In-Store Customer Analytics using Face Detection and Recognition

Final Year Project Report

by

Muhammad Saad Tariq [224921]

Osama Akhlaq [213452]

Ershad Hussain [237885]

In Partial Fulfillment

Of the Requirements for the degree

Bachelors of Engineering in Software Engineering (BESE)

School of Electrical Engineering and Computer Science

National University of Sciences and Technology

Islamabad, Pakistan

(2021)

DECLARATION

We hereby declare that this project report entitled “In-Store Customer Analytics using Face Detection and Recognition” submitted to the “Department of computing”, is a record of an original work done by us under the guidance of Supervisor “Dr. Moazam Fraz” and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Computer Science.

Team Members	**Signature**
Muhammad Saad Tariq	
Osama Akhlaq	
Ershad Hussain	

Supervisor:	**Signature**
Dr. Moazam Fraz	_____

Date:

1st June 2021

Place:

School of Electrical Engineering and Computer Science - NUST

DEDICATION

To our Parents, Teachers and our Dreams

ACKNOWLEDGEMENTS

First and foremost, we thank the Almighty for making it possible for us to complete a project of this scale in such troubled times during a global pandemic. We are humbly thankful to our Supervisor Dr. Moazam Fraz and our Co-Advisor Dr. Muhammad Shahzad for guiding us to take up a project that has such vast practical implications and really allowed us to build our forte in various fields of the subject. Lastly, we would like to acknowledge ourselves for managing to work remotely and as a team, overcoming all challenges that we faced during the completion of this project.

TABLE OF CONTENTS

DECLARATION 2
DEDICATION 3
ACKNOWLEDGEMENTS 4
TABLE OF CONTENTS 5
LIST OF FIGURES 7
ABSTRACT 8
INTRODUCTION 9
LITERATURE REVIEW 13
PROBLEM DEFINITION 17
METHODOLOGY 18
DETAILED DESIGN AND ARCHITECTURE 20
5.1 SYSTEM ARCHITECTURE 20
5.1.1 Architecture Design Approach 20
5.1.2 Architecture Design 21
5.1.3 Subsystem Architecture 21
5.2 DETAILED SYSTEM DESIGN 23
5.2.1 Client Application 23
5.2.1.1 Classification 23
5.2.1.2 Definition 24
5.2.1.3 Responsibilities 24
5.2.1.4 Constraints 24
5.2.1.5 Composition 25
5.2.1.6 Interactions 25
5.2.1.7 Resources 25
5.2.1.8 Processing 25
5.2.1.9 Interface/Exports 25
5.2.1.10 Detailed Subsystem Design 32
5.2.2 APIs 32
5.2.2.1 Classification 32
5.2.2.2 Definition 32
5.2.2.3 Responsibilities 33
5.2.2.4 Constraints 33
5.2.2.5 Composition 34
5.2.2.6 Interactions 34

5.2.2.7 Resources 34
5.2.2.8 Processing 34
5.2.2.9 Interface/Exports 35
5.2.2.10 Detailed Subsystem Design 35
5.2.3 Server 36
5.2.3.1 Classification 36
5.2.3.2 Definition 36
5.2.3.3 Responsibilities 36
5.2.3.4 Constraints 36
5.2.3.5 Composition 36
5.2.3.6 Interactions 37
5.2.3.7 Resources 37
5.2.3.8 Processing 37
5.2.3.9 Interface/Exports 37
5.2.3.10 Detailed Subsystem Design 37
5.2.4 MongoDB Database 38
5.2.4.1 Classification 38
5.2.4.2 Definition 38
5.2.4.3 Responsibilities 38
5.2.4.4 Constraints 38
5.2.4.5 Composition 39
5.2.4.6 Interactions 39
5.2.4.7 Resources 39
5.2.4.8 Processing 39
5.2.4.9 Interface/Exports 39
5.2.4.10 Detailed Subsystem Design 40
5.3 COMPONENT DIAGRAM 41
IMPLEMENTATION AND TESTING 42
RESULTS AND DISCUSSION 44
CONCLUSION AND FUTURE WORK 45
REFERENCES 46

LIST OF FIGURES

Figure 1. Methodology Overview 19
Figure 2. Architecture Design Approach 20
Figure 3. Architecture Design 21
Figure 4. Subsystem Architecture 23
Figure 5. Customer Detection UI 26
Figure 6. Purchase Recommendation UI 26
Figure 7. Updating Existing Customer UI 27
Figure 8. Adding new Customer UI 27
Figure 9. Sales Report UI 28
Figure 10. Sales Dashboard UI 28
Figure 11. Customer Analytics I 29
Figure 12. Customer Analytics II 29
Figure 13. Item List UI 30
Figure 14. Edit Product 30
Figure 15. Create Adjustment UI 31
Figure 16. View Order UI 31
Figure 17. Client Application Design 32
Figure 18. API structure 35
Figure 19. Server Design 37
Figure 20. Node JS server abstraction 38
Figure 21. Mongo Atlas Interface 40
Figure 22. Database Design 40
Figure 23. Component Diagram 41

Table 1. Application Configuration Specifications 44

ABSTRACT

Face detection and recognition are widely used methods in the field of computer vision and are being implemented in numerous large-scale applications. Statistics about customer behaviors and patterns are extremely valuable for businesses such as retail stores as it helps develop efficient business strategies. We, therefore, present an automated web-based approach to study and display customer behavior through in-store analytics. Our system combines face detection and recognition, best view estimation, repeat customer identification, and customer purchasing behavior analysis using an advanced web-based POS dashboard.

Chapter 1

INTRODUCTION

Face Recognition and Verification are critical modern methods that have vast applications in the world today. Face recognition refers to the process of matching a face against one or more known faces from an existing database; whereas face verification refers to confirming whether two inputs belong to the same person or not. Conventionally, there are 4 primary stages in the face recognition system: face detection, face alignment, face representation, and face matching [binary]. Face detection refers to identifying a human face from a given input, while face alignment is the technique to further analyze the geometric structure of the face and distinguish between all facial components. This extracted information needs to be stored using a mathematical representation which is defined as face representation, and finally, the stored representations of different faces are compared with each other to identify the same faces – known as face matching. Face recognition is classified as a pattern recognition problem with the 2 major components being face representation and face matching. For face representation the goal is to extract distinguishing features to be able to discriminate between different images while for face matching it is to build classifiers that can discern different face patterns [1].

Face representation tends to have a drastic effect on face recognition systems since input images can be greatly affected by multiple environmental factors such as pose, expression, light intensity, backgrounds, and image resolutions which contribute to reducing the similarity of face samples belonging to the same individual [1]. These representations used in detection systems are termed as descriptors. These descriptors are fundamentally vectors consisting of quantized facial features, allowing convenient storage of facial information along with the ability of determining similarities.

The ability for systems to successfully detect and recognize human faces has had vast applications in various fields of security, surveillance, banking, and retail etc. over

the past years and continues to rapidly expand. Recently, this technology has found its way into the retail market where it is being used to extract valuable insights pertaining to different retail sections such as product tracking, customer behavior detection, customer location tracking, staff monitoring as well as surveillance. Inferences from such analytics play a vital role in making paramount decisions and predictions affecting business performance [2].

Studying customer behaviors has always been the pinnacle of market strategy, and has been transformed with the possibility of automated face recognition. Customer behavior in retail stores deals with the identification of customers and their buying patterns – *who* buys *what*, *where*, *when*, *and how* [3]. This helps to learn about customer response to sale promotion devices and consequently adopt more effective marketing strategies [3]. Face recognition thus allows having an instantaneous match of a customer profile to their visual appearance and can verify the occurrence of the same individual at multiple instances, maintaining a comprehensive record of all their activities.

The shopping experience plays a central role in determining the success of the business in the context of retail, heavily impacting customer's purchase probability, satisfaction and loyalty. In simple terms, the more captivating the experience will be, the more engaged the customer is. We learn that the emotional impact of the customer is responsible for their relation with any brand, their behavior and decisions. Therefore, it becomes necessary to examine and analyze customer emotions and behaviors in order to positively affect their satisfaction standards in the future, and provide the best shopping experience – maximizing customer response [4]. Acquiring customer information and utilizing it not only enhances customer experience but boosts business development as well. Actively responding to customer feedback and considering customer sentiments create a friendly and cooperative ecosystem that has community-wide benefits.

Due to such rapid changes in the technology today, small retailers are struggling to keep up with the latest trends, and in countries like Pakistan where technology has

not really taken over the local business community, businesses lack the use of any sort of modern analytical tools. Customer service is also often neglected in the local community sphere, and is overlooked as a key business trait, thus stunting the growth and potential of many small-scale businesses. Lately, applications for identifying and tracking targets have already found their way into the retail market and such techniques can be used to extract useful data from customer interactions as an automated, effortless mechanism without the need for conventional lengthy membership processes and loyalty schemes.

Successful implementations of face recognition for retail analytics have paved the way for more efficient systems. [5]'s implementation of an individual face analysis for retail analytics at the Hom Krun coffee shop using convolutional neural networks and OpenCV for face detection, recognition, gender classification, age classification, and emotion classification provides a basis for the execution of such a system on real life physical environments. Similarly, [6] built an online, cost efficient customer recognition system that uses cloud computing resources to perform face recognition, age and gender estimation, with a collaborative tracking model. The system was tested in a controlled lab environment and has provided promising results, paving the path for field tests. [4] also present a robust emotional recognition platform that extensively studies customer behavior and actively keeps track of their sentiments, issuing warnings upon negative sentiment build ups. Although they use a complex back-end system to track customer features, it is implemented with a web-based user interface and a cloud-based environment.

Building up to the above-mentioned practical implementations of face recognition in retail, we observe the majority of the practices to be computationally extensive and require expensive hardware. We, therefore, propose a more active analysis of the customer behavior information through the use of video analytics, and image processing. We present an automated web-based approach to study and display customer behavior through in-store analytics. Our system combines face detection and recognition, repeat customer identification, and customer purchasing behavior analysis. Moreover, we propose using an advanced react based POS dashboard to be

integrated with our face recognition model and log customer purchases alongside, thus allowing retail owners to manage their sales analytics as an added feature as well.

Chapter 2

LITERATURE REVIEW

There has been extensive work done on the methods of face representation in recent years. Face representation can be broadly classified into two categories: holistic features and local features. Holistic features involve converting each face image into a high-dimensional feature vector lexicographically and consequently learn the feature subspace to store the facial information. Local features on the other hand, prescribe the structure pattern of each local component of the face - referred to as patch, and then combine the metrics of all patches to form a concatenated feature vector [1]. This however, tends to be more complex and computationally expensive and therefore most systems tend to use holistic face representations.

A number of feature learning methods have been developed in recent years amongst which convolutional neural networks (CNNs) have gained a lot of recognition. CNNs are a class of feed-forward neural networks consisting of many layers, comprising filters or kernels with learnable weights and biases. Each filter is fed some input and respectively performs convolution. Conventional CNNs for face recognition consist of 4 layers: Convolution layer, Pooling layer, ReLU layer, and the Fully Connected layer. The convolution layer is responsible for most of the computation. The purpose of this layer is to extract the features from the input face images; it maintains spatial relation between the pixels of the image by learning its features and outputs a feature map. The pooling layer then reduces the dimensionality of each feature map while retaining the most important information; the key purpose being generalization. The ReLU layer performs element wise operations on each pixel and reconditions all negative values in the feature map by zero and finally the fully connected layer classifies the input images to different classes from the training set based on their processed feature maps [7]. Such proposed architectures are computationally expensive though and require the use of extensive hardware. Therefore, a lot of the popular libraries and models implementing the CNN technology for face recognition

have now been configured for API use, in order to minimize load on the user side, allowing swift and easy use for custom applications.

Valter Borges, P. Duartes, A. Cunha implement a negative emotions detector using facial recognition API technology. They aim to identify customers in distress or in confusion and upon detection of negative emotions such as confusion, disgust, or sadness, the retail assistants are immediately informed to cater to the customer. The system uses Microsoft Azure's Face API to send captured images of customer faces and the API performs face detection, recognition, analysis and returns facial information in a JSON format. This information includes face ID, face landmarks, age, emotion, gender and hair [8]. A certain threshold is assigned a maximum limit and upon the percentage value of the negative sentiments surpassing that threshold, respective analysis is performed, and assistants are informed immediately. Similarly, [9] proposes using the Microsoft face API for face recognition to implement a smart door system. The application operates over Microsoft Visual Studio IDE, which is hosted over Microsoft Azure cloud to reduce the whole computational time. Feeds are sent to Microsoft Face API for recognition. When the user approaches the door, the user's face is captured and cropped out, which is then sent to the API. If the user's face is already present in the database, the user gets an audio "welcome User Name", and the relay module opens the door; otherwise, a new user needs to be added first [9]. Such systems are some of the applications of API technology and their use allows for greater abstraction, thus providing ease of integrating face recognition with customized applications.

There has been comprehensive contribution in the field of in-store video analytics in the past decade as In-store video analytics introduces a new dimension in delivering actionable insights, enabling strategies that enhance the shopping experience, stimulate shopper retention and increase sales. Retailers may improve shop performance and operations in a variety of ways by measuring hotspots, customer flow, dwell time, and product display activity and comparing patterns over time. The majority of the current work being done involves analyzing instore videos for theft and fraud detection, assessing the shopper's positive or negative appreciation of the

focused products by analyzing the non-verbal behavior of the shopper. Some of the earliest works date back to 2001 where Haritaoglu and M. Flickner suggested a method for counting waiting shopping groups in checkout queues, as well as the use of swarming algorithms to organize customers tracked throughout a store into shopping groups [10].

Several former works have addressed customer analytics problems at retails and markets through video understanding, transaction logs and customer metrics. A.W. Senior and L. Brown suggested a retail application where footages from multiple streams are collected and presented in a meaningful way such as alerts, reports and visualizations. A subsystem is deployed to manage the multiple routes and storage of footages as well compression and distribution protocols. On each surveillance stream, a smart surveillance engine is deployed which generates video analytics metadata and is sent to the main server running the Middleware for Large scale Surveillance (MILS). The main algorithms involved in the video analytics are object tracking and face tracking that extract object appearance, their trajectory and keyframe information [11]. Once a face is detected, correlation tracker is used to maintain tracking objects. These techniques are applied to monitor customers and their movements, and different forms of data are collected such as the amount of time a customer spends at a particular section, repetitive customer movements, head counts and traffic flow. These are then analyzed and targeted to challenge loss prevention - deficits in retail accounts and unnecessary losses - critical aspects that businesses are desperate to overcome. The information extracted also defends against return frauds by firmly tracking customer movements and providing a detailed account of transaction logs, integrated with a point of sales system [11].

In another study, Andrea Generosi, Silvia Ceccacci and Maura Mengoni propose to exploit emotional recognition tools to monitor customer experience in retail. They combine Facial Expression and Emotions Recognition, Age and Gender Recognition, Gaze Detection, Speech Recognition and Biofeedback Analysis to enhance customer shopping encounters. The proposed system uses an embedded CNN that takes face images as input and outputs the main Ekman’s emotions linked with the face as

percentage values for expressions and emotions recognition. The model is trained using two crowd-source datasets namely FER+ and EmotioNet trained on 36K and 1M images respectively. Another CNN trained with the IMDB-Wiki dataset on more than 500K labelled images is implemented for the age and gender recognition and open-source datasets contributed to the generation of the gaze detection module which analyzes every frame to detect the positioning of the face and pupil in order to track and identify products that might appeal to the customer. An OpenCV extension library is used for Biofeedback, which makes use of digital image processing to extract blood pulse rate and estimate breathing frequency, hence deducing customer's breath rate and heartbeat [4]. All these modules combine together to extensively study customer's behavior indications at any single point in time. All these modules process and feed data to a web GUI comprising a series of dashboards relaying customer information at every touchpoint installed.

The discussed literature provides an overview of the previously conducted work in the field of face recognition for customer analytics but also shows the vast areas of improvements that can be worked upon. Face recognition systems usually require complex model engineering and are resource intensive, and most efforts discussed, observed a fairly limited access of the system to the retailer themselves. Retailer's interaction with the interface is rather diminished and therefore, there is potential to integrate the concept of face recognition with a more user friendly, and modern interface framework. Lighter and more interactive applications can be designed using state-of- the-art facial recognition through quick and responsive API's - contributing to more well-rounded programs with vast benefits to retailers.

Chapter 3

PROBLEM DEFINITION

Customer satisfaction is of utmost importance to all sorts of businesses and local businesses lack the resources of modern customer analytics. Therefore, we propose to develop a customer analytics system using state-of-the-art facial recognition and verification to assist local businesses in creating detailed customer profiles, provide better customer-specific products and perform customer behavior analysis.

Chapter 4

METHODOLOGY

The goal of our project is to create an easy-to-implement face detection and recognition system for better customer analytics. Our proposed solution is to analyze in-store video footages and apply face verification and recognition techniques to identify new and frequent loyal customers, thus tracking their purchasing behavior, such as likes/dislikes and purchase patterns. Different statistical models can then be made using recorded customer information which will provide the owner the ability to visualize the customer needs in a better way and to optimize the store accordingly. Analysis will be shown on the dashboard which will allow the admin to have a better understanding of customer demands through various aspects. Analyses include:

- Individualized purchase recommendations based on purchase history
- Customer age and gender distribution
- Customer emotion distribution based on gender and time of visit
- Frequency of customers based on time of day

For our front-end display, the system uses a React-based Point-of-Sales (POS) dashboard consisting of the following features:

- Recording customer purchases with their detected faces through POS.
- Sales and customer analytics visualization.
- Creating, updating, and managing inventory and orders.

Tools and libraries required:

- React JS

- TensorFlow JS
- Good Enough Recommendations Engine (GER) Engine
- Node JS and Express
- MongoDB

Let us consider the following use case of a customer entering a store:

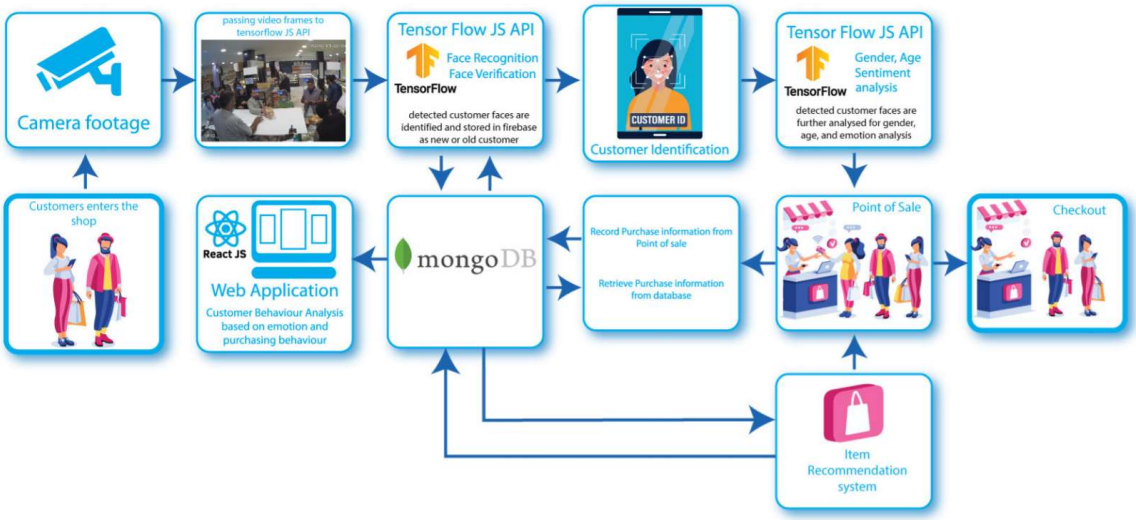


Figure 1. Methodology Overview

1. As soon as the customer enters the shop, they are captured using the set-up camera feeds. Frames are extracted from the video feed and passed onto Tensor Flow JS API, where the customer face is detected and assigned an ID.
2. The detected face is then crosschecked with already stored face descriptors to recognize if the customer is a new customer or an already existing one.
3. Furthermore, the detected face is processed for gender, age and sentiment detection using the same Tensor Flow JS API.
4. A React based POS dashboard is used to store purchase information with the customer's face and ID and the additions to the profile are stored in the database.
5. This stored data is then extracted and different analysis based on purchase history, gender and age are created applying statistical predictive algorithms to recommend purchases and provide customer and sales analytics.

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

The designed system comprises a web-based platform for retail owners to manage their POS transactions. The client-side enables POS transactions to be executed such as adding items to inventory, initiating an order, adjustment to an order, and bill generation. It also enables the user to view the analyzed data based on customers and sales information. A purchase recommendation mechanism has also been built which would recommend items to the customer. Moreover, the designed system uses Faceapi.js as well as Node and DB JSON server to communicate with the MongoDB database for storage of transaction details and other analytical data.

5.1.1 Architecture Design Approach

Our designed system follows a MVVM architecture pattern. Our database deployed on MongoDB Atlas acts as our model whereas our web platform which uses the React JS and ES6 frameworks are our views.

This design approach enables user testing and prototype demonstration to run parallel to the development lifecycle till the completion of system design. Furthermore, this approach can also help in the scalability of our applications on other platforms such as mobile.

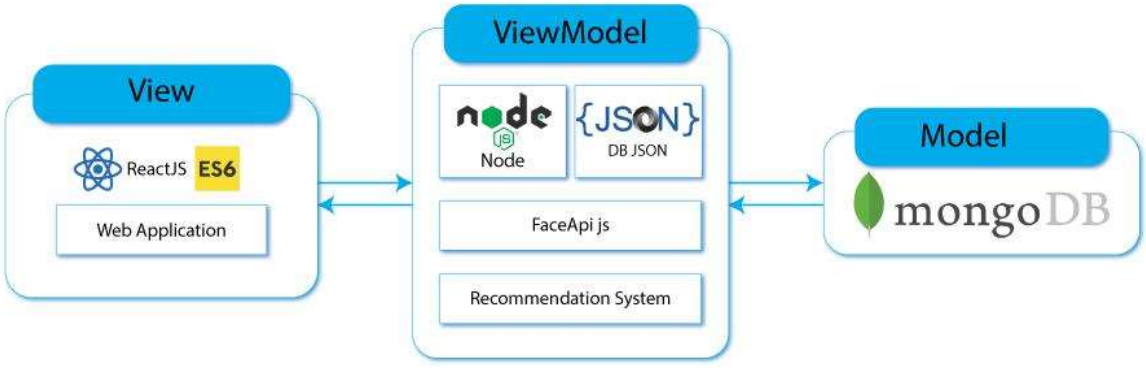


Figure 2. Architecture Design Approach

5.1.2 Architecture Design

The following diagram depicts the architecture design of the In-store customer analytics system. Partitioned into client, , server, and database, the dataflow in the system has been clearly shown with arrows. The main functions of the system are designed in modules such as Face detection and recognition, POS dashboard, Purchase recommendation, reports, and analytics as well as inventory management.

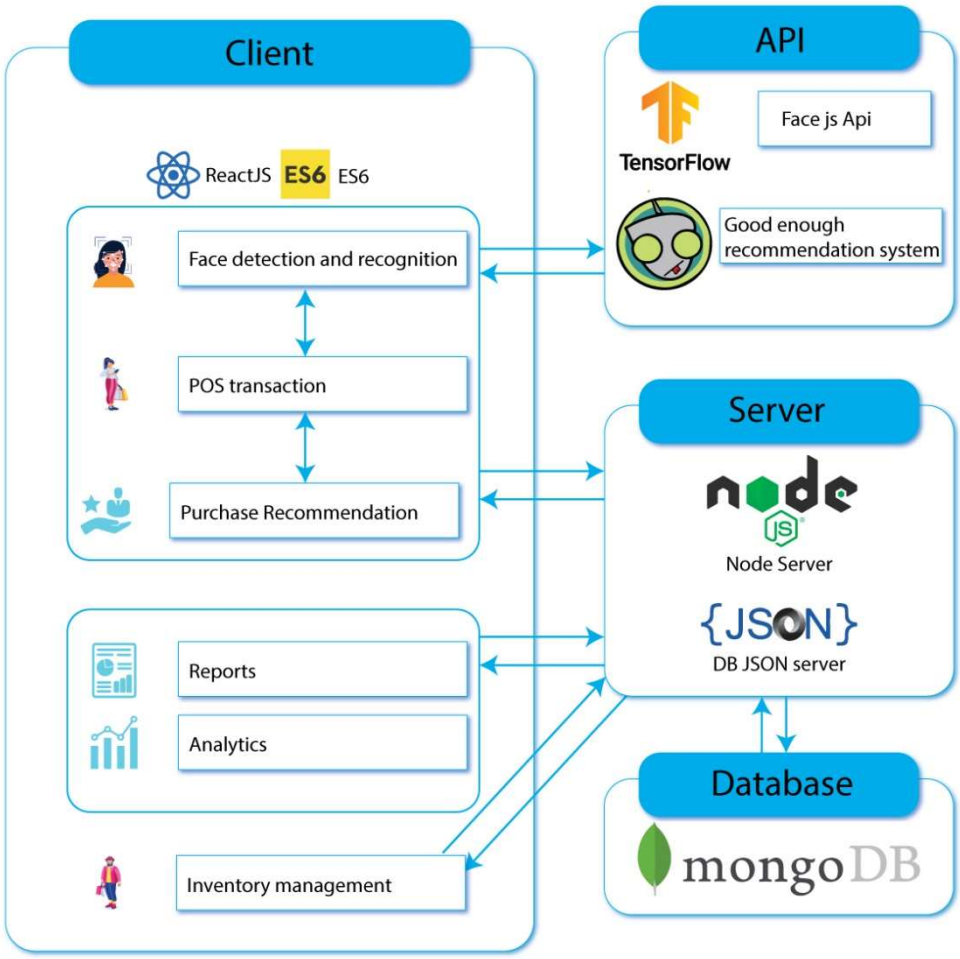


Figure 3. Architecture Design

5.1.3 Subsystem Architecture

The diagram below provides a comprehensive understanding of the flow of information and control all through the system at a generic (non-technical) level. All the major systems of the system have been labelled into their constituent parts.

Face detection and recognition module detects customer face, age, gender, and facial expression using Faceapi JS and compares the detected face descriptor with already stored descriptors in MongoDB database using node.js server. Upon a successful match, the customer's face is recognized and a POS transaction is initiated. The designed dashboard is equipped with all POS features required to execute sales such as barcode scanning, total bill generation, item adjustment, etc... Purchase recommendation system also recommends items to customers based upon likes/dislikes metrics as well as previous sales data of the customer. Sales are associated with each respective customer profile and upon checkout, POS transaction data are sent via node server to MongoDB database. Moreover, on another tab, reports and analytics based on sales and customers data such as sales reports, customer age, gender, and facial expression are displayed in graphs and bar charts. The inventory management system allows for the addition of new items, adjustments, and timely updates of stock.

APIs such as Faceapi JS and Good recommendation engine (GER) work asynchronously with other modules. Meanwhile, node and DBJSON server are responsible for fetching data to and from the MongoDB database.

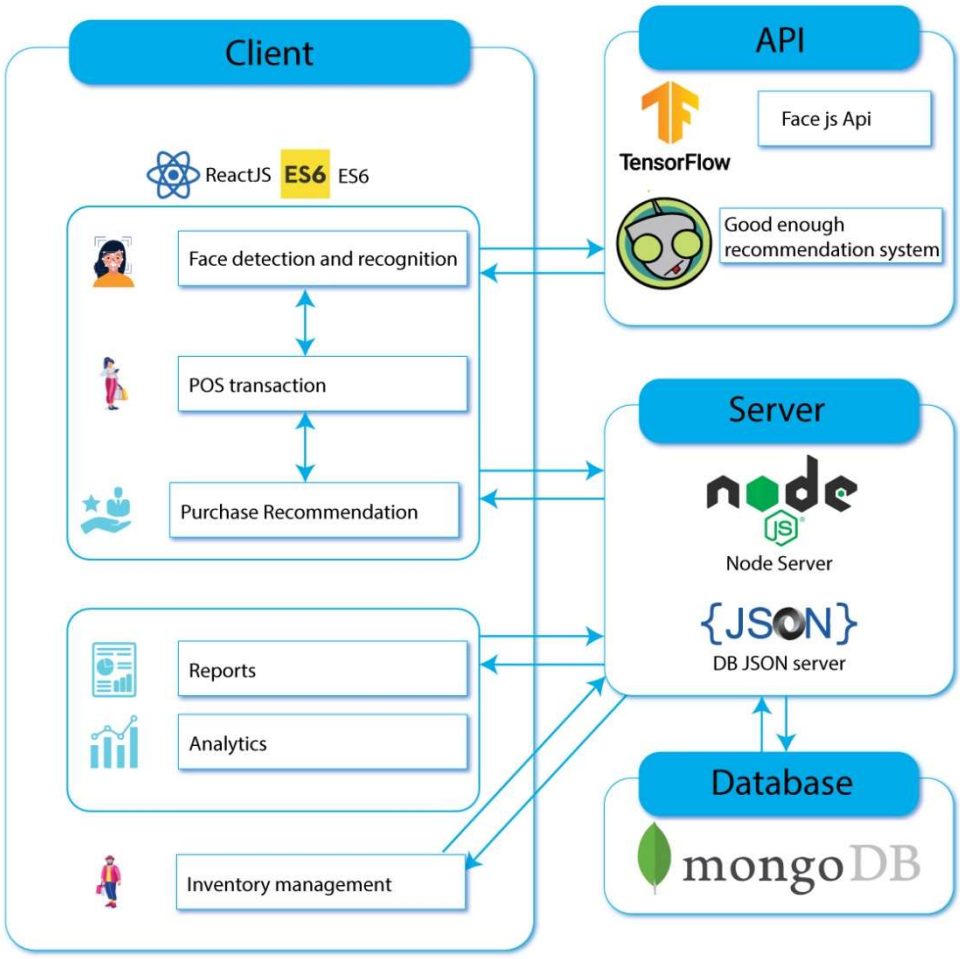


Figure 4. Subsystem Architecture

5.2 DETAILED SYSTEM DESIGN

The following section provides a comprehensive breakdown of each major module of the system along with its role and responsibility in the overall system's performance.

5.2.1 Client Application

5.2.1.1 Classification

This represents the web application built on React JS & ES6 for the user to interact with the system.

5.2.1.2 Definition

The purpose of this module is to provide a rich User Experience to the owners of retail stores and supermarkets to handle POS transactions as well as be able to view customer face detection and recognition on the interface. Moreover, they will be able to use the recommendation system to recommend items to customers as well as use the analytical graphs such as average customers per hour, customer gender distribution, facial expression, and sales reports to increase sales.

5.2.1.3 Responsibilities

It is responsible for the following functions:

- Customer face detection and recognition
- POS transaction execution.
- Purchase recommendation system
- Sales reports and customer analytics
- Inventory management

5.2.1.4 Constraints

- Customer face detection and recognition is hardware intensive and therefore with an increase in the number of customers faces being detected, the system might lag while processing both face recognition and POS transactions.
- POS transaction execution need real-time access to the database and it doesn't work offline
- Purchase recommendation system might give incorrect recommendation due to lack of customer data
- Item's information must be fed manually into the Inventory management system, making data entry a slow process.
- The assumption is that the internet connection will be available all the time because the back-end is all implemented on node and MongoDB and for the functioning of this module, the internet is vital. Lack of Internet connection or absence of web services would affect the working of this module.

5.2.1.5 Composition

The module comprises different layout files and java files with logic implementation, adapters that are working as views to the user.

5.2.1.6 Interactions

This module interacts with APIs such as Faceapi JS and Good Enough recommendation (GER). Moreover, this module is also connected with servers to interact with the MongoDB database. This module will run on any web browser platform.

5.2.1.7 Resources

This module will run on any web platform hence its support for interaction with humans would be any computer supporting a browser.

5.2.1.8 Processing

Following major processing are performed by this module:

- Face Descriptor extraction and comparing for facial recognition
- Age, gender, and facial expression extraction
- POS transaction execution such as bill generation.
- Purchase recommendation to the user
- Reports and analytical graph generation based on sales and customer data
- Inventory management such as addition, removal, and updates of item details.

5.2.1.9 Interface/Exports

Customer Face Detection and Recognition:

Upon customer face detection and recognition, his/her profile is loaded that includes age, number of visits, and total spending.

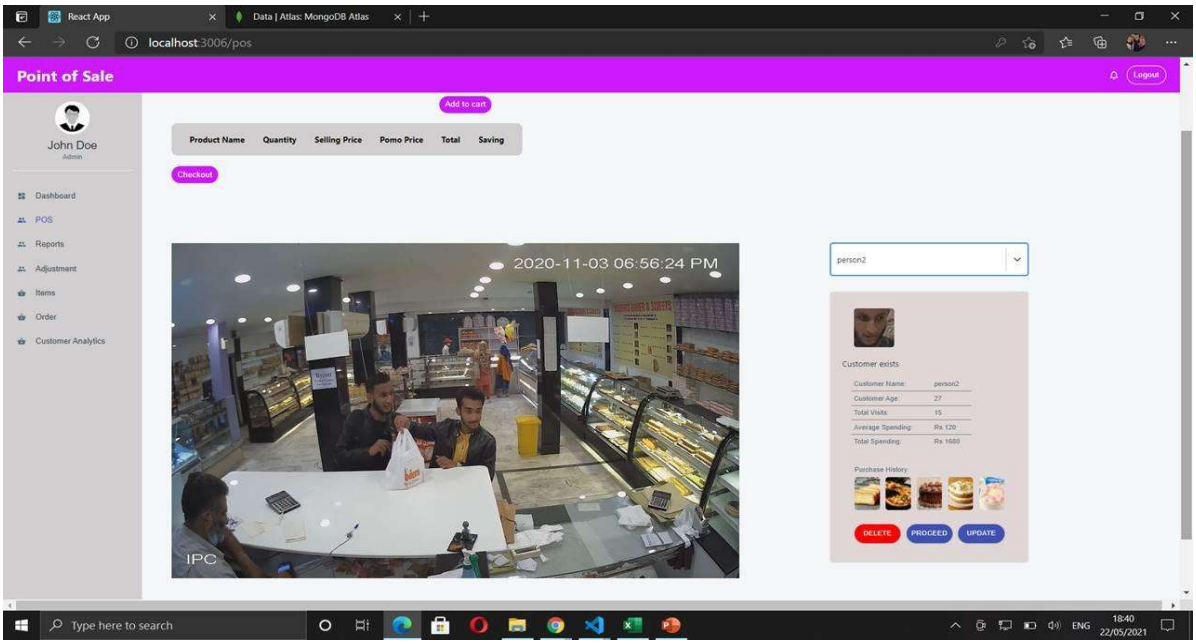


Figure 5. Customer Detection UI

Purchase recommendation:

Before initiating POS transaction, items are recommended to the customer for purchase based on previous sales and customer data.

Figure 6. Purchase Recommendation UI

Point of sale transaction:

Upon selection of POS, video footage of camera upon which face detection and recognition will occur as well as POS bill generation and customer profile are made visible. After customer selection, a POS transaction is initiated. Items are added to the order and a total bill is generated at the end.

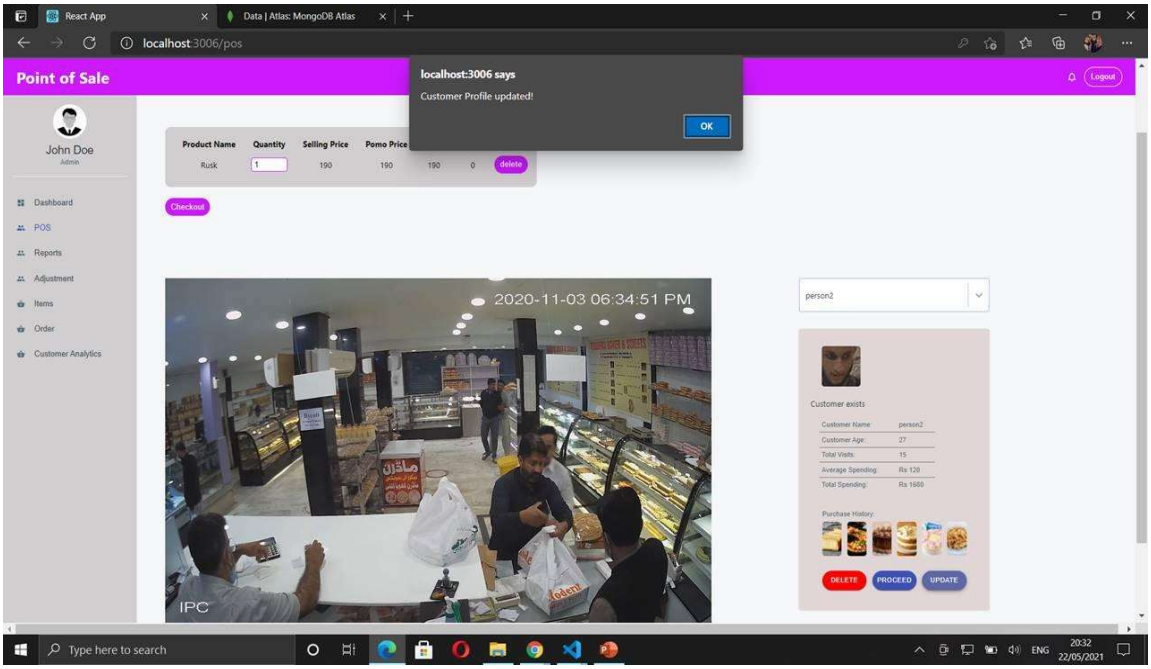


Figure 7. Updating Existing Customer UI

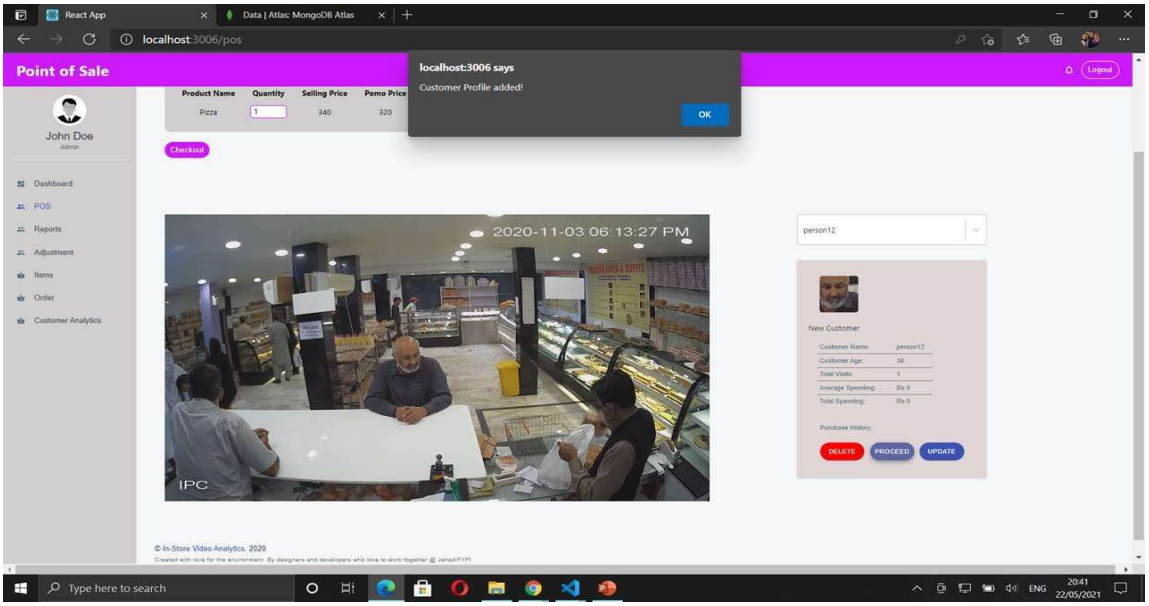


Figure 8. Adding new Customer UI

Sales reports:

Upon selection of Reports option on menu, Sales reports are shown in form of pie charts and bar charts. Moreover, sales reports are generated for any desired duration and monthly sales can also be analyzed accordingly.

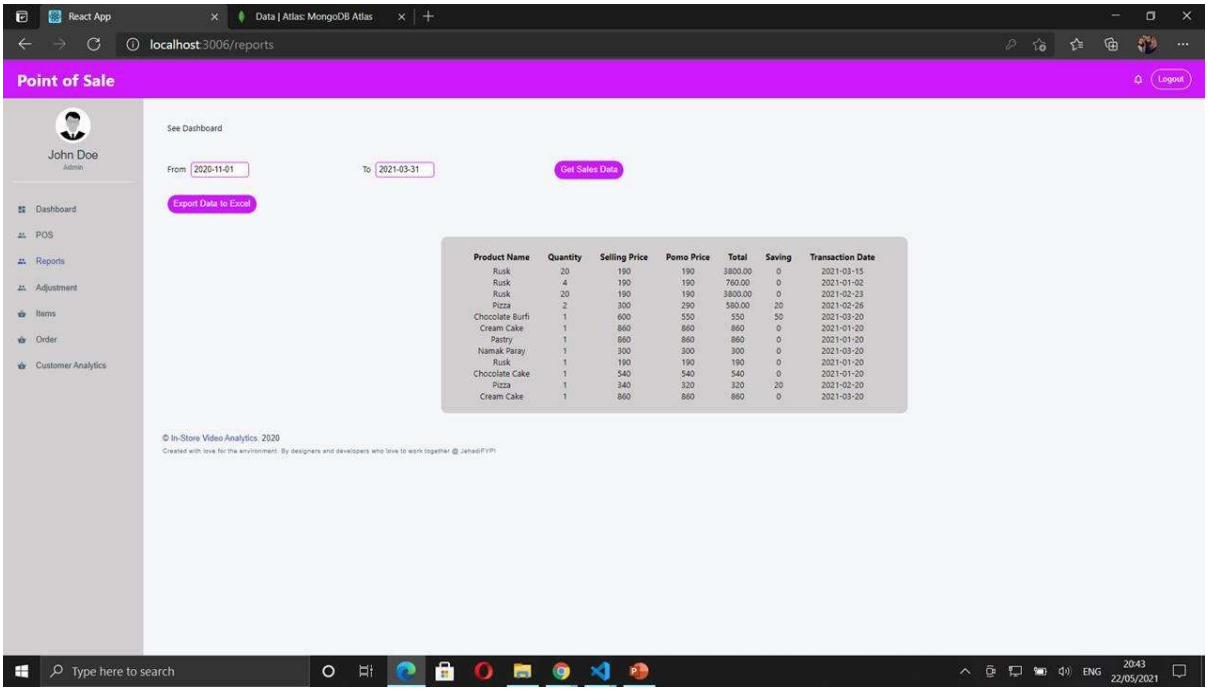


Figure 9. Sales Report UI

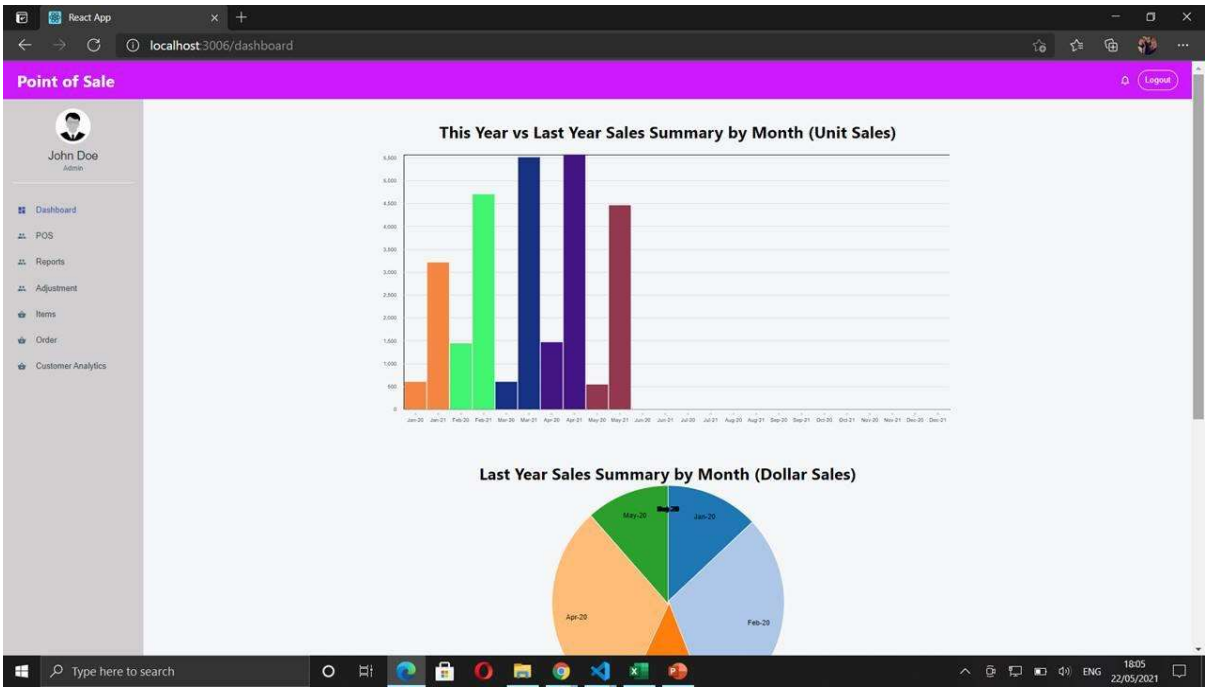


Figure 10. Sales Dashboard UI

Customer Analytics:

Upon selection of customer analytics option in tab menu, customer analytical data is shown in bar charts and pie charts, which are generated on accumulated customer data in MongoDB, such as age distribution, gender distribution, and facial expression of the customers.

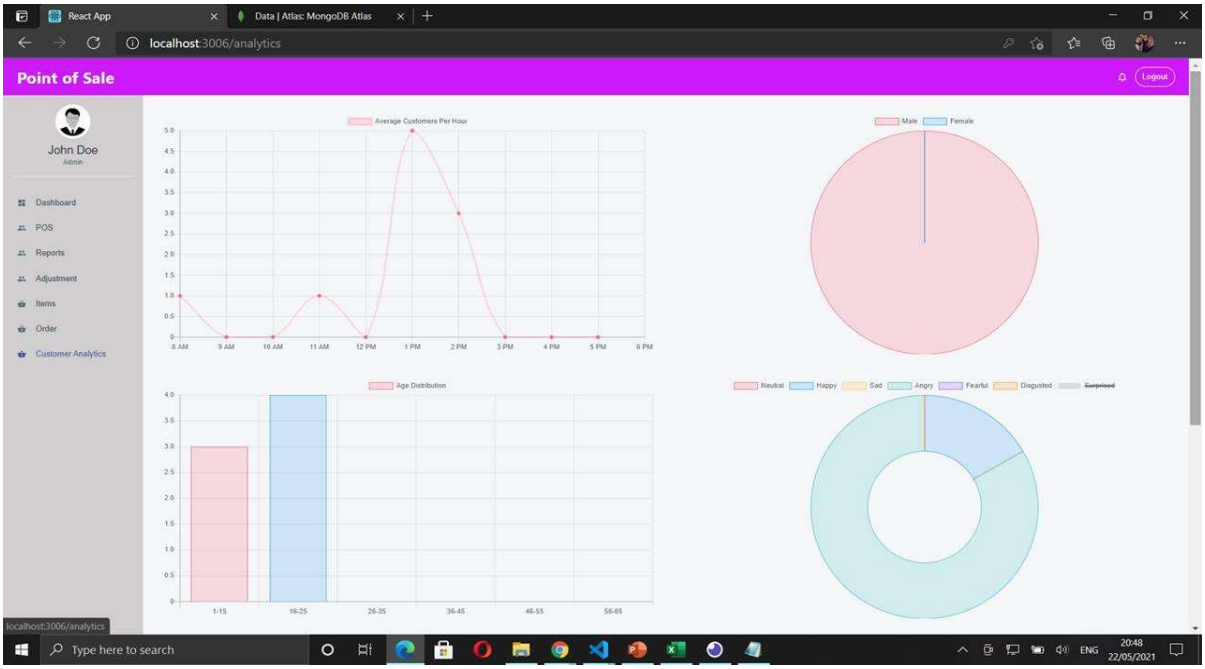


Figure 11. Customer Analytics I

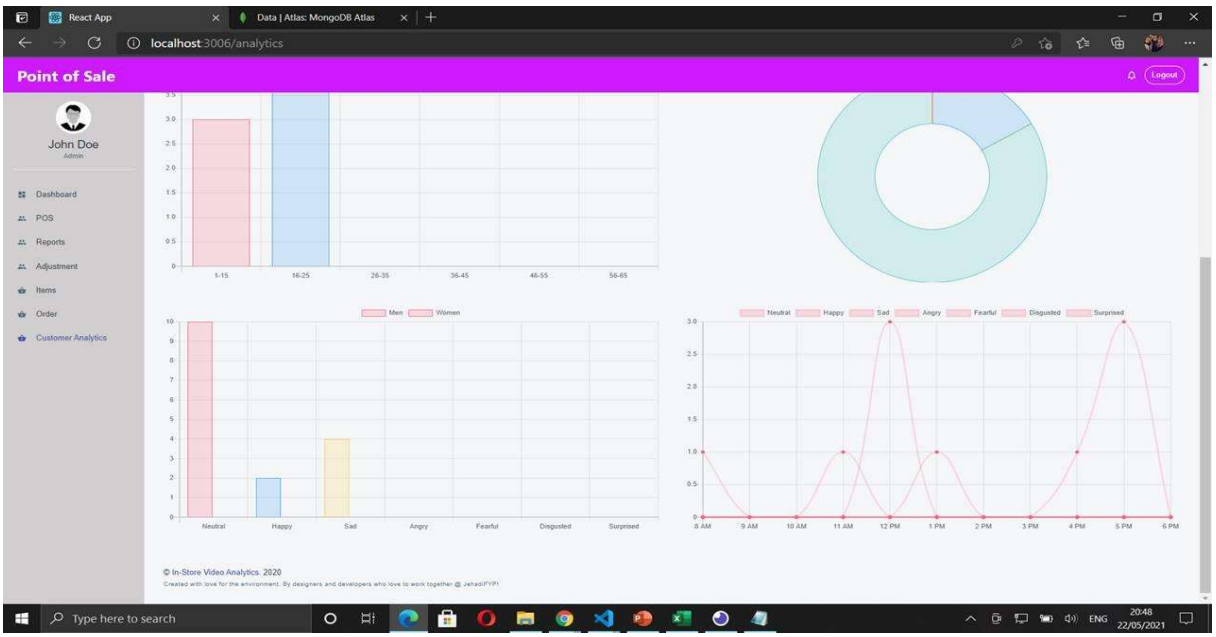


Figure 12. Customer Analytics II

Inventory management system

Upon selection of the Items option in the menu, a list of all items in inventory is shown. Item's information is editable and adjustments can be made to its name, barcode number, quantity, and cost.

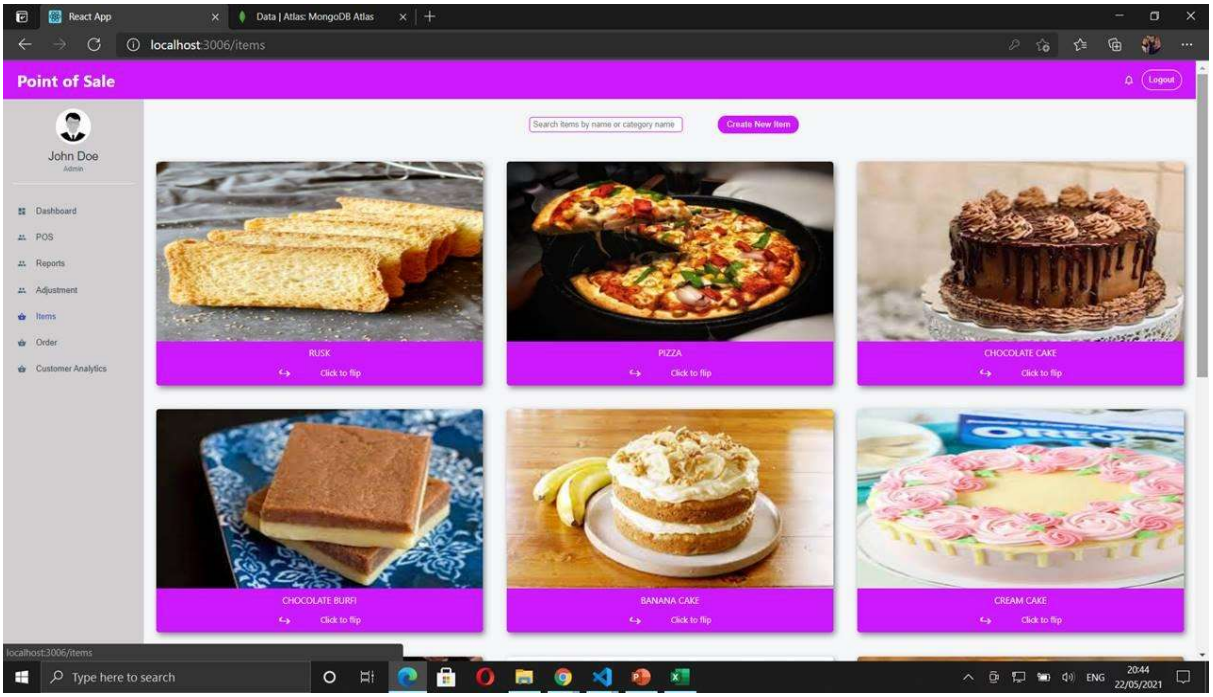


Figure 13. Item List UI

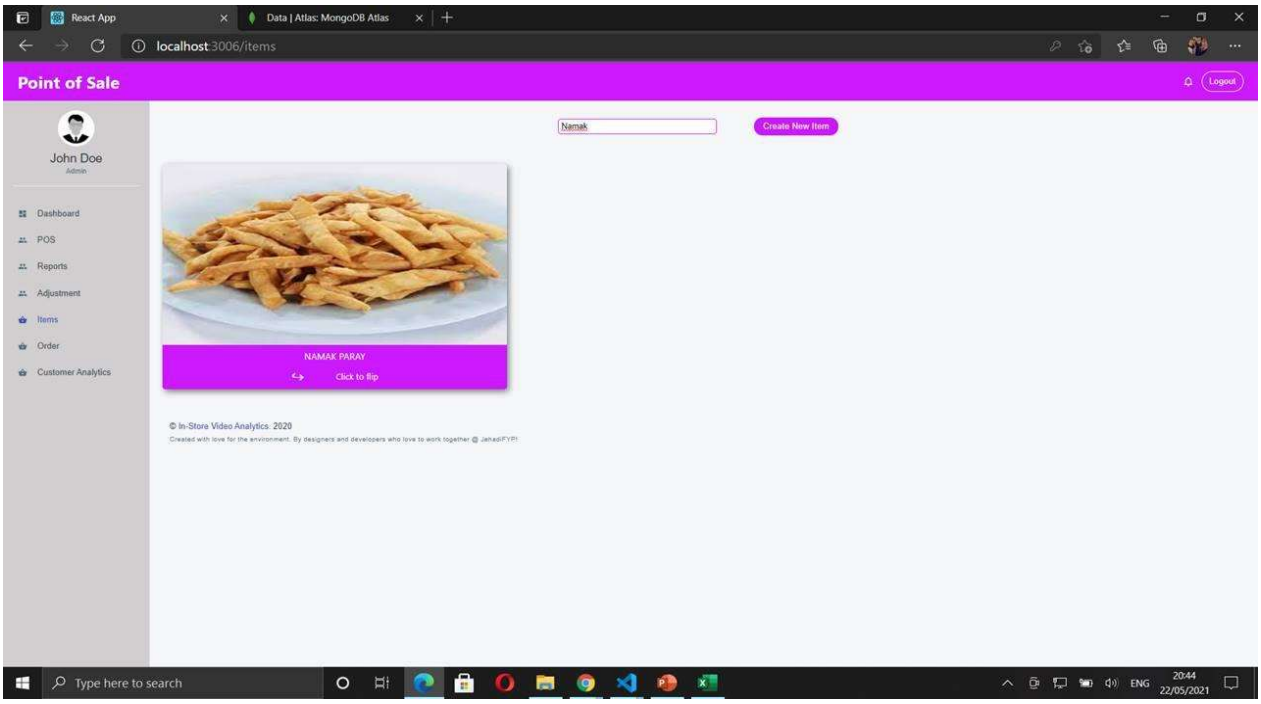


Figure 14. Edit Product

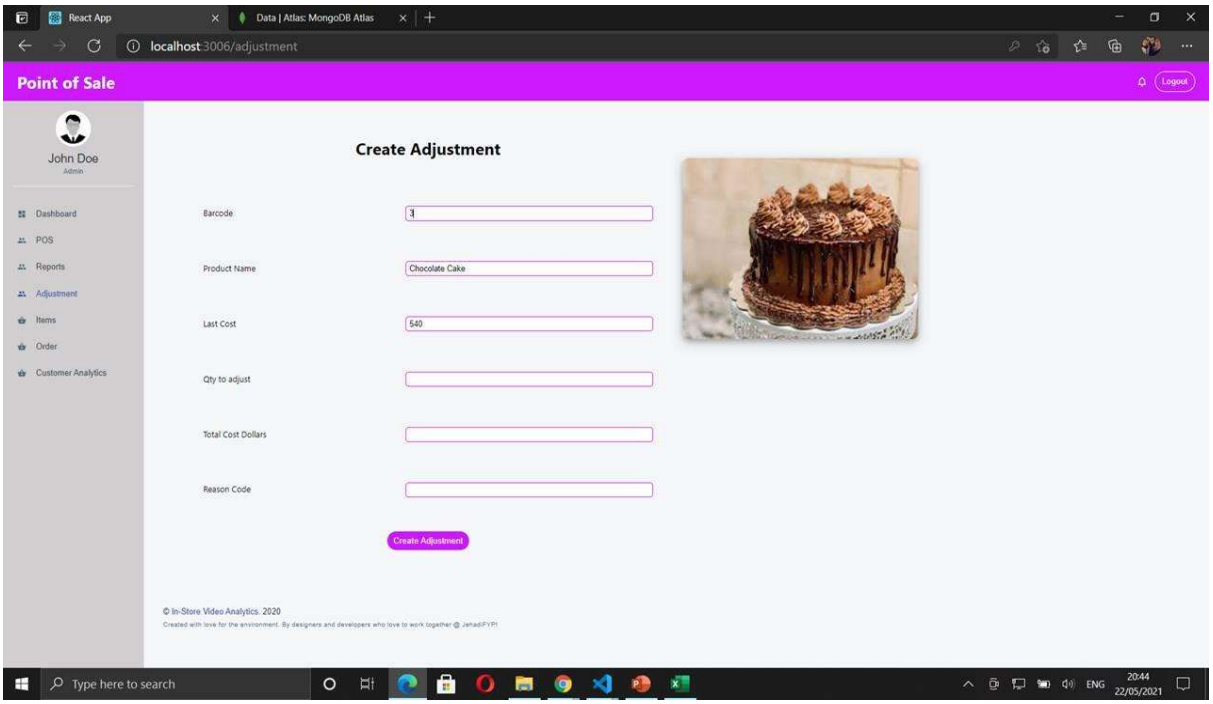


Figure 15. Create Adjustment UI

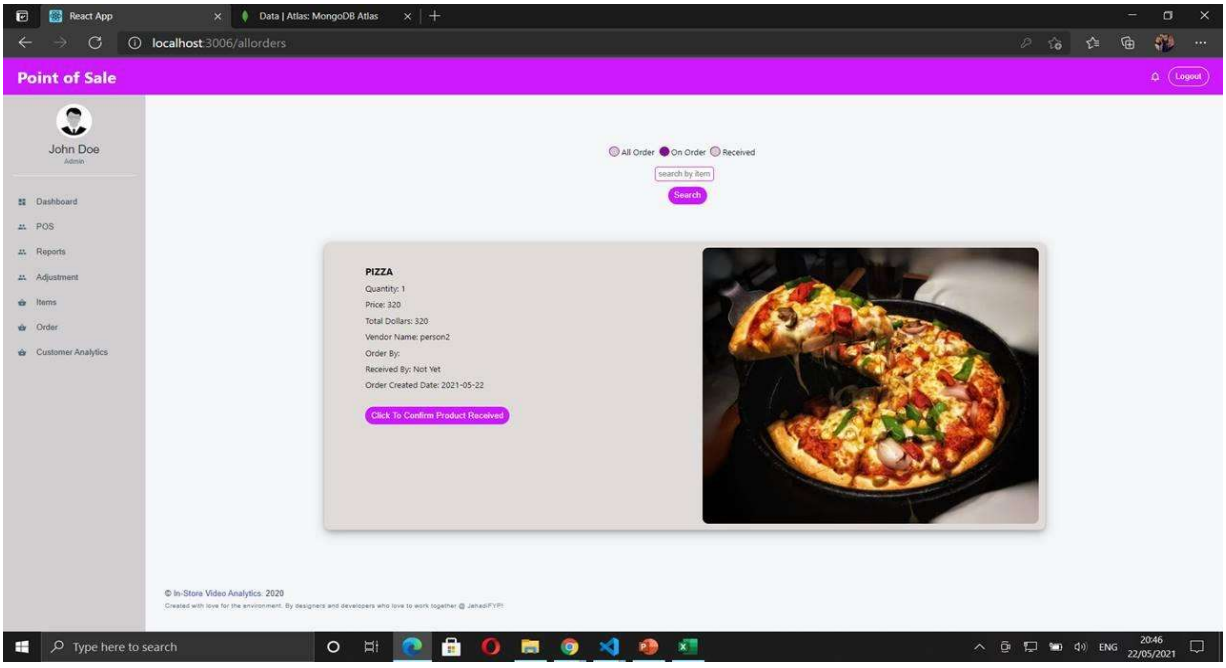


Figure 16. View Order UI

5.2.1.10 Detailed Subsystem Design

A detailed overview of this module can be seen in the following diagram:

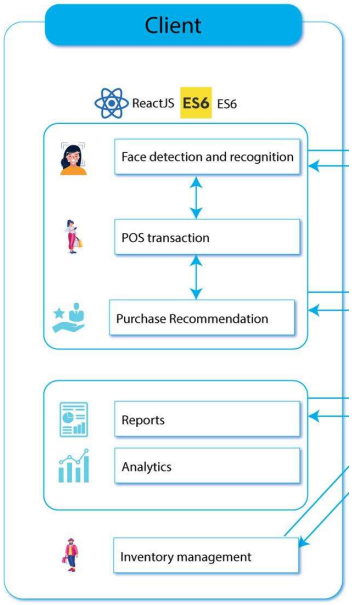


Figure 17. Client Application Design

The Client-side of this application comprises UI modules as well as logical controllers for displaying customer profile data, sales data, inventory data, sales report data, analytical data, and real-time facial detection and recognition. Facial descriptors extracted during facial detection and recognition are used to profile customers during the execution of POS transactions. Purchase recommendation system using GER API suggests items to the customer and upon successful execution of a transaction, sales reports, customer analytics, and inventory data are updated accordingly.

5.2.2 APIs

5.2.2.1 Classification

This represents the APIs such as Faceapi JS and GER (Good enough recommendation) API which are used by face detection and recognition as well as a purchase recommendation system for the customer.

5.2.2.2 Definition

The purpose of this module is to asynchronously respond to any calls made by the face

detection and recognition module or purchase recommendation module during POS transactions.

The Tiny Face Detector is a very performant, real-time face detector. This model is extremely mobile and web-friendly, and the size of the quantized model is only 190 KB (tiny_face_detector_model). The face detector has been trained on a custom dataset of ~14K images labelled with bounding boxes. Furthermore, the model has been trained to predict bounding boxes, which entirely cover facial feature points, thus in general it produces better results in combination with subsequent face landmark detection. For face recognition, a ResNet-34 like architecture is implemented to compute a face descriptor (a feature vector with 128 values) from any given face image, which is used to describe the characteristics of a person's face. we can determine the similarity of two arbitrary faces by comparing their face descriptors, for example by computing the Euclidean distance.

Good Enough Recommendations (GER) is an open-source recommendation engine that is scalable, easily usable, and easy to integrate. GER's goal is to generate good enough recommendations for any application or product so that it can provide value quickly and painlessly. GER is implemented in Coffee-Script on top of Node.js. The core logic is implemented in an abstraction called an Event Store Manager (ESM), which is the persistency where many calculations occur.

5.2.2.3 Responsibilities

Face-api.js:

- Detect faces
- Recognize faces
- Extract age, gender, facial expression

Good Enough Recommendations (GER):

- Recommend items to a user based upon likes/dislikes and previous sales data

5.2.2.4 Constraints

Face-api.js is a hardware-intensive API which does not perform well enough on CPUs with limited cores.

Moreover, Good Enough Recommendations (GER) recommendation accuracy is reduced with lesser customer likes/dislikes metrics.

The assumption is that internet connection will be available all the time because the back-end is all implemented on node.js and for the functioning of this module, the internet is vital. Lack of Internet connection or absence of MongoDB services would affect the working of this module.

5.2.2.5 Composition

This module majorly comprises face-api.js as well as Good Enough Recommendations (GER). Face-api.js is a JavaScript module, built on top of tensorflow.js core, which implements several CNNs (Convolutional Neural Networks) to solve face detection, face recognition, and face landmark detection, optimized for the web and for mobile devices. Meanwhile, Good Enough Recommendations (GER) is implemented in Coffee-Script on top of Node.js. The core logic is implemented in an abstraction called an Event Store Manager (ESM), which is the persistency where many calculations occur.

5.2.2.6 Interactions

This module only interacts with the client-side module for real-time face detection and recognition as well as customer purchase recommendation during POS transactions.

5.2.2.7 Resources

This module will run on any web platform hence its support for interaction with humans would be any computer supporting a browser.

5.2.2.8 Processing

Following major processing are performed by this module:

- Face Descriptor extraction and comparing for facial recognition
- Age, gender, and facial expression extraction
- Purchase recommendation to the user upon likes and dislike metrics

5.2.2.9 Interface/Exports

The output of the stated APIs are shown in client modules and they do not have an interface of their own for the user to interact with.

5.2.2.10 Detailed Subsystem Design

A detailed overview of this module is shown in the following figure:

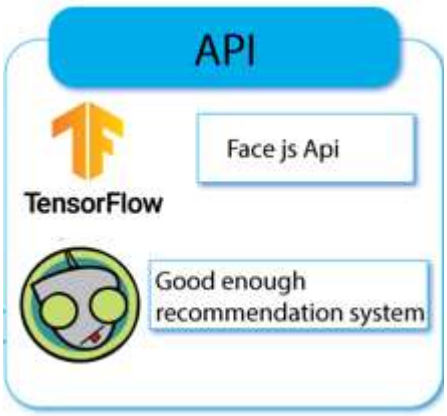


Figure 18. API structure

5.2.3 Server

5.2.3.1 Classification

This module represents the server module of our system for which 2 servers are deployed; node.js for fetching data to and from the MongoDB database to interact with client modules and DBJSON server to keep the items inventory updated.

5.2.3.2 Definition

The purpose of this module is to manage the queries to the MongoDB database and items JSON data file since most of the other system components such as face detection and recognition, POS transactions, GER, Sales reports, and customer analytics as well as inventory management system fetch data to and from the MongoDB database asynchronously. Therefore, in order to manage the queries load, we have used server's module to balance the load as well as monitor access to the database.

5.2.3.3 Responsibilities

It is responsible for the following functions:

- Manage exchange of data between database and user.
- Keep all system components updated so that the views are updated in time.
- Prevent unwanted access to the database.

5.2.3.4 Constraints

The assumption is that internet connection will be available all the time because the back-end is all implemented on node.js and for the functioning of this module, the internet is vital. Lack of Internet connection or absence of MongoDB would affect the working of this module.

5.2.3.5 Composition

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command-line tools and for server-side scripting—

running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

5.2.3.6 Interactions

This module interacts with all other modules because this is the module, where all the data passes through and it is necessary for every other module to interact with this module to perform CRUD operations to update the database.

5.2.3.7 Resources

This module is a back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

5.2.3.8 Processing

CRUD operations on the data.

5.2.3.9 Interface/Exports

No interface.

5.2.3.10 Detailed Subsystem Design

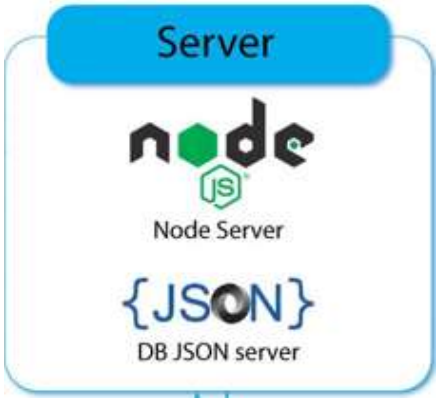


Figure 19. Server Design

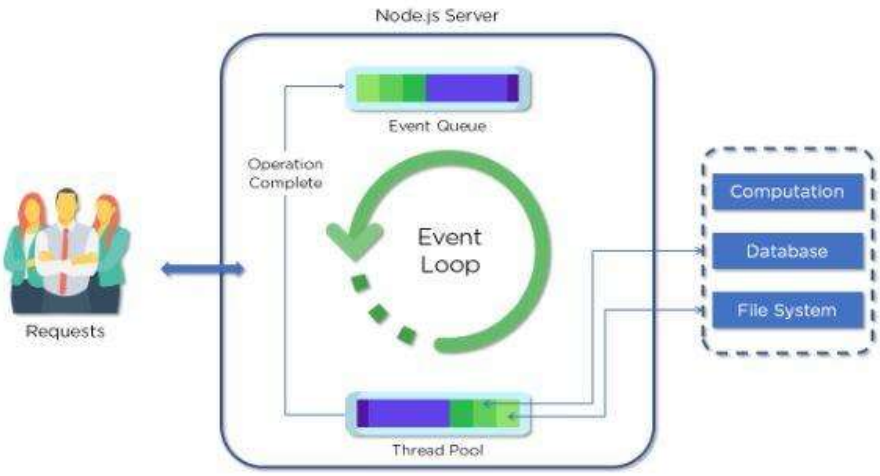


Figure 20. Node JS server abstraction

5.2.4 MongoDB Database

5.2.4.1 Classification

This module represents the cloud database service for Real-time database to store customer analytics and sales data accordingly

5.2.4.2 Definition

The purpose of this module is to provide one central Real-time NoSQL Database for our system. We have used NoSQL Database as there is less relation in our data and we want our system to be fast. The module's main purpose is to avoid data redundancy.

5.2.4.3 Responsibilities

It is responsible for the following functions:

- To catch the updates to data.
- Store and update customer profile data as well as Sales data.
- To update all the devices when there is an update to the database.
- Provide security rules to enable the security of user's important data.

5.2.4.4 Constraints

The assumption is that the internet connection will be available all the time because the

back-end is all implemented on node.js and for the functioning of this module, the internet is vital. Lack of Internet connection or absence of MongoDB services would affect the working of this module.

5.2.4.5 Composition

This module works on the principle of the MVVM architecture of our system. Our database acts as the model of our system for which the server updates the view whenever a change in the state of the database is noticed. It provides a NoSQL database with security rules.

5.2.4.6 Interactions

This module interacts with all other modules through the server because this is the module, where all the data resides and it is necessary for every other module to interact with this module to perform CRUD operations to update the data.

5.2.4.7 Resources

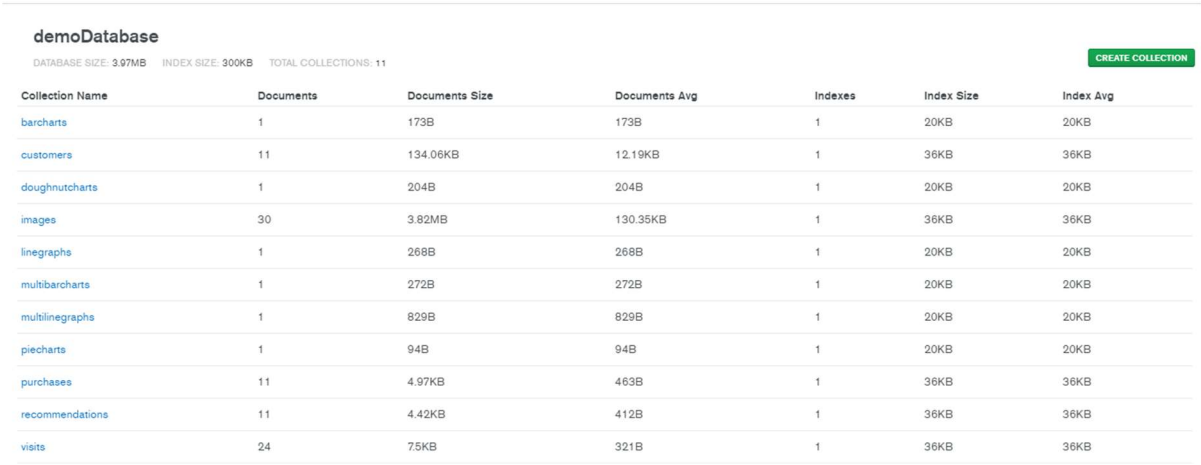
This module is cloud-based and MongoDB atlas managing it.

5.2.4.8 Processing

CRUD operations on the data.

5.2.4.9 Interface/Exports

Online Interface to interact with the NoSQL database of our system.



demoDatabase

DATABASE SIZE: 3.97MB INDEX SIZE: 300KB TOTAL COLLECTIONS: 11

CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
barcharts	1	173B	173B	1	20KB	20KB
customers	11	134.06KB	12.19KB	1	36KB	36KB
doughnutcharts	1	204B	204B	1	20KB	20KB
images	30	3.82MB	130.35KB	1	36KB	36KB
linegraphs	1	268B	268B	1	20KB	20KB
multibarcharts	1	272B	272B	1	20KB	20KB
multilinegraphs	1	829B	829B	1	20KB	20KB
piecharts	1	94B	94B	1	20KB	20KB
purchases	11	4.97KB	463B	1	36KB	36KB
recommendations	11	4.42KB	412B	1	36KB	36KB
visits	24	7.5KB	321B	1	36KB	36KB

Figure 21. Mongo Atlas Interface

5.2.4.10Detailed Subsystem Design

The following diagram shows the connection of the database module with the server. All queries to the database are made through the server.

Figure 22. Database Design

5.3 COMPONENT DIAGRAM

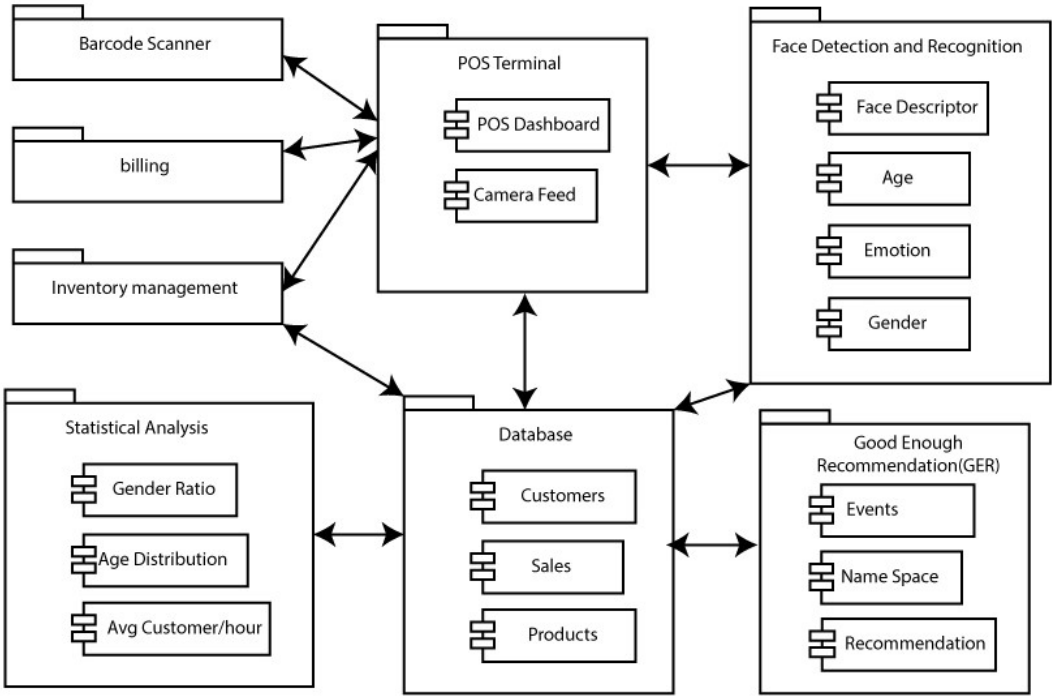


Figure 23. Component Diagram

5.4 ER DIAGRAM

(Our system uses a non-relational database that is very responsive to fast queries, which is why there is no ER diagram of our database.)

Chapter 6

IMPLEMENTATION AND TESTING

The application has been implemented and tested in a bakery store by the name of "Modern Bakers and Sweets" situated in Aabpara, Islamabad. All the collaborations were made with this group for the requirement gathering and elicitation process. Surveillance video feeds were gathered and after pre-processing, facial recognition and detection techniques were applied to generate customer analytics. Similarly for product analytics, a POS system was implemented on-site. Finally, the POS system and a facial recognition model were integrated and a user-friendly dashboard was created to be used by the end-users. Each and every feature of the application has been tested in the runtime environment and all the components are fully functional.

Some of the design and implementation constraints are,

- The accuracy of the face detection and recognition model may be compromised if the number of faces to be detected and recognized exceed 4, the system gives 100% accuracy when there is only one person at the counter, which is an ideal condition and is not possible in a busy environment.
- Since the system is running on a browser, it may lag in case the system comprises less than 8 GB RAM.
- This web-based system has a disabled network firewall to speed up internal operations. Enabling the firewall would surely prevent attacks but it would slow down the server process.
- Customers wearing masks and face coverings might not be detected and recognized and the system might register old customers wearing masks as new customers.
- It is recommended to have a proper queuing system in place physically in stores, so that customers and their respective purchases can be recorded in an orderly and efficient manner.

The system has been tested for conformance and in accordance with the initially laid down requirements and specifications however certain implementation changes have been made after thorough discussion with and feedback from potential clients. The system has been tested on a small set of repeating customers at the aforementioned bakery store to ensure all core functionalities of the system are met.

Chapter 7

RESULTS AND DISCUSSION

The web application works to its absolute optimum provided the recommended system configurations are met and a stable internet connection is set up. The system has been optimized to work to its full potential. After keen discussion with potential stakeholders, the system has been modelled to cover all key requirements of the stakeholders that were previously lagging in the systems that are currently being used. The system has been discussed in detail and devised in accordance, with the needs and expectations of the stakeholders.

Table 1. Application Configuration Specifications

Application Configuration	***Specs***
RAM	8 GB
Input size (video frame size)	960
Score Threshold (for detecting a face)	0.4
Max Threshold (best match for recognition)	0.5

The successful implementation of our system on the above-mentioned configuration specs allowed us to detect and recognize at most 4 customers at a single instance in the sample video feeds. Considering a queueing system is implemented at the given store, this figure more than allows for successful detection and recognition of customers.

The successful extraction of customer and purchase information allowed us to generate customer profiles, based on the repeating customers' test set, which were then used to generate important business statistics and customer recommendations. These statistics allowed the business owner to revisit certain overlooked aspects such as the time of day and the customer mood - hinting the performance of the customer service at certain shifts.

The system has the potential to be launched into the product market as it was implemented in full accordance with the needs and requirements of the current bakery store and tends to introduce an entirely new technique to generate customer and product analysis.

Chapter 8

CONCLUSION AND FUTURE WORK

The above system revolutionizes POS technology in the local community. It equips local store owners, vendors, and retailers with the latest trends and tools being implemented in the business sphere, worldwide. The designed system was tested in an actual working environment at a bakery store in Islamabad, and the owners found the performance of the system to be quite impressive. This shows that the system with a few tweaks can be easily deployed in the market sphere.

The system is feasible - not requiring any specific system requirements or any expensive hardware. The application proves to be easily scalable as components such as sales analytics and inventory management can be extended to mobile platforms as well for remote accessibility. The application uses Mongo Atlas as a database which is a DBaaS (Database as a Service) - a cloud computing management service - allows clients to expand or shift their businesses, if need be, without any physical hassle.

Our designed system achieves our goal of introducing a customer analytics system, completely new to the local business community, and hopes to vastly improve customer service standards in the country. The system aids business owners by allowing them to personalize with their customers and provide a better insight into their customers' choices - thus improving customer satisfaction levels.

REFERENCES

[1] J. Lu, E. Liong, X. Zhou and J. Zhou, "Learning Compact Binary Face Descriptor for Face Recognition," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 37, no. 10, pp. 2041-2056, October 2015.

[2] H. Rai, K. Jonna and P. R. Krishna, "Video analytics solution for tracking customer locations in retail shopping malls," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, San Diego, 2011.

[3] W. Applebaum, "Studying Customer Behavior in Retail Stores," *Stop & Shop, Inc.,* vol. 16, no. 2, pp. 172-178, 1 October 1951.

[4] A. Generosi, S. Ceccacci and M. Mengoni, "A deep learning-based system to track and analyze customer behavior in retail store," in *Conference: 2018 IEEE 8th International Conference on Consumer Electronics* , Berlin, 2018.

[5] N. T. Karim, S. Jain, J. Moonrinta, M. N. Dailey and M. Ekpanyapong, "Customer and target individual face analysis for retail analytics," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Chiang Mai, 7-9 Jan. 2018.

[6] Y. Song, Y. Xue, C. Li, X. Zhao, S. Liu, X. Zhuo, K. Zhang, B. Yan, X. Ning, Y. Wang and X. Feng, "Online Cost Efficient Customer Recognition System for Retail Analytics," in *2017 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Santa Rosa, CA, 2017.

[7] M. Coşkun, A. Uçar, Ö. Yildirim and Y. Demir, "Face recognition based on convolutional neural network," in *International Conference on Modern Electrical and Energy Systems (MEES)*, Kremenchuk, 2017.

[8] V. Borges, R. P. Duarte and C. A. Cunha, "Are you Lost? Using Facial Recognition to Detect Customer Emotions in Retail Stores," in *Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2019)*, Valencia, 2019.

[9] K. Maheshwar and N. N, "Facial Recognition Enabled Smart Door Using Microsoft Face API," *International Journal of Engineering Trends and Applications (IJETA),* vol. 4, no. 3, May - June 2017.

[10] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauau, HI, 2001.

[11] A. Senior, L. Brown, A. Hampapur, C.-F.Shu, Y. Zhai, R. Feris, Y.-L. Tian, S.Borger and C.Carlson, "Video analytics for retail," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007.*, 2007.

[12] F. OpenSource, "React," [Online]. Available: https://reactjs.org/.

[13] V. Mühler, "face-api.js," Github, 2018.

[14] N. T. Karim, S. Jain, J. Moonrinta, M. N. Dailey and M. Ekpanyapong, "Customer and target individual face analysis for retail analytics," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Chiang Mai, 2018.