, Raspberry

Pi, etc).

* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.
* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

**Features of Python:**

#### 1. Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high-level programming language.

#### 2. Expressive Language

Python language is more expressive means that it is more understandable and readable

#### 3. Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

#### 4. Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

#### 5. Free and Open Source

Python language is freely available at official web address. The source-code is also available.Therefore, it is open source.

#### 6. Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence.

#### 7. Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

#### 8. Large Standard Library

Python has a large and broad library and prvides rich set of module and functions for rapid application development.

**9. GUI Programming Support**

Graphical user interfaces can be developed using Python.

#### 10. Integrated

* It can be easily integrated with languages like C, C++, JAVA etc.

* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

* Python was designed for readability, and has some similarities to the English language with influence from mathematics.

* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

#### 4.3.2 MongoDB

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage retrieval of data in the MongoDB are not in the form table.

The MongoDB database is developed and managed by MongoDB.Inc under SSPL (ServerSide Public License) and initially released in February 2009. It also provides official driver support for all the popular languages like C, C++,

C#, and .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid. So, that you can create an application using any of these languages. Nowadays there are so many companies that used MongoDB like Facebook, Nokia, eBay, Adobe, Google, etc. to store their large amount of data.

##### Features of MongoDB

* **Schema-less Database:**It is the great feature provided by the MongoDB. A Schemaless database means one collection can hold different types of documents in it. Or in other words, in the MongoDB database, a single collection can hold multiple documents and these documents may consist of the different numbers of fields, content, and size. It is not necessary that the one document is similar to another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases.
* **Document Oriented:**In MongoDB, all the data stored in the documents instead of tables like in RDBMS. In these documents, the data is stored in fields (key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document

contains its unique object id.

* **Indexing:**In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.
* **Scalability:**MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.
* **Replication**:MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.
* **Aggregation**:It allows to perform operations on the grouped data and get a single result or computed result. It is like the SQL GROUPBY clause. It provides three different aggregations i.e., aggregation pipeline, mapreduce function, and singlepurpose aggregation methods.
* **High Performance**:The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.

Django is a Python-based free and open-source web framework that follows the modeltemplate-view (MTV) architectural pattern. It is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel.

##### Features of Django

* **Ridiculously Fast:**Django was designed to help developers take

applications from concept to completion as quickly as possible.

* **Reassuring Secure:**Django takes security seriously and helps developers avoid many common security mistakes.
* **Framework Support:**Django has built-in support for Ajax, RSS, Caching and various other frameworks.
* **Administration GUI:**Django provides a nice ready-to-use user interface for administrative activities.
* **Development Environment:**Django comes with a lightweight web

server to facilitate end-to end application development and testing.

#### 4.3.4 Google Vision

Recently, I covered how computers can see,hear,feel,smell ad taste.One of the ways your code can “see” is with the Google Vision API. Google Vision API connects your code to Google’s image recognition capabilities. You can think of Google Image Search as a kind of API/REST interface to images.google.com, but it does much more than show you similar images.Google Vision can detect whether you’re a cat or a human, as well as the parts of your face. It tries to detect whether you’re posed or doing something that wouldn’t be okay for Google Safe Search—or not. It even tries to detect if you’re happy or sad.

#### 4.3.5 Open CV

Open CV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Open CV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects. classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. Open CV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million.. It has C ++, Python, Java and MATLAB interfaces and supports Windows, Linux. Android and Mac OS. Open CV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and Open CL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. Open CV is written natively in C++ and has a template interface that works seamlessly with STL containers.

#### 4.3.6 TensorFlow

TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy. If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

#### 4.3.6 Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

**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 detect 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 out pass 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 personnel’s.

* **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. They 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** :

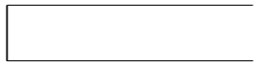


Figure 5.1 DFD Components

**5.2.1 Project DFD**

**Level 0 (Context Level):**

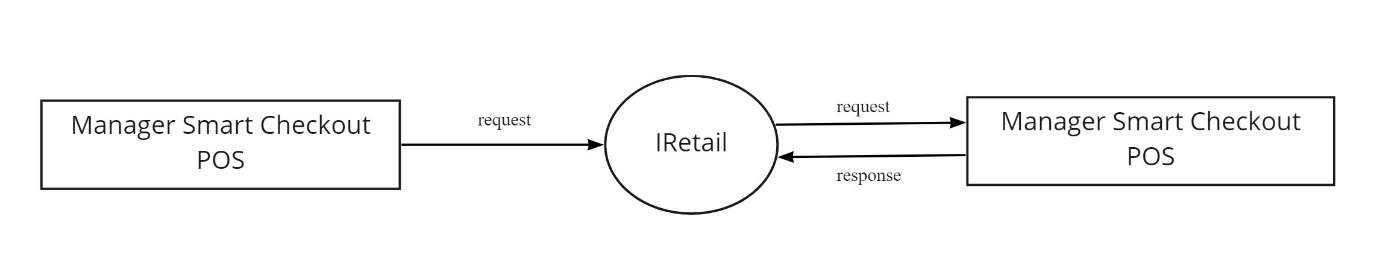


Figure 5.2 Context Diagram (Level 0)

**Level 1 :**

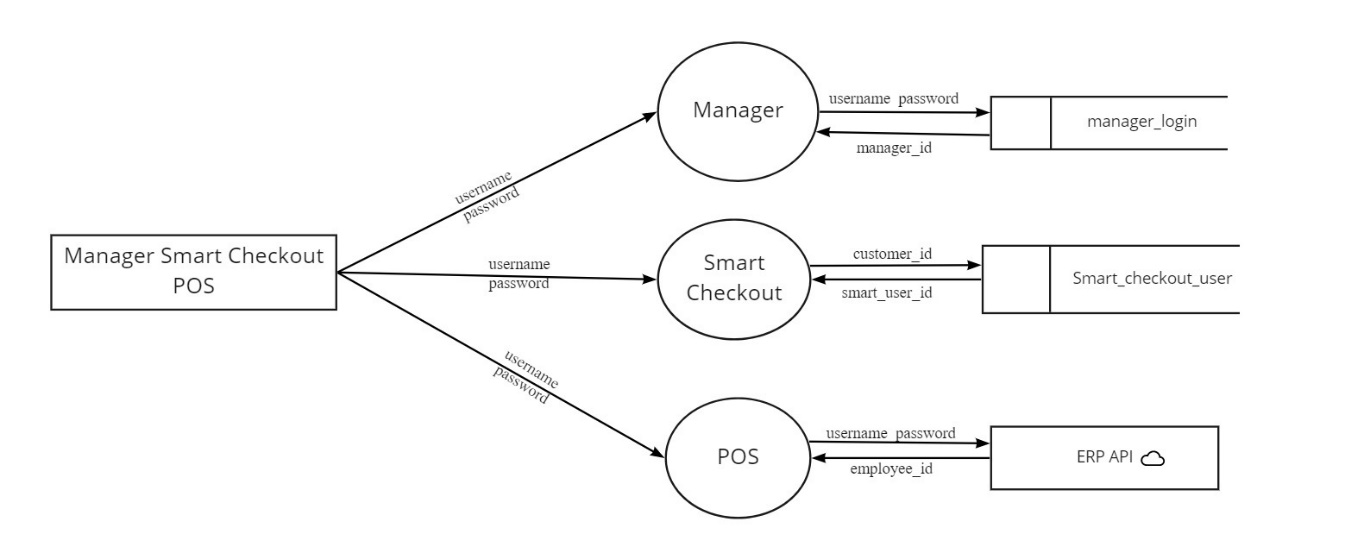
****

Figure 5.3 Module Description (Level 1)

**Level 1.0: Manager Module**

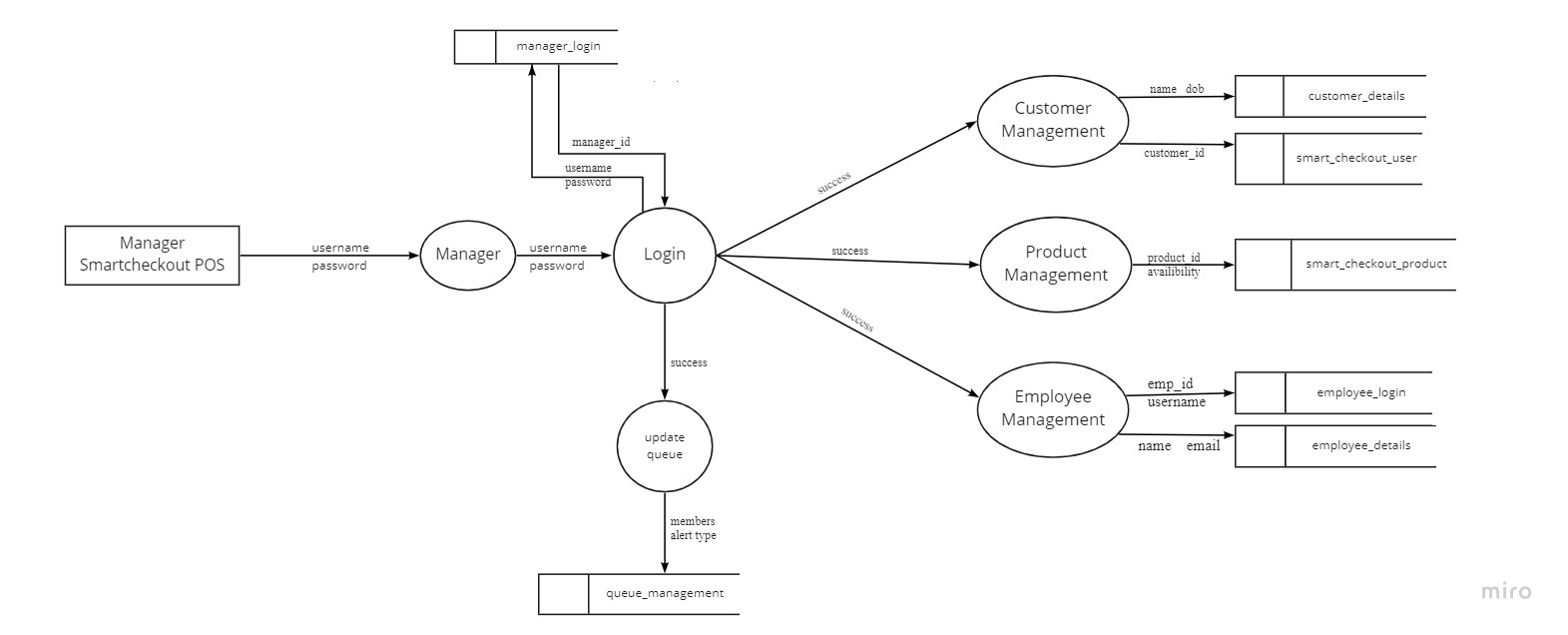


Figure 5.4 Manager Login (Level 1.0)

**Level 1.0.1: Customer Management**

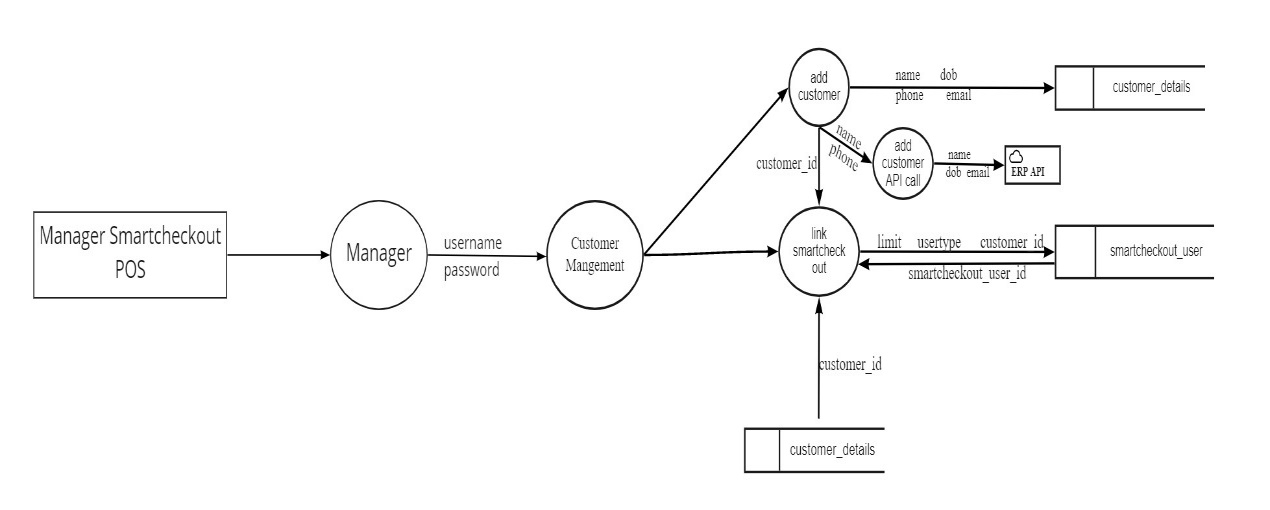


Figure 5.5 Customer Management (Level 1.0.1)

**Level 1.0.2 : Product Management**

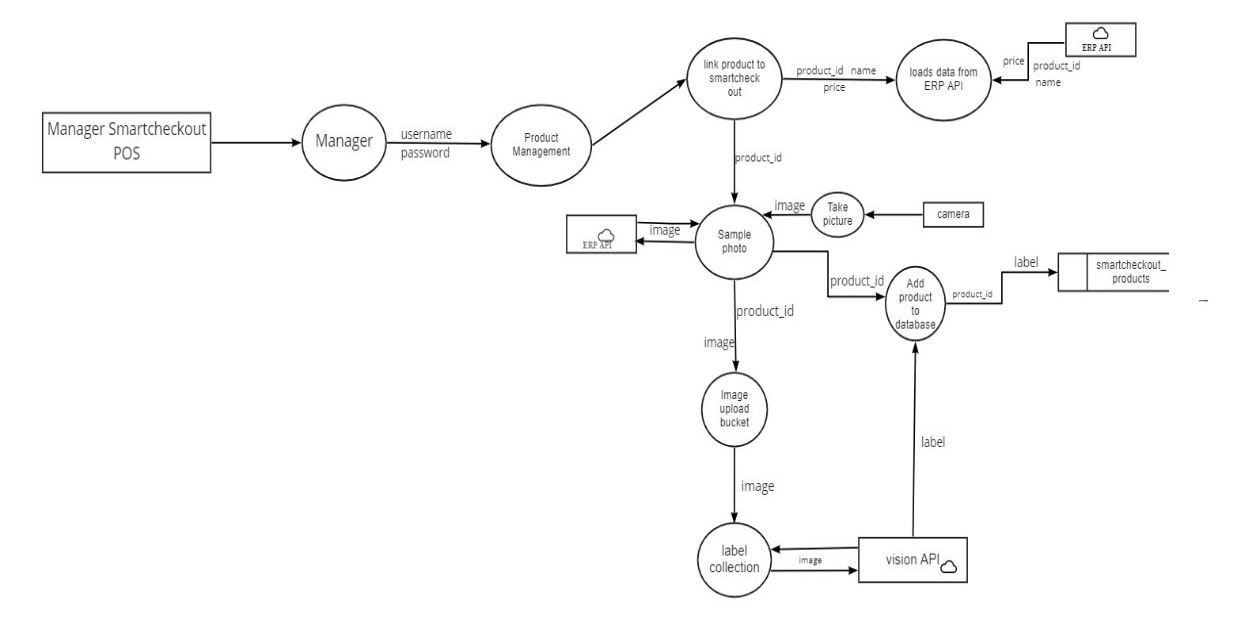


Figure 5.6 Product Management (Level 1.0.2)

**Level 1.0.3: Employee Management**

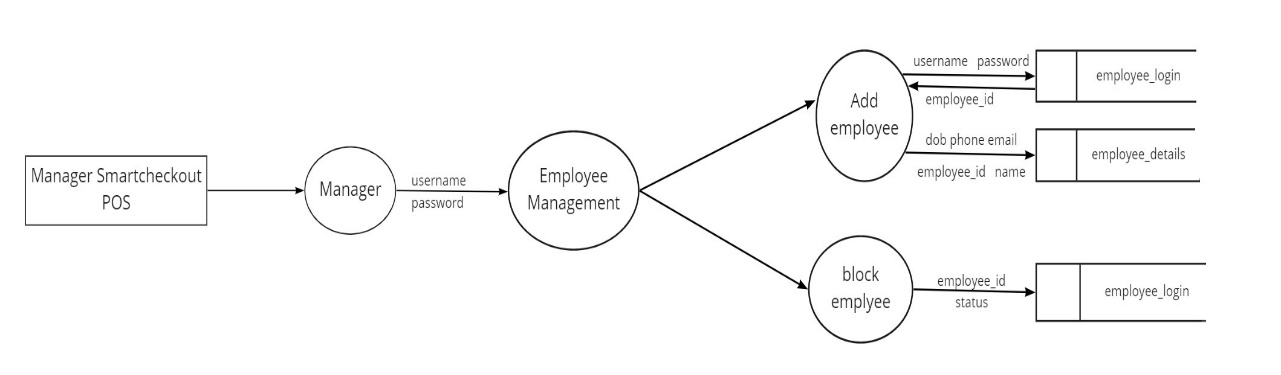


Figure 5.6 Employee Management (Level 1.0.3)

**Level 1.1: Smart Checkout**

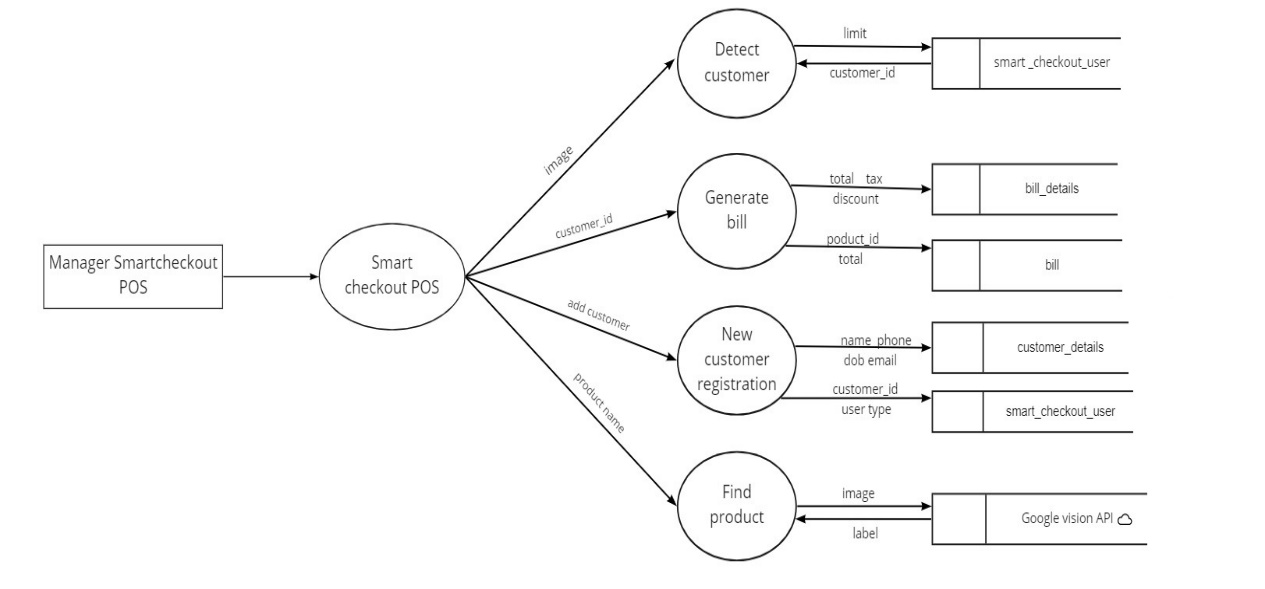
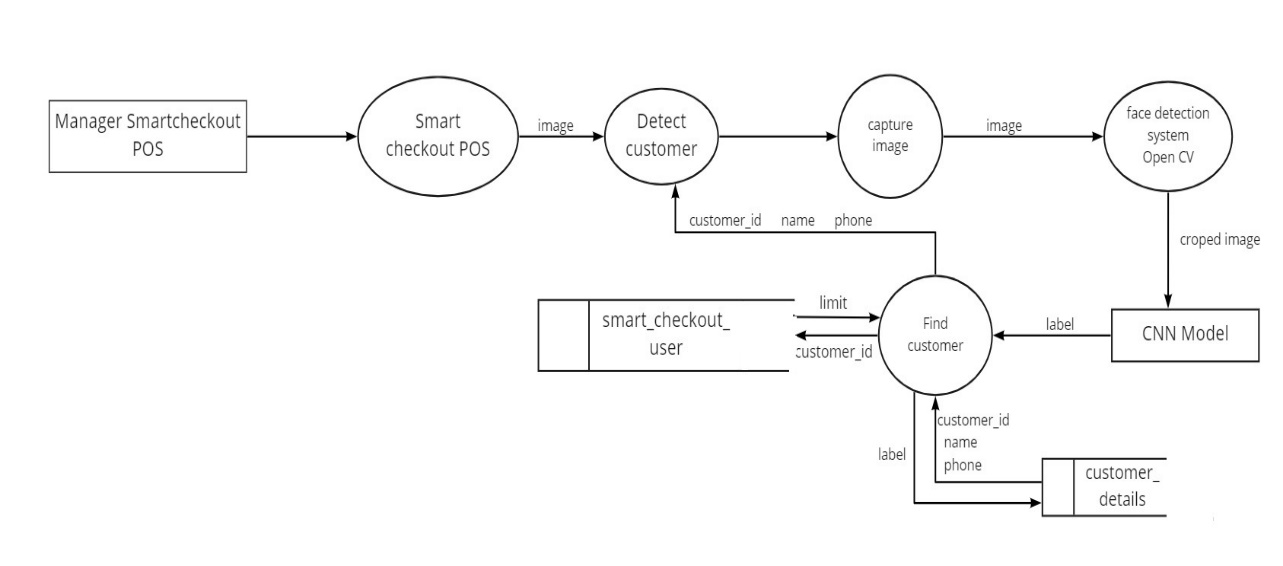
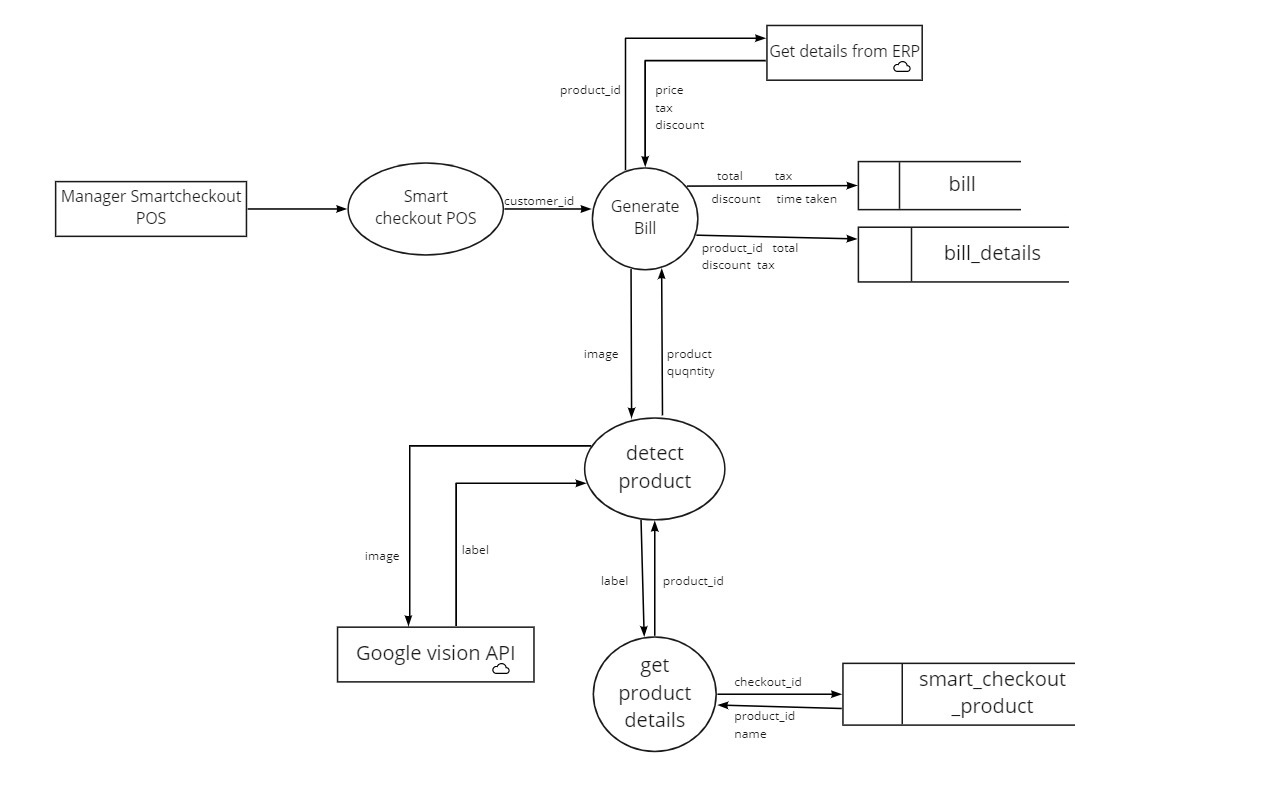
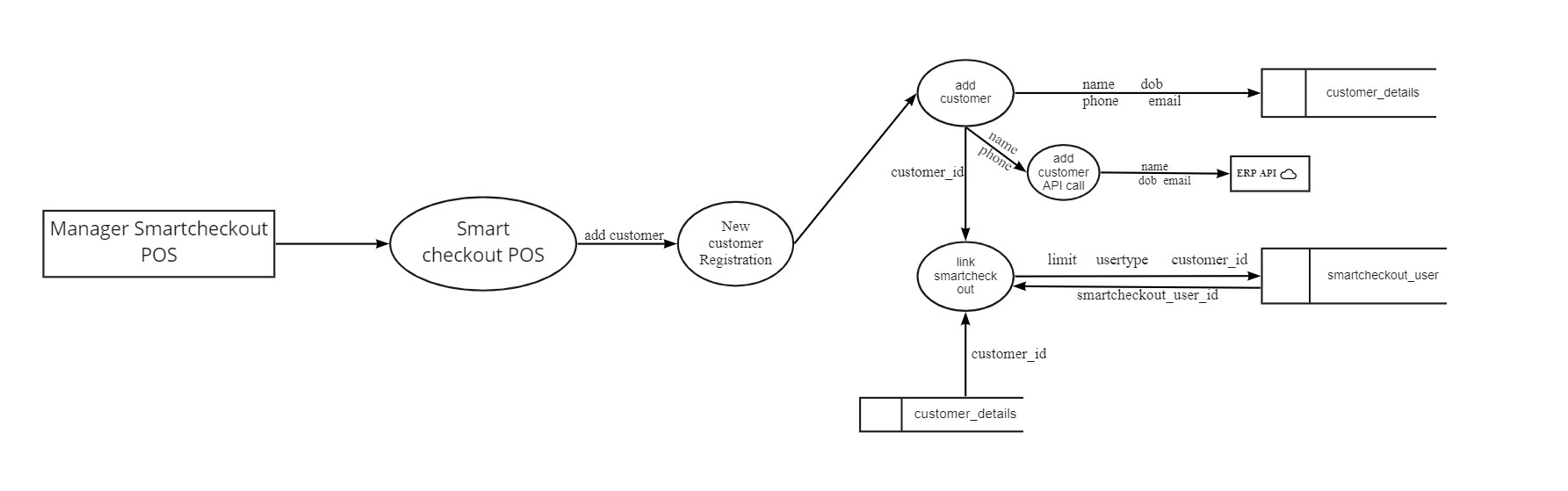


Figure 5.7 Smart Checkout (Level 1.1)



****

****

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

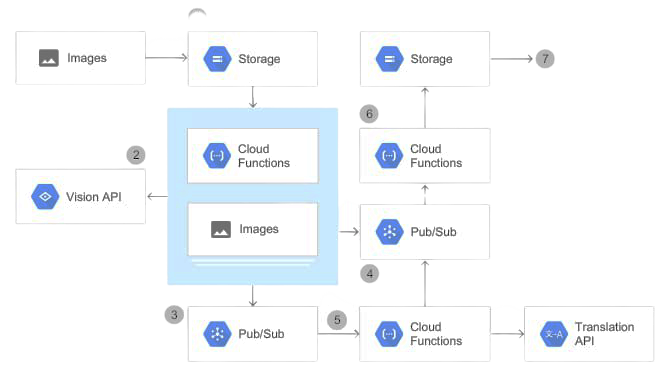
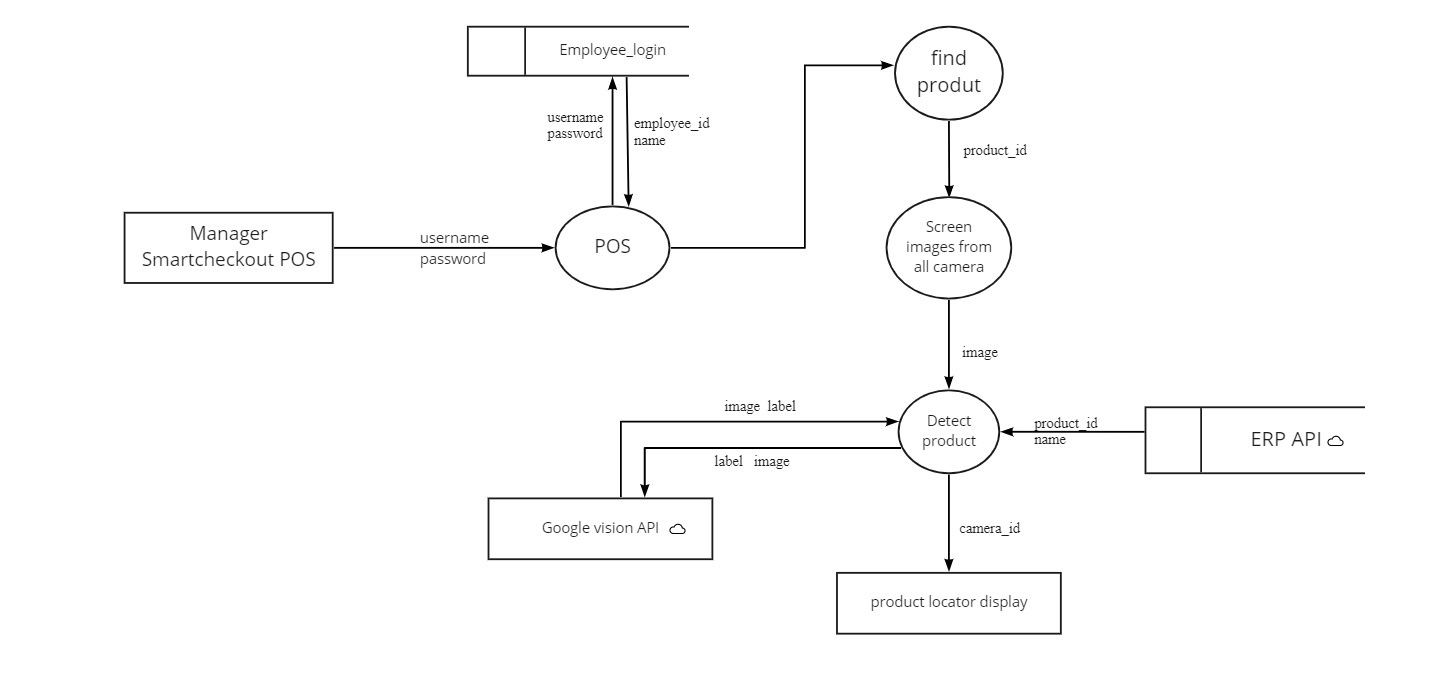


Figure 5.8 Find Product (Level 1.1.2)

**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.

|  |  |  |
| --- | --- | --- |
| **Process** | **Input design** | **Output design** |
| Login page for manager | Enter user name, password | Show home page |
| Login page for smart checkout | Enter user name, password | Show home page |
| Login page for POS | Enter user name, password | Show home page |
| Add customer | Enter customer details | Customer added successfully |
| Link smart check out | Enter smart check out details | Linked successfully |
| Link product to smart check out | Select product details | Linked successfully |
| Take photo | Using camera | Show product details |
| Add employee | Enter employee details | Employee added successfully |
| Block employee | Enter employee details | Blocked successfully |
| Find customer | Enter customer details | Customer identified successfully |
| Detect product | Enter product image | Product identified successfully |

Table 5.10 Input & Output Design

###### **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 from ERP API, 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: Once logged in the manager has the privilege to manage billing,detect customer,new customer registration,find product.

Step 3: Smart checkout function 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

iv. 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 check out 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 behaviour, 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 behaviour of the system. This behaviour 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 non-functional 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 strategies

An engineered product can be tested in one of these two ways. These testing strategies include:

* Black box testing
* White box testing

### White box testing

White-box testing is a method of testing the application at the level of the source code. White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are chooses inputs to exercise paths through the code and determine the expected outputs.

**Black box testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

## 7.2 Unit testing

In computer programming, unit testing is a software method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures ,are tested to determine whether they are fit for use intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.

Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. If forms the basis for component testing. In the project each module is tested individually and is found to be an error free one.

## 7.3 Integration testing

This is the final step in testing. In this case all the modules were combined and given the test data. The combined module works successfully without any side effect on other programs. Everything was found to be working correctly.

In this the entire system was tested as a whole with all modules. This form of testing is popularly known as Black Box testing or system testing. Black Box testing methods focus on the functional requirement of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external database access, performance errors and initialization errors and termination errors.

In this project 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.

## 7.4 System testing

Software testing is critical element of software quality assurance and represents the ultimate review of specifications, design and code generation. System testing is the stage of implementation, it is aimed for ensuring that the system works accurately and efficiently before live operations commences. Nothing is complete without testing, as it is vital success of the system.

Testing Objectives:

There are several rules that can serve as testing objectives, they are

* Testing is a process of executing a program with the intent of finding an error
* A good test case is one that has high probability of finding an undiscovered error.
* A successful test is one that uncovers an undiscovered errors.

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for used to design test cases. The tester effective and consistent regression testing.

## 7.5 Testing Results

Table number: 7.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.No** | **Test Case** | **Input** | **Expected Output** | **Result**  **Pass/fail** |
| 1 | Login Page  (Manager) | Correct username and password | Manager Login | Pass |
| 2 | Login Page  (Manager) | Incorrect username and password | Invalid username or password | Fail |
| 3 | Login Page  (Smartcheck out) | Correct username and password | Smartcheck out Login | Pass |
| 4 | Login Page  (Smartcheck out) | Incorrect username and password | Invalid username or password | Fail |
| 5 | Login Page  (POS) | Correct username and password | POS Login | Pass |
| 6 | Login Page  (POS) | Incorrect username and password | Invalid username or password | Fail |
| 7 | Add Customer | Name,dob,phone number,email | Customer added | Pass |
| 8 | Link smart check out | Usertype,customer id,smart check out user id | Linked successfully | Pass |
| 9 | View registered customer details | Name,city,phone number,total purchase | View registered customer details | Pass |
| 10 | Add products | Taking image by placing in correct marked position | Product added successfully | Pass |
| 11 | Add products | Placed in incorrect postion | Please place the product in correct position | Fail |
| 12 | Link product to smart checkout | Product id,name,price | Linked successfully | Pass |
| 13 | Add employee | Name,phone number,email,city,  username,  Password,retype password,address | Employee added | Pass |
| 14 | Block employee | block | Blocked succesfully | Pass |
| 15 | View employee | Employee id,name,phone,email  ,username | View employee details | Pass |
| 16 | Find customer | Image | Registered customer | Pass |
| 17 | Find customer | Image | Unregistered customer | Fail |
| 18 | Detect product | Products placed infront of camera | Show product details and quantity | Pass |
| 19 | Smartcheck out | Scan products placed infront of camera | View bill details | Pass |
| 20 | Link product | Image(product id,name,price) | Linked successfully | Pass |
| 21 | User view | Bill number,date,  time,price,tax,total | Checkout details | Pas |
| 22 | Find product | Product id,name | Show product place (section and row)and image | Pass |
| 23 | Find product | Incorrect product id,name | Product not available | Fail |

CHAPTER 8

RESULTS AND DISCUSSION

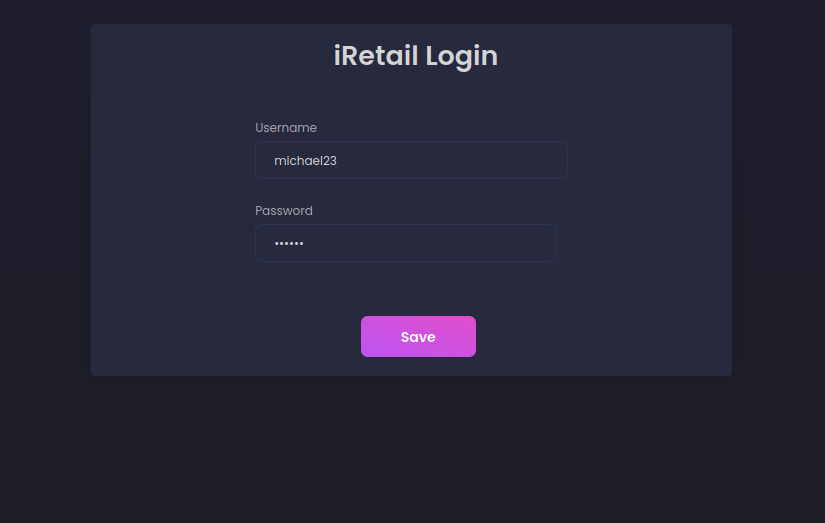
##### 8.1 Results (Salient Features)

The main motivation and objective of this system is to provide a solution to reduce the inconvenience in finding a unique diet for the user. Enabling the user experiencing a friendly user interface. Systematic handling of the schedules in such a way is a key to increasing its manageability and its competence. The proposed system incorporated with the following features.

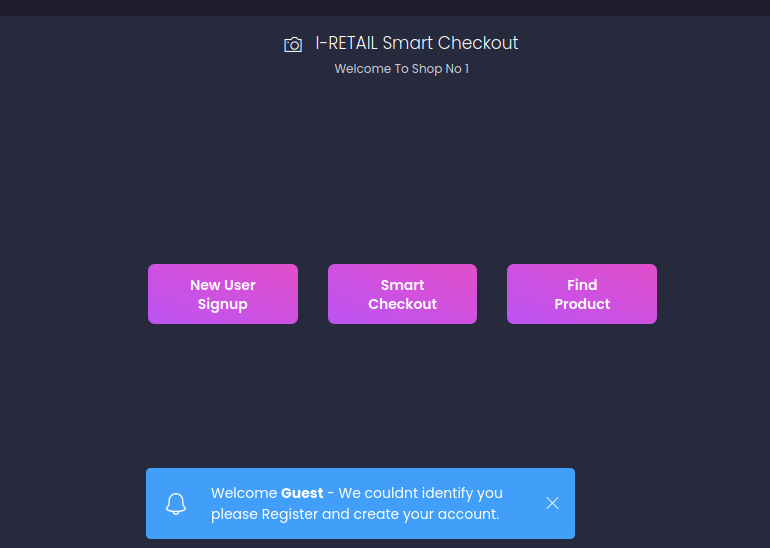
* Human effort can be reduced
* Improved efficiency
* Dynamic Price prediction
* Face recognition
* Customer identification
* Identify products
* Smart checkout system
* Image based product search
* Billing
* Categorize Customers

##### 8.2 SCREEN SHOTS

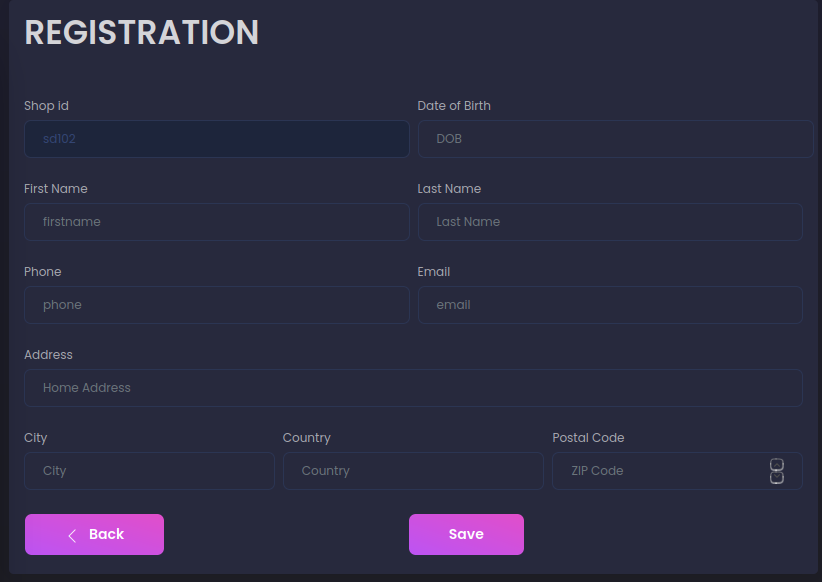
1. **Manager Login**



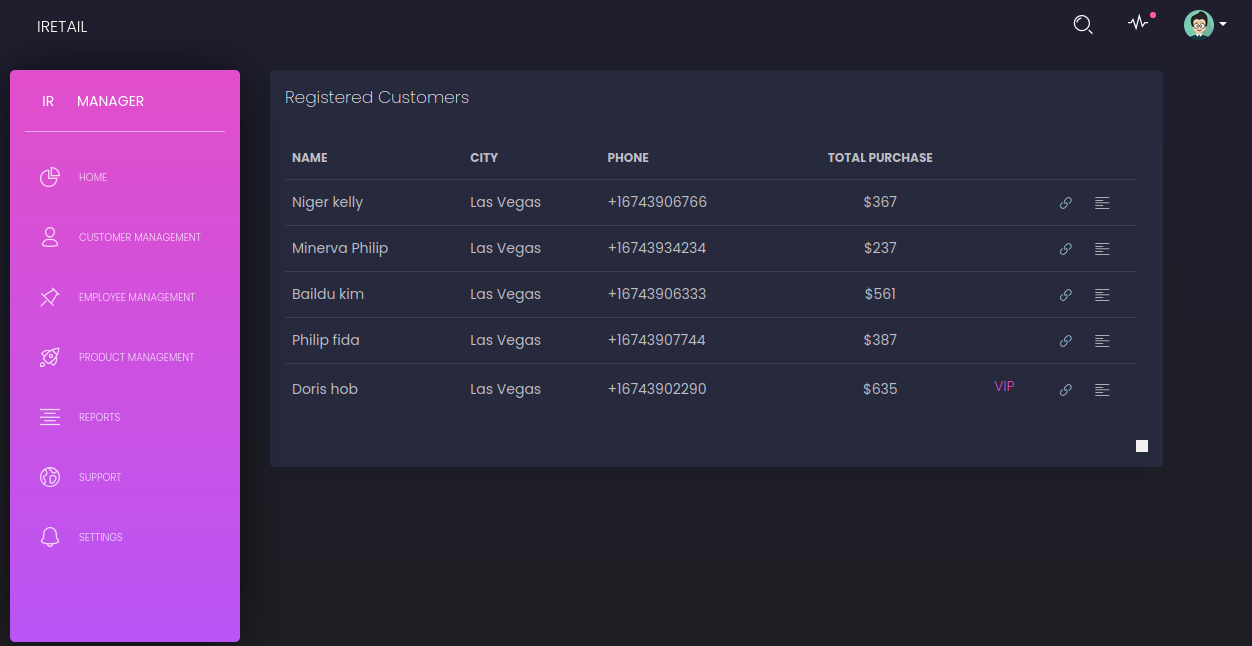
1. **Home**



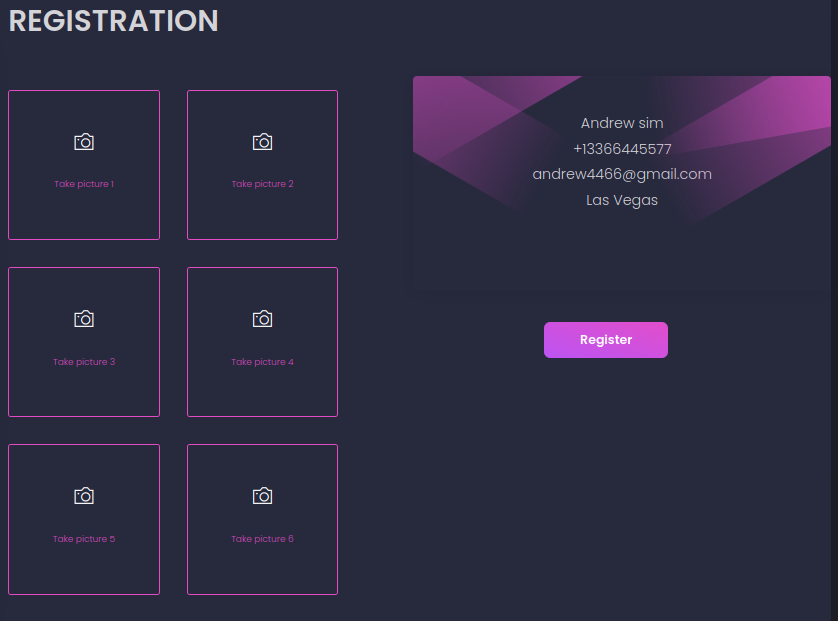
1. **Customer Registration**



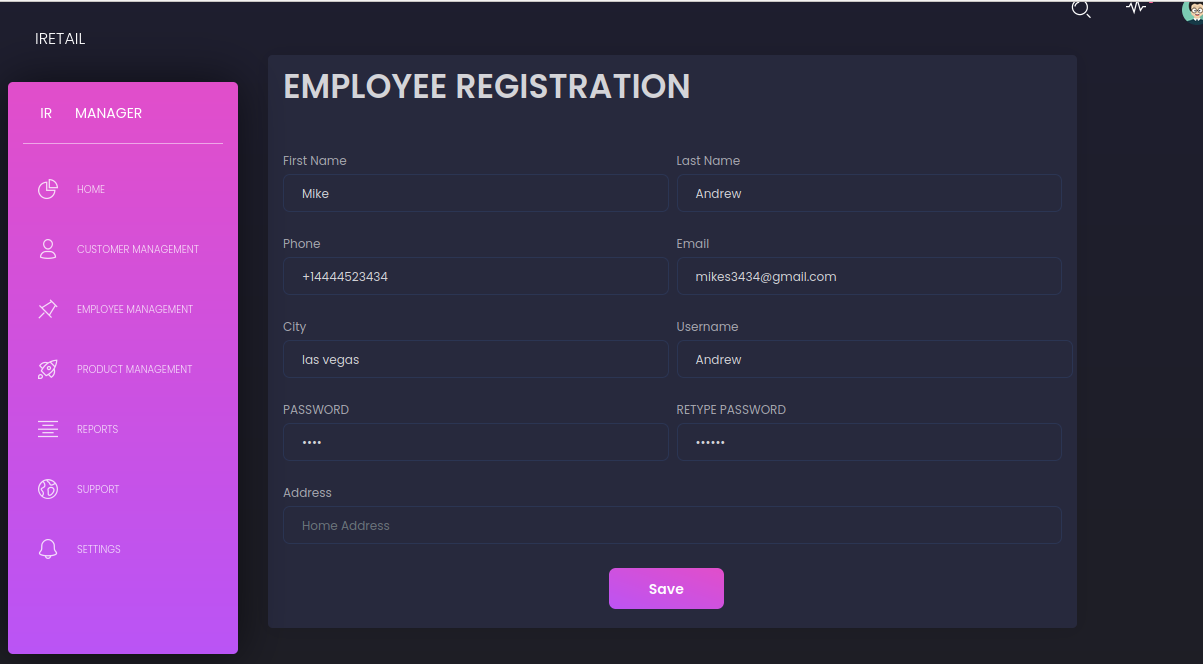
1. **View Customers**



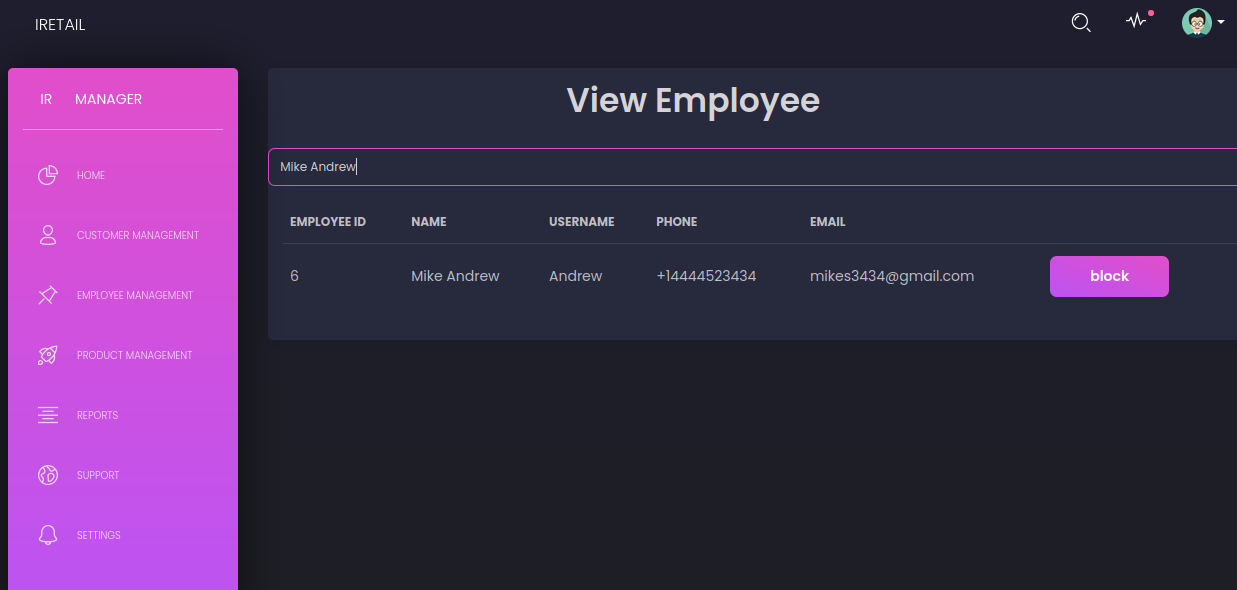
1. **Registration-2**



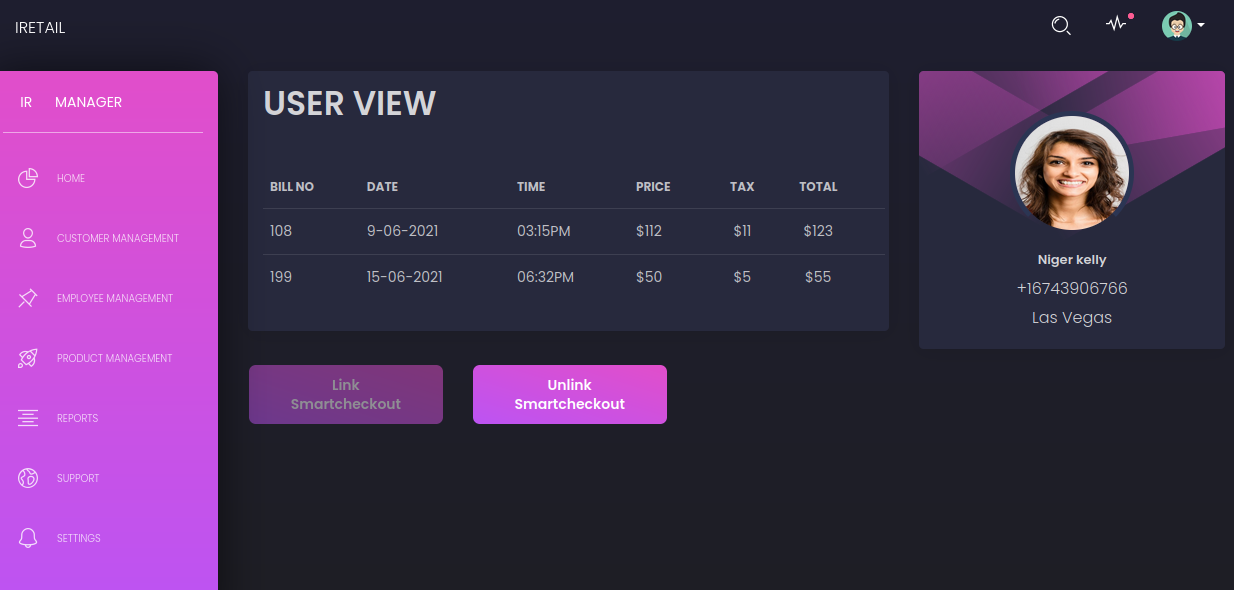
**vi.EmployeeRegistration**



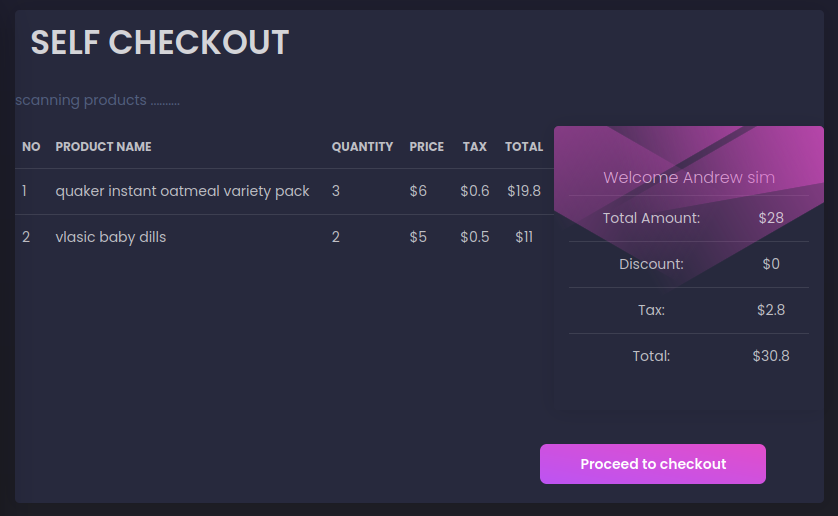
**vii.View Employee Details**



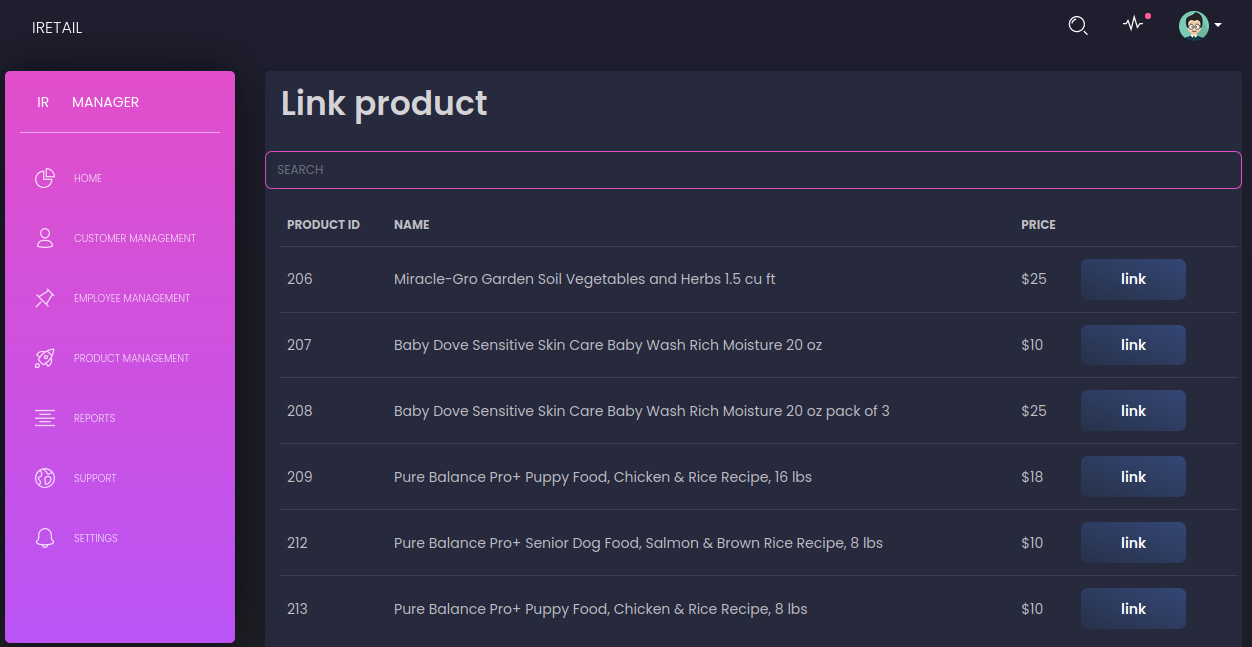
**viii.Link Smart Checkout**



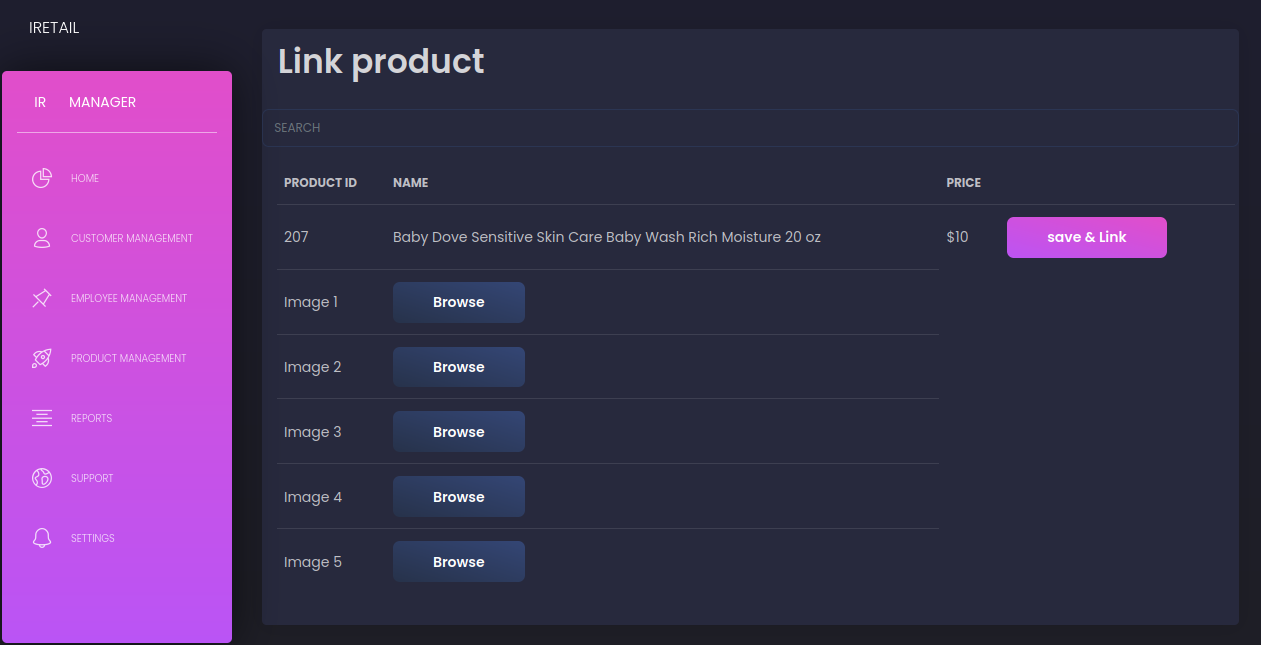
1. **Self Checkout**



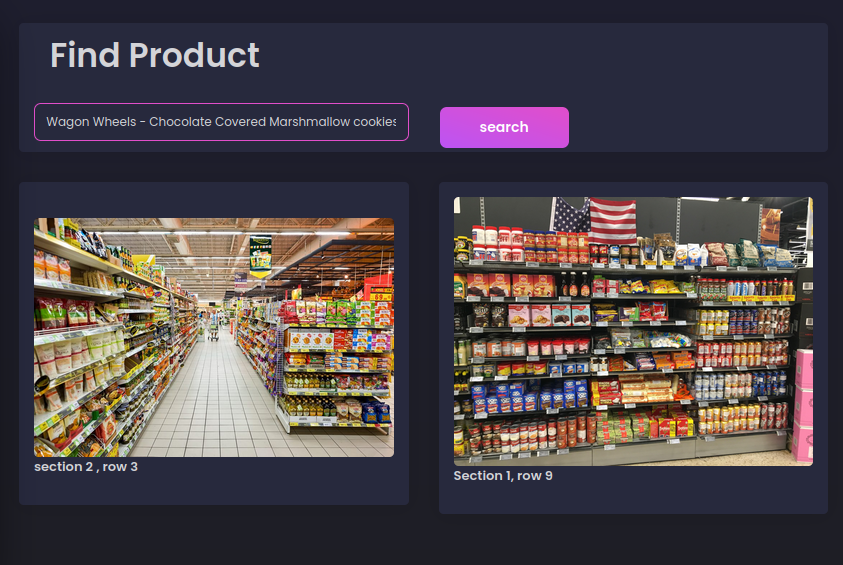
1. **Link Product-1**



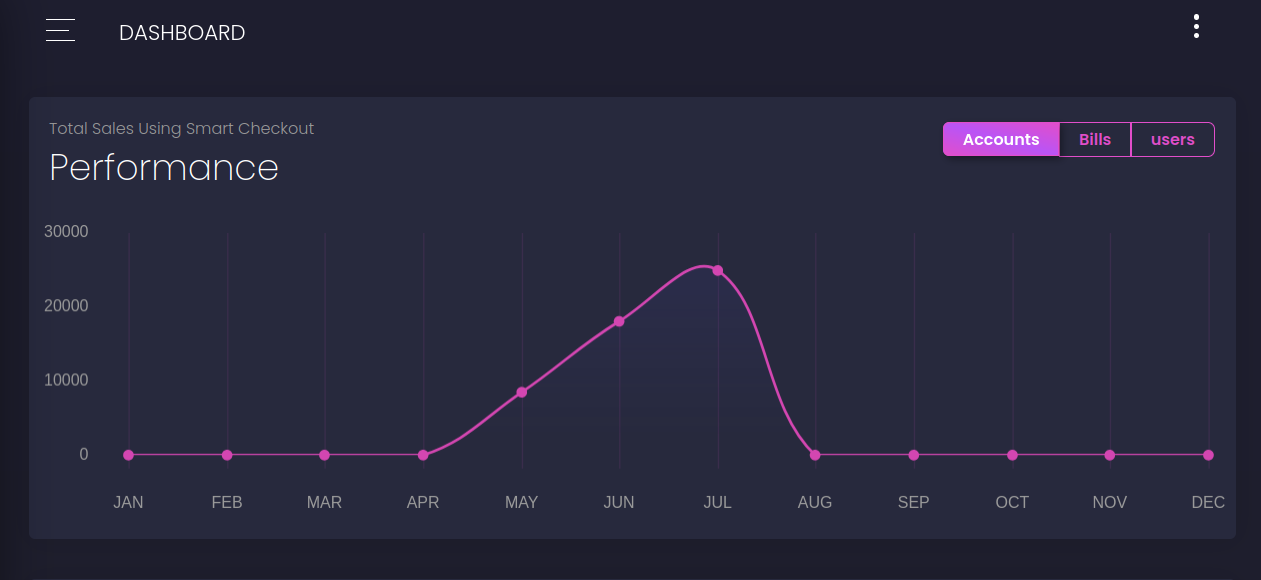
1. **Link Product -2**



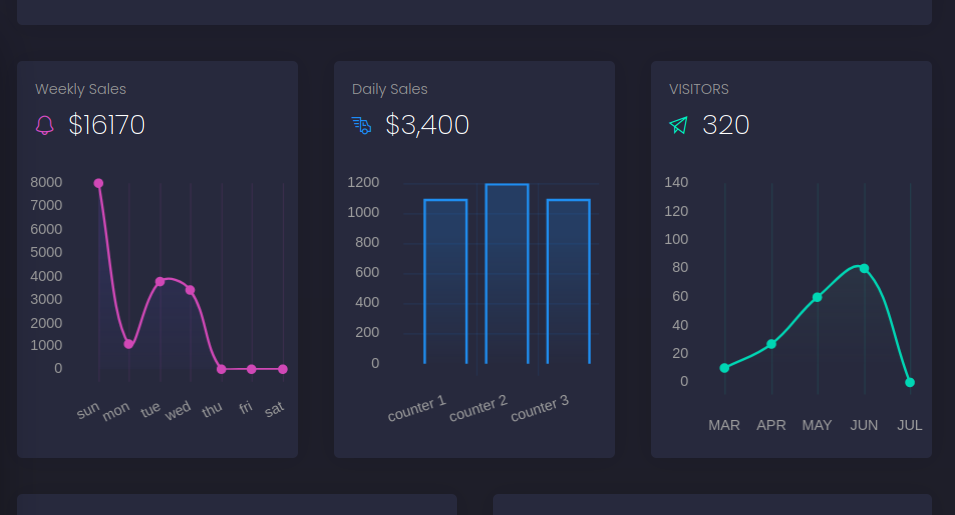
1. **Find Product**



1. **Report-1**



1. **Report-2**



### CHAPTER 9

### CONCLUSION

##### 9.1 System Implementation

##### The purpose of System implementation as making the new system making available to be prepared set of users and positioning ongoing support and maintenance of the system within the performing organization. At a final level of detail, deploying the system consists of executing all steps necessary to educate the consumers on the use of a new system, placing the newly developed system into production, continuing that all data required at the start of the operations is available and accurate, and validating that business functions that interact with the system are functioning properly.

**9.2 Conclusion**

Using item identification method, the system detects any object in the image with outlined rectangular boxes and classify each object in the image and place its tag with these methods and algorithms through the deep learning machines. The object was identified through the process of training of datasets. Thousands of images for each object was taken to improve the accuracy. Then, the object in the image was outlined and labelled to be identified during real-time detection. Upon completion of training of datasets, each object can be identified with the proper labelling. Hence, more research and improvement are done on object detection to create a better algorithm. Retailers can easily gain insight on transactions, interests, and hotspots. When it comes to efficiency, kiosks offer the following:

* Reduced wait times compared to using cash registers. The customers can just walk in the store, order and pay without needing to wait in the line.
* Reduced labour costs as one member of staff can overlook several self-checkout kiosks.

As the items are usually still handed out by employees, or can be picked up by accessing digital lockers, the risk of theft is minimised with this solution.In terms of speed of transaction, this system is ahead some, but less advanced than some other options. Perceived control, reliability, ease of use and enjoyment are as optimal as they can be for customers using this system. All in all, next to online pre-ordering and checkout, kiosks are one of the most convenient self-checkout options for retailers selling meals, and should also be considered for other types of products and stores.

**9.3 Future Enhancement**

Consumers preferring self-service over human contact and the number increasing over time, retailers should consider adding some form of self-checkout to their stores.With an increased customer expectation for retailers to make the shopping experience more convenient and personalised, self-checkout is becoming a standard, and mobile self-checkout a desirable upgrade.

Since self-checkout increases risks of theft, it is important to have good security and random checks, but also train staff to make sure that customers are made as comfortable with these checks as possible.

* Self-checkout units with RFID: Retailers can label all their products with RFID tags. When costumers place their shopping bag on the counter, all the products get scanned at once without any effort and counter displays a full list of the products and total price.
* Weight security is made more flexible by the ability to set weight security thresholds for entire categories and subcategories of items rather than solely for individual items.
* Can make it more accessbile and friendly for customer with adding technologies using IOTs,Robots etc.Customers can get in to the shop using their card, simply take the product and walk out from the shops.

BIBLIOGRAPHY

#### Books

* Kenneth S.Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process,First Edition
* Allen B. Downey Think Python: An Introduction to Software Design, Second Edition
* Geeta kumari, Neha Kaleramna and K.M. Pandey(1015), ‘study on Performance management system of private companies: a case study of endurance Pvt.Ltd

#### Website

* https://www.javatpoint.com/python-tutorial
* <https://www.tutorialspoint.com/python/index.html>
* <https://www.w3schools.com/python/python_mongodb_getstarted.asp>
* <https://www.youtube.com/watch?v=hK9-_s55CBY>
* <https://cloud.google.com/vision>

#### 3.Journal and Publications

# Smart Self-Checkout Carts Based on Deep Learning for Shopping Activity Recognition-2020 IEEE Conference paper

* Allen B. Downey Think Python: An Introduction to Software Design, Second Edition

### APPENDICES I

#### 1. SCRUM MODEL

**i. Git**

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed, data integrity, and support for distributed, nonlinear workflows.

**ii.Git Repositories**

A Git repository contains the history of a collection of files starting from a certain directory. The process of copying an existing Git repository via the Git tooling is called cloning. After cloning a repository the user has the complete repository with its history on his local machine. Of course, Git also supports the creation of new repositories. If you want to delete a Git repository, you can simply delete the folder which contains the repository. If you clone a Git repository, by default, Git assumes that you want to work in this repository as a user. Git also supports the creation of repositories targeting the usage on a server.

#### iii.Scrum

Scrum is an agile way to manage a project, usually software development. Agile software development with Scrum is often perceived as a methodology; but rather than viewing Scrum as methodology, think of it as a framework for managing a

process. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, 77 much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. In the agile Scrum world, instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it is left up to the Scrum software development team. This is because the team will know best how to solve the problem they are presented. Within agile development, Scrum teams are supported by two specific roles. The first is a Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level. The product owner (PO) is the other role, and in Scrum software development, represents the business, customers or users, and guides the team toward building the right product.

#### iv Git History

### APPENDICES II

1. LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Name** | **Page No** |
| 5.1 | Manger\_login | 42 |
| 5.2 | queue\_management | 42 |
| 5.3 | customer\_details | 43 |
| 5.4 | smartcheckout\_product | 43 |
| 5.5 | smartcheckout\_user | 44 |
| 5.6 | Employee\_login | 44 |
| 5.7 | Employee\_details | 45 |
| 5.8 | bill | 45 |
| 5.9 | bill\_details | 46 |
| 5.10 | Inputs and Output | 46 |
| 7.1 | Testing Results | 58 |

**2. LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| 5.1 | DFD Symbols | 32 |
| 5.2. | Context Diagram (Level 0) | 33 |
| 5.3. | Module Description (Level 1) | 34 |
| 5.4. | Manager Login(Level 1.1) | 35 |
| 5.5. | Customer Management (Level 1.1.1) | 36 |
| 5.6. | Product Management (Level 1.1.2) | 37 |
| 5.7. | Employee Management(Level 1.1.3) | 38 |
| 5.8 | Smart Checkout (Level 1.2) | 39 |
| 5.9 | Detect Customer (Level 1.2.1) | 40 |
| 5.10. | Generate Bill | 69 |
| 5.11 | New Customer Registration | 69 |
| 5.12 | Find Product(Google vision API) | 70 |
| 5.13 | POS | 71 |
| i. | Manger Login | 71 |
| ii. | Home | 72 |
| iii. | Customer Registration | 72 |
| iv. | View Customer | 73 |
| v. | Registration-2 | 73 |
| vi. | Employee Registration | 74 |
| vii. | Employee Details | 74 |
| viii. | Smart checkout | 75 |
| ix. | Self Checkout | 75 |
| x. | Link Product-1 | 75 |
| xi. | Link Product-2 | 75 |
| xii. | Find Product |  |
| xiii. | Report-1 |  |
| xiv. | Report-2 |  |