Decision Support System for Retail

Software Design Description

Prepared by
Abdullah
Muhammad Uzair
Muhammad Umar Bin Ejaz

20 December 2020

Table of Contents

Introduction:

Software Design Description document is a detailed document for any software product and project. It includes all the granular details for the software and the system.

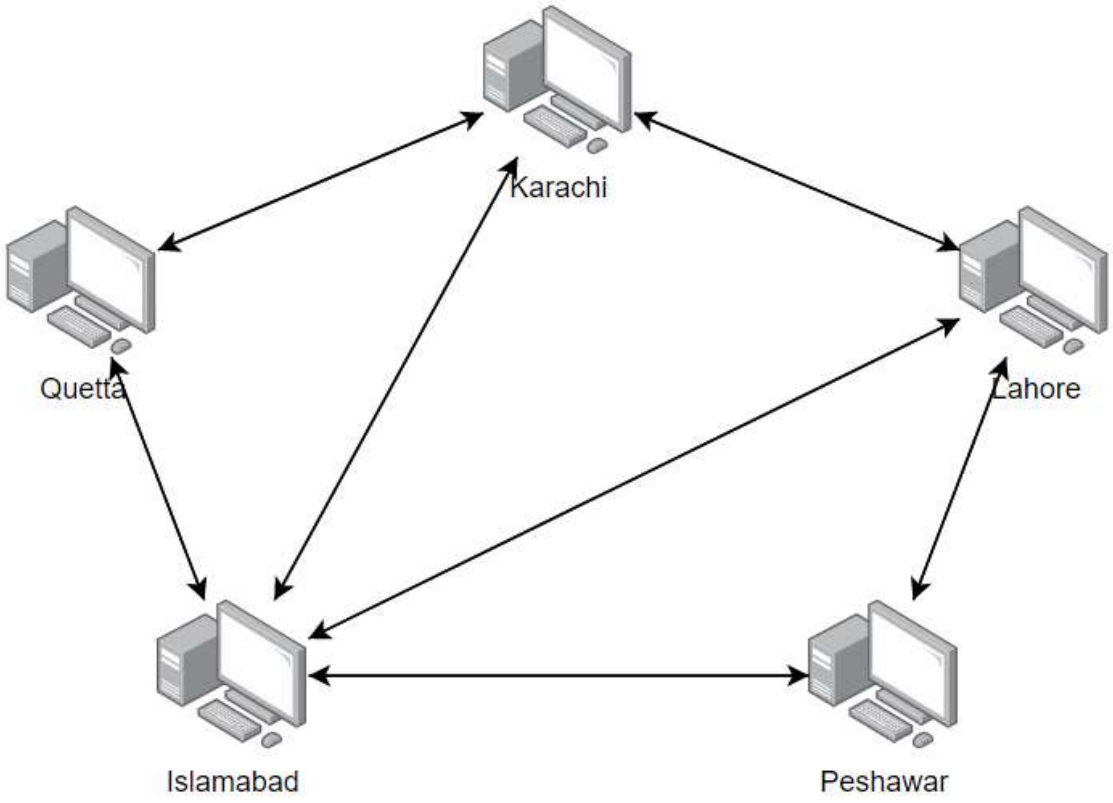
Purpose:

The purpose of this document is to understand different terms that will help developers and testers in developing the system. This document will provide all the details about the software that developers want to implement. It will explain the architectural viewpoints of the system in detail.

Scope:

Software Detailed Designed document is for the detailed description of the complex engineering problem that is decision support system for retail. This document will be helpful in explanation and understanding of the architecture of the system to the system developers and it is made by the system architect. It will explain the different technologies that will be implemented in smart retail.

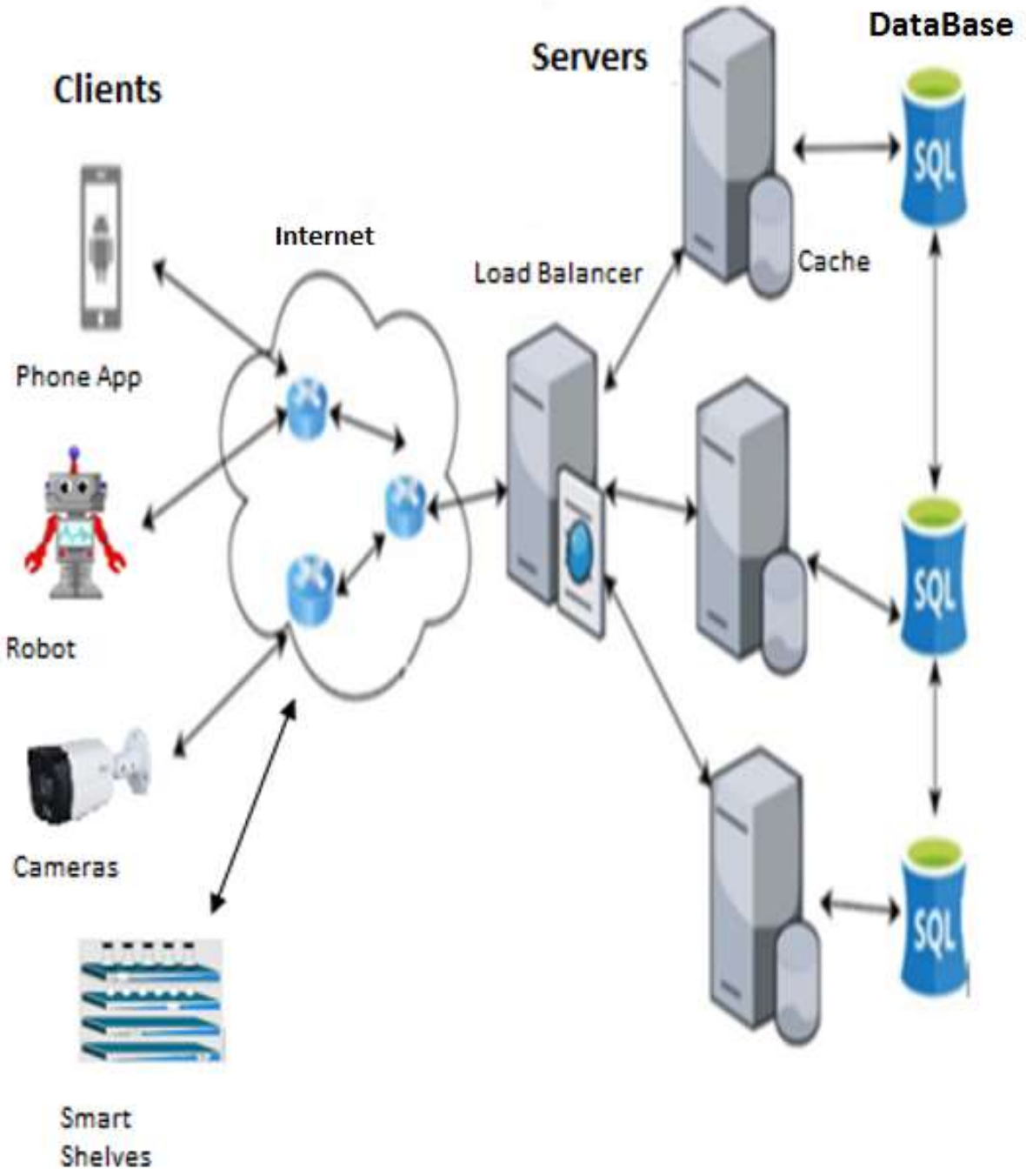
Overall Architecture



From a top-down view, the overall application is implemented using a peer-to-peer architecture. The headquarters at Islamabad acts as the super peer i.e., it is connected with all of the peers and assists in connecting all of the other peers to each other. Each geographical location is completely independent and only contacts the other locations for information such as stock and trends and for accessing the databases. The databases of other locations can be accessed remotely, this can help certain customers when a product is out of stock in one location, they can request the product to be purchased and shipped to them from another location.

With each geographical location being independently operated and with such low reliance on a centralized server it is easier to maintain each area and each area can have their own teams for maintenance.

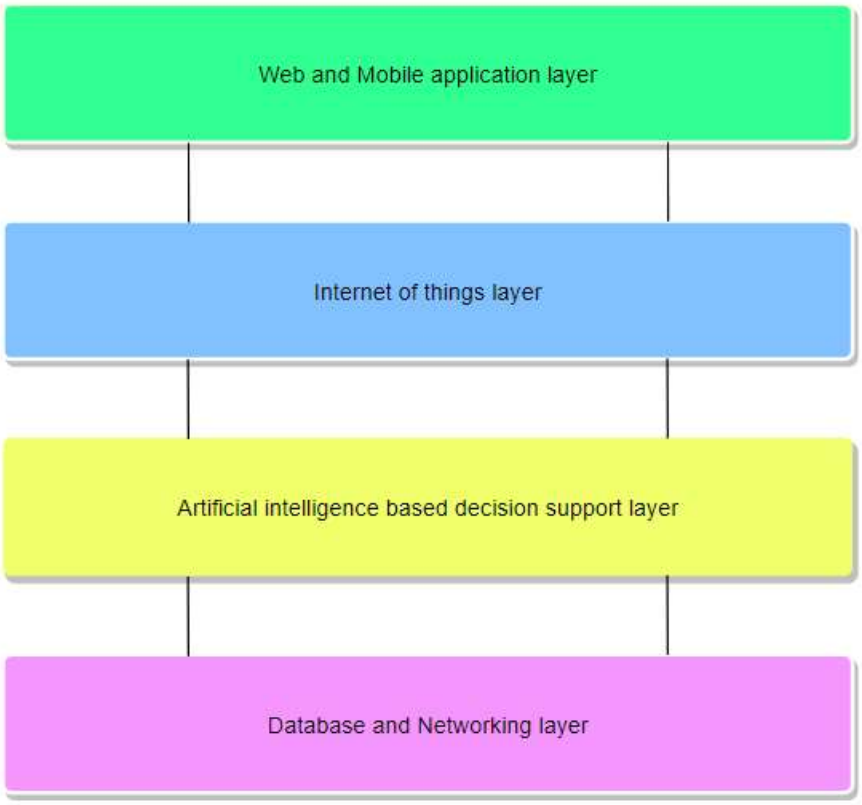
Deployment Diagram



Architecture of Peers (sub systems)

At every peer described in overall system architecture there will be independent sub systems deployed on each peer. These sub systems will use layered architecture in order to reduce

complexity and to take the advantage of modularity. The layers in layered architecture of these sub systems will include



1. **Web and Mobile application layer**
   In this layer only that part of system will be covered where users will interact with the system. Users are customers as well as staff members.
2. **Internet of things layer**
   In this layer the internet of things part of system is implemented. In this layer we have sensors, video cameras, actuators, smart shelves etc.

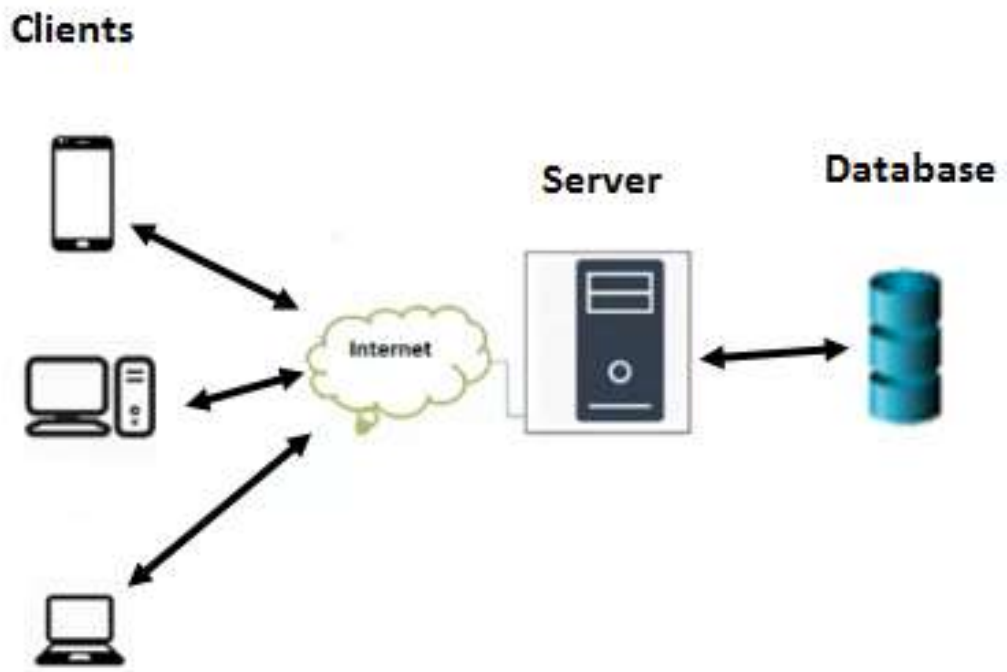
3. **Artificial intelligence based decision support layer**
   In this layer the processing part takes place. Decisions are made with the help of data gathered by internet of things layer and stored in database layer

4. **Database and Networking layer**
   In this layer all networking and data storage activities are performed. In this layer we cover actual deployment of physical resources of the system

Presentation Layer (Web and Mobile application layer):

In it the client will send request to the server and server will look that data in the database and then it will send response to the client. Client-server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection. Client-server architecture is also known as a networking computing model or client-server network because all the requests and services are delivered over a network.

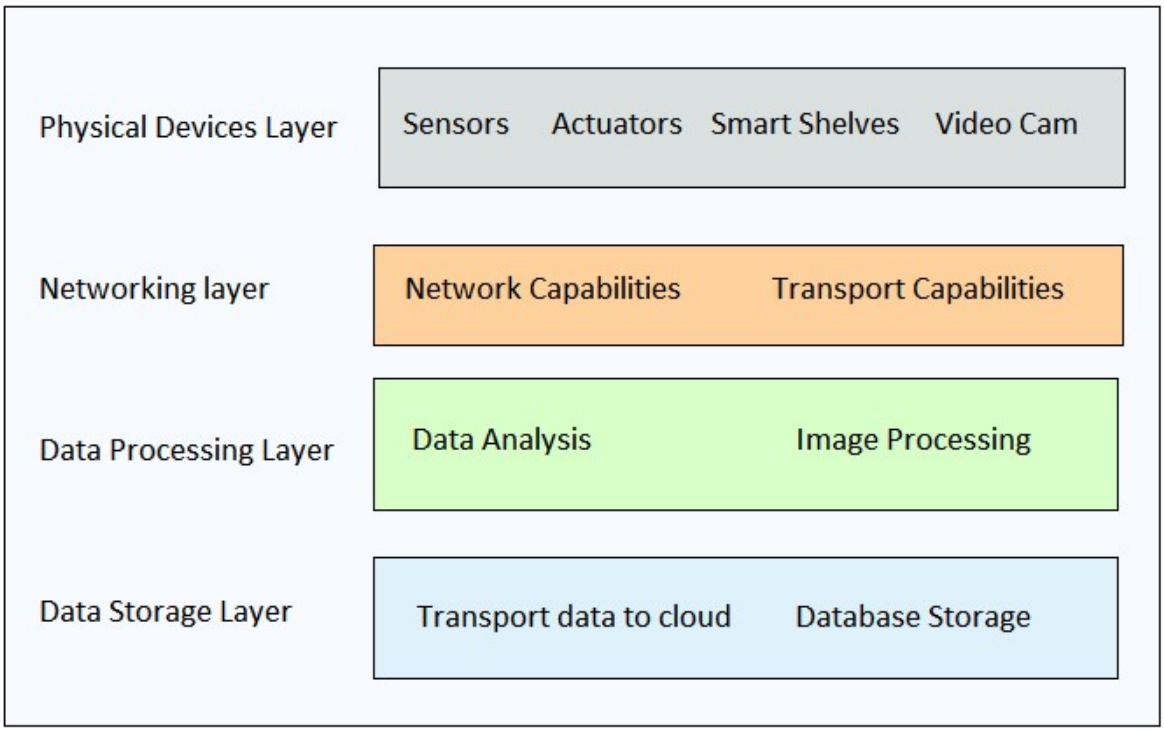
The clients here are of 2 types. The ones are the customers who have app installed on there mobile and will get all the updates using smart mobile app. The second type of clients are staff members who manage all the things and will look what is going on in retail.



Implementation of Internet of things layer

At higher level of abstraction, the internet of things is a layer but the inside of Internet of things layer because layered architecture decomposes the system into smaller

entities. reduces complexity and make it easier to debug system so it is very useful in IoT based solutions so the implementation of IoT layer is again based on layered architecture.
We have the following layers in our sub system



1) **Physical devices layer**
   In physical devices layer we have all the sensors, video cameras, actuators, smart shelves. This layer concerns with installation of these devices.

2) **Networking layer**
   In networking layer, we deal with network between the physical devices and the server where computation is being performed

3) **Data processing layer**
   In data processing layer we perform different operations on the data gathered by physical devices to make them understandable by other layers e.g., image processing.

4) **Data storage layer**
   In data storage layer we send processed data to the database server. We cover only the transport of data towards server

Components

Smart Shelves:

Smart Shelves is a type of event driven architecture. It is an architectural style which uses an event dispatcher to mediate between components.

Event-driven architectures have three key components:

- Event producers
- Event routers
- Event consumers.

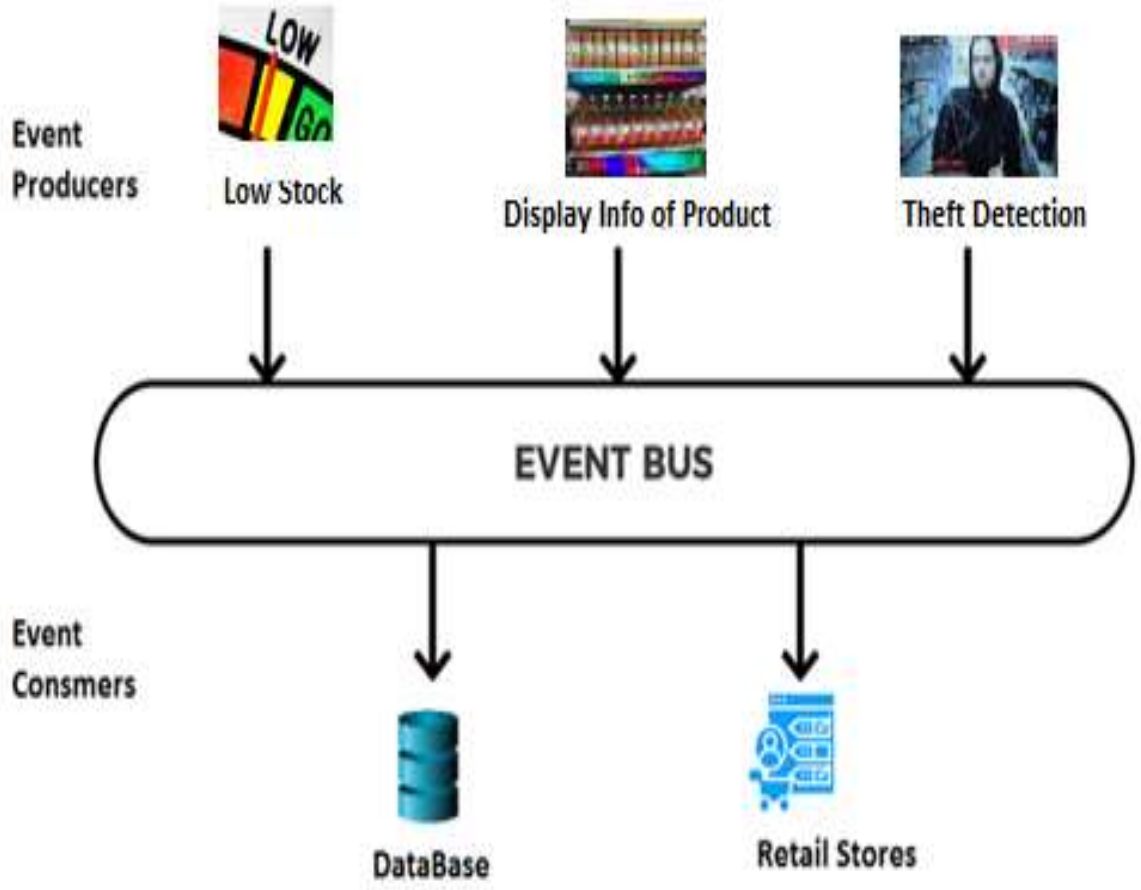
A producer publishes an event to the router, which filters and pushes the events to consumers. Producer services and consumer services are decoupled, which allows them to be scaled, updated, and deployed independently.

Smart shelves which are also known as “intelligent shelves” are electronically connected shelves which are designed to automatically keep track of inventory in any retail establishment. Manufactures can collect real-time data about their products usage and get sales data. Smart shelves use a combination of sensors, digital displays and RFID tags to provide detailed product information, marketing and cross-selling suggestions, while providing retailers invaluable insights into customer preferences and shopping patterns. Smart shelves technology enables automated tracking of stock availability and informs retail stores if the items are running low.

Smart shelves displays the price and all details about the products. It also senses if there is any suspicious activity going on like theft or if any item is misplaced. By the sensors that are installed, it senses these kind of activities. It also recognize the customers because it saves the customer info in the database and it also know what is the interest of that particular customer.

When the customer picks an item from the shelf, product levels get low, Smart Shelves send a refill notification to employees/ store keepers on their connected mobile device. The stock is replenished by the store keepers.

Smart Shelves can interact with apps installed on customers’ smartphones that contain purchase history information. As customers walk down an aisle, retailers can provide personalized offers based on past purchases of similar products. If customers create a shopping list in the app, Smart Shelves can direct them to where the products are located.



Robots:

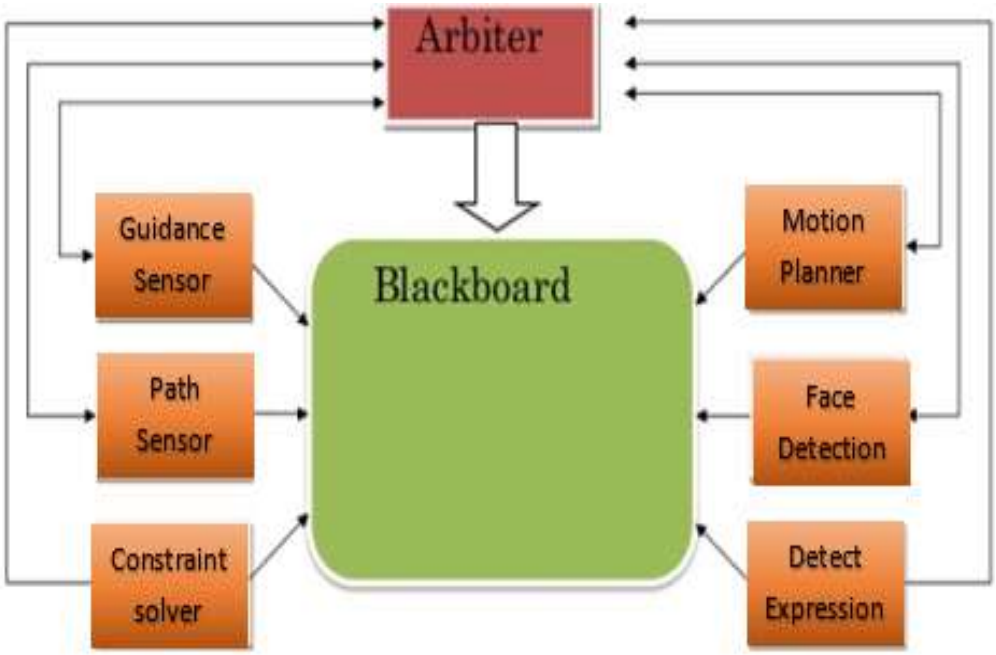
Robots in Smart Retail Industry is an example of blackboard architecture. In blackboard architecture, the data store is active and its clients are passive. Therefore the logical flow is determined by the current data status in data store. It has a blackboard component, acting as a central data repository, and an internal representation is built and acted upon by different computational elements. the data store is active and its clients are passive. Therefore the logical flow is determined by the current data status in data store. It has a blackboard component, acting as a central data repository, and an internal representation is built and acted upon by different computational elements. In this style, the components interact only through the blackboard. The data-store alerts the clients whenever there is a data-store change.

In Blackboard architecture, eash knowledge source has a special knowledge for a problem. Knowledge source read and write information to blackboard. Arbiter is basically a controller that controls all the knowledge sources.

Robots have rolled into retail, from six-foot-tall free-moving machines spotting spills in Giant Foods Stores to autonomous shelf-scanners checking inventory in Walmart. These robots free up workers from routine tasks, presumably giving humans more time for customer interaction — but that's only the beginning.

In robots there will be a guidance sensor that guides the customers about what they want. Robots can also identify the facial expressions of humans like if they look confused then they will help them to guide what they want. Robots also have the path detection sensor that detect if there is any obstacle in their path. It can also plans its motion like how it will go. The real benefit of retail robots is the opportunity to capture more granular data about the products on the shelves and customer buying patterns, which can increase efficiency and accuracy in inventory management.

Now a days, robots in retail coming equipped with machine vision algorithms that are able to capture and analyze images and videos and respond accordingly. They can also detect unclean areas and clean it by themselves. They also greets customers when they enter the stores and move the items to the right shelves.



Smart Mobile App:

Smart mobile app is an example of Publish and subscribe architecture. In Publish and Subscribe architecture there are three main components:

- Publisherse
- Event bus/broker
- Subscribers.

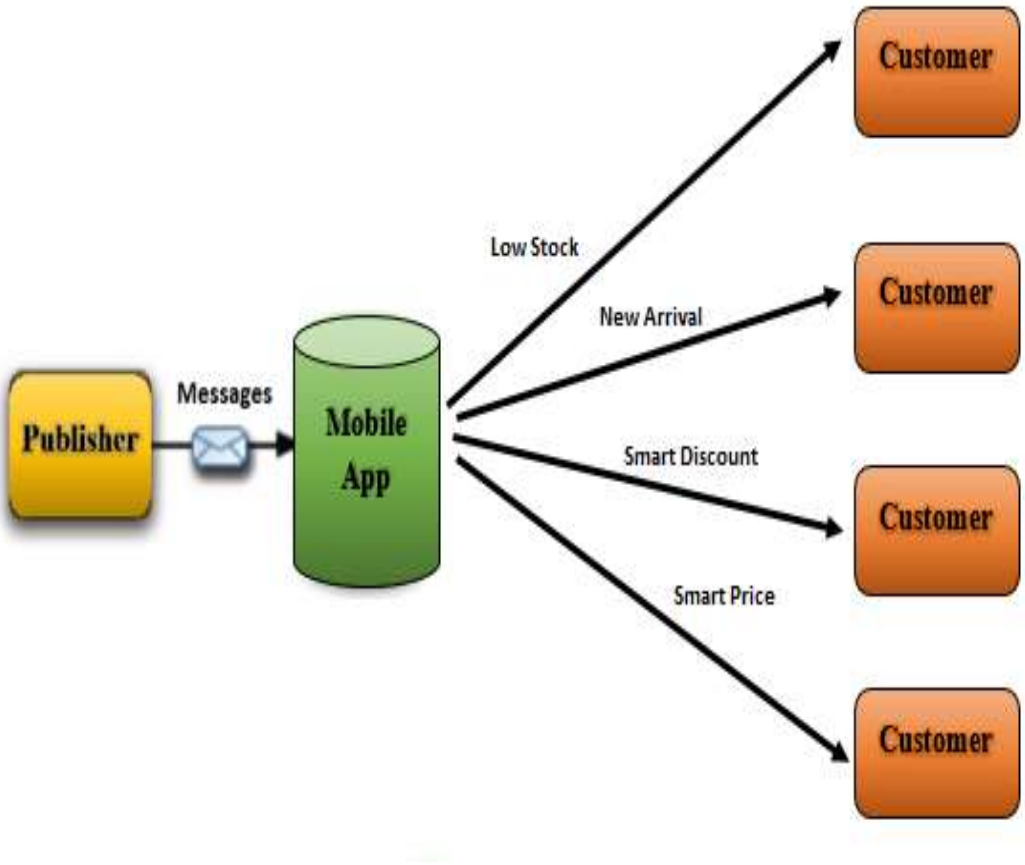
A Publish Subscribe Architecture is a messaging pattern where the publishers broadcast messages, with no knowledge of the subscribers. Similarly the subscribers listen out for messages regarding topic/categories that they are interested in without any knowledge of who the publishers are. The event bus transfers the messages from the publishers to the subscribers.

Smart mobile apps in retail are the apps that clients have installed in there cell phones. Smart retail mobile app understands the human behaviours. When a customer enter in smart retail shop, smart mobile app delivers a push notification with a personalized greeting and an exclusive offer.

Smart mobile app also tells the customer if there is any new arrival in the retail. It also tells the customer about the low stock of the item that is the customer's favourite item. It can detect the customer's interests and according to that gives offers. When a customer passes through a shelve, it will push a notification about the product and customer got all the details about the product in his/her phone.

If there is any smart discount on any product for example it knows about the person interest then it will give the discount to a specific person. From camera it detects what the customer want from database and then it will give discount to a specific person. It will also don't sent notification to the customer who are likely to pay full. This will be done by the information or data that is carried out by database.

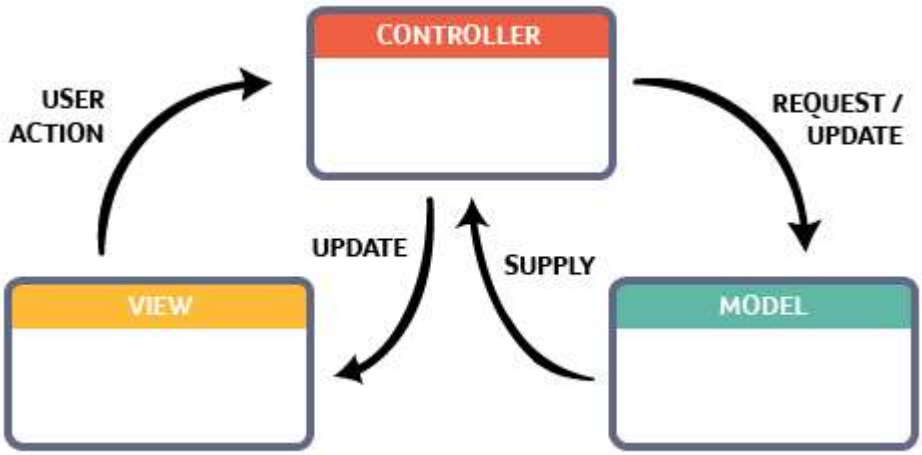
It also tells customers about the smart price. For Example if multiple people are purchasing same thing then its price will reduce. Mobile app will sends the notification about that item. Smart Pricing let the customers to set their prices to automatically go up or down based on changes in demand for listings like theirs.



Web and Mobile application layer

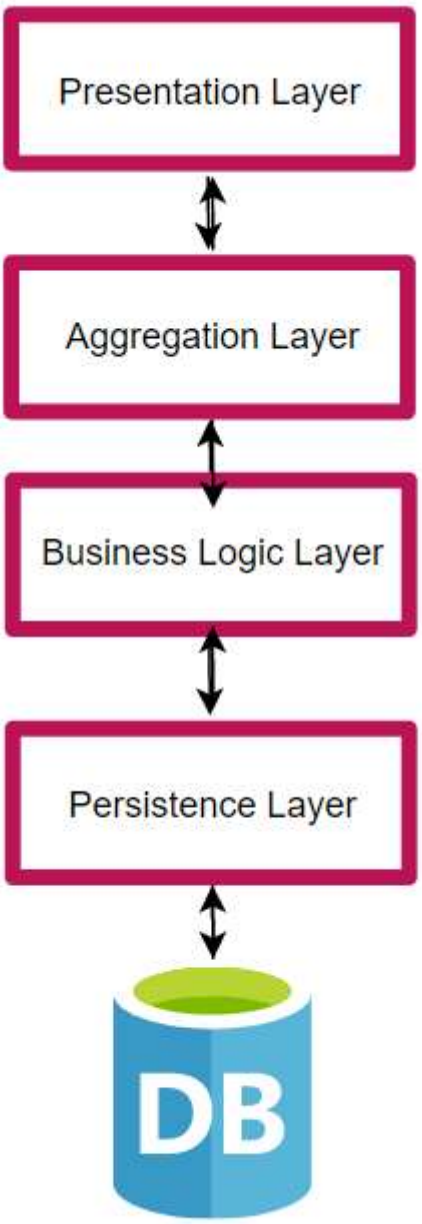
Earlier in this document we mentioned the architecture of web and mobile applications as client server architecture that is because of we are considering it from higher level of abstraction to show communication between cloud and client application but now as we are moving towards details we are now focusing on only the client part of these applications.

Now we will use Model view controller style for these applications We are using mvc for these applications because the model view controller pattern separates user interface functionality from application functionality. With MVC, application functionality is divided into three types of components: models, which contain the application data; views, which display the underlying data and interact with the user; and controllers, which mediate between the model and the view and manage state changes. The MVC pattern supports usability, as it allows the user interface to be designed and implemented separately from the rest of the application.



Database and Networking layer

Each location has its own database and persistence layer. The persistence layer is a middle ground between the database and business logic layer. It allows for easy access of data from the database and provides it to the business logic layer. This layer provides a good abstraction between the two layers and allows them to be independent of each other. This means the business logic layer is not concerned with how the data is stored in the database or how it is retrieved. And the database layer is not concerned with how the business logic is implemented. They can later both be upgraded or completely changed, and the other layer wouldn't be effected.


Presentation Layer
Aggregation Layer
Business Logic Layer
Persistence Layer
DB