

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION  
CSE 4317: SENIOR DESIGN II  
SUMMER 2021**



**LAMINAR WORKFLOW  
QUICK QUOTE**

**ERIC KENG  
SINDHU PARAJULI  
NISHAN THAPA  
ALISHA KUNWAR  
LAITH MARZOUQ**

## REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	04.06.2021	EK, LM, AK, NT, SP	document creation, Front Page
0.2	04.08.2021	EK, LM	4.1, All Figures
0.3	04.10.2021	AK, NT	2.2, 5, 5.1, 5.1.1, 5.1.2, 6
0.4	04.12.2021	EK	4.2, 4.3, Introduction
1.0	04.12.2021	LM	Final Draft Review
1.1	04.23.2021	EK, LM, AK, NT, SP	2.1, 2.2, 4.2 Edited
2.0	05.04.2021	EK, LM, AK, NT, SP	Final review of Architecture Design Specification Version-2
2.1	06.15.2021	EK	mongoDB to AWS
2.1	08.02.2021	SP, AK, NT, LM, AK	Close Out Review

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>System Overview</b>	<b>6</b>
2.1	User Layer Description . . . . .	7
2.2	Server Layer Description . . . . .	7
2.3	Database Layer Description . . . . .	7
<b>3</b>	<b>Subsystem Definitions &amp; Data Flow</b>	<b>8</b>
<b>4</b>	<b>User Layer Subsystems</b>	<b>9</b>
4.1	Authentication . . . . .	9
4.2	Buyer Interface . . . . .	10
4.3	Shop Interface . . . . .	11
<b>5</b>	<b>Server Layer Subsystems</b>	<b>13</b>
5.1	Parsing . . . . .	13
5.2	Price calculation . . . . .	13
5.3	Payment . . . . .	14
<b>6</b>	<b>Database Layer Subsystems</b>	<b>16</b>
6.1	Buyer Information . . . . .	16
6.2	Shop information . . . . .	16
6.3	Orders and DXF storage . . . . .	18

## LIST OF FIGURES

1	Architectural Diagram . . . . .	6
2	Data Flow Diagram . . . . .	8
3	Authentication Subsystem . . . . .	9
4	Buyer Interface Subsystem . . . . .	10
5	Shop Interface Subsystem . . . . .	11
6	Parsing Subsystem . . . . .	13
7	Price Calculation Subsystem . . . . .	14
8	Payment Subsystem . . . . .	15
9	Buyer information Subsystem . . . . .	16
10	Shop information Subsystem . . . . .	17
11	Order and DXF storage subsystem . . . . .	18

## LIST OF TABLES

2	Authentication interfaces . . . . .	10
3	Buyer interface . . . . .	11
4	Shop Interface . . . . .	12
5	Parsing interfaces . . . . .	13
6	Price calculation interfaces . . . . .	14
7	Payment Interfaces . . . . .	15
8	Buyer Information interfaces . . . . .	16
9	Shop Information interfaces . . . . .	17
10	Orders and DXF Storage interfaces . . . . .	18

## 1 INTRODUCTION

The quick quote application is a web based application that provides an instant quoting service to 2D cutting shops. This application is a tool for 2D cutting shops to use that will provide their buyers with instant price quotes for their services.

This application will be a full stack application hosting 3 main layers: React JS as the User Layer, Springboot JS as the Server Layer, and Amazon Web Services (AWS) as the database. React will host the view for all users of the application, Springboot will provide the processing power, and AWS will provide all of the storage for the application.

There are many subsystems to each layer. The user layer subsystems include: authentication, buyer interface, and shop interface. The server layer subsystems include: parsing algorithm, price calculation, and payment. The database layer subsystems include: buyer information, order / DXF storage, and shop information.

## 2 SYSTEM OVERVIEW

The Quick Quote application will be separated into three layers. Which are the User layer, Server layer, and Database layer. The User Layer is the primary interface for buyers and sellers using the Quick Quote application. The Server layer will perform all the application logic, and It acts as an interface between the user and database layers. The database layer is the layer where all of the information and data is stored.

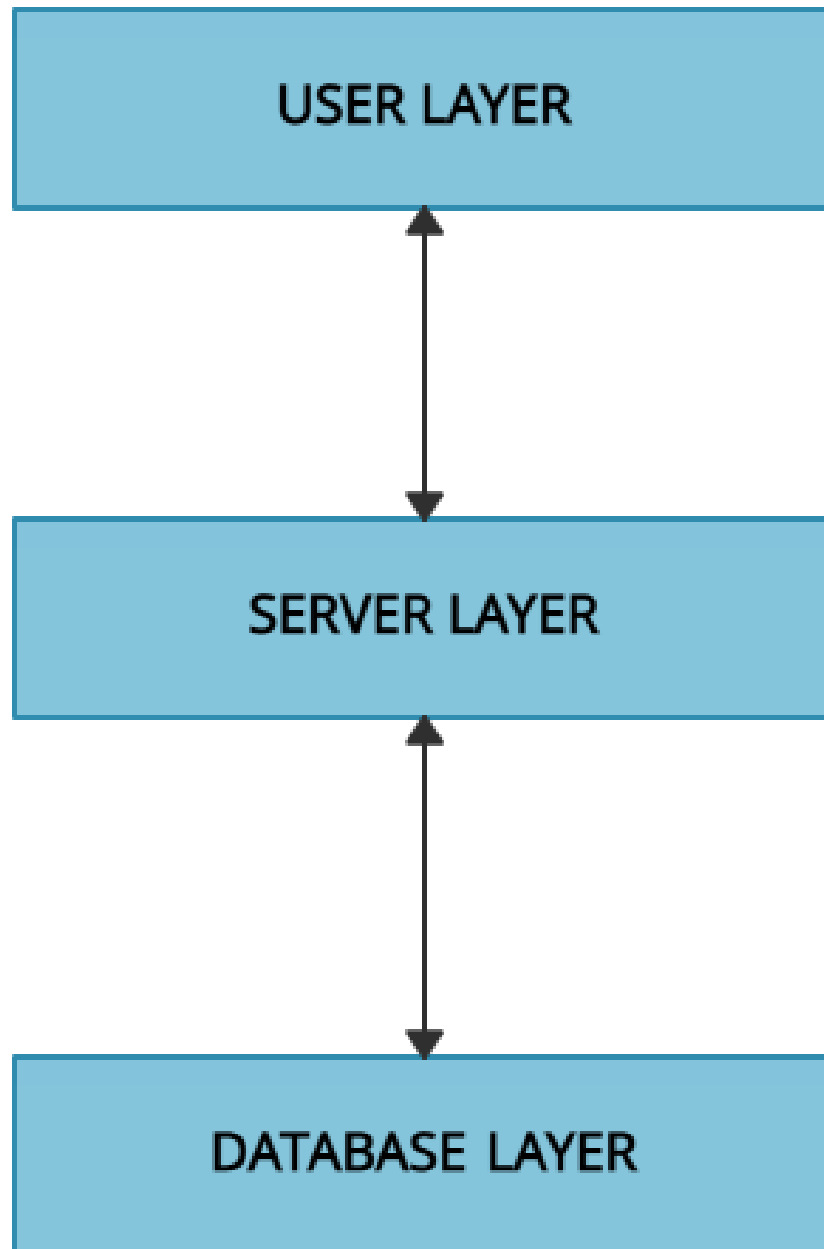


Figure 1: Architectural Diagram

## **2.1 USER LAYER DESCRIPTION**

The User Layer is the interface for buyers and sellers using the Quick Quote application. It allows easy access for sellers to customize their companies' landing page and keep up with orders, it also allows buyers to upload DXF files to easily request instant quotes from sellers and submit orders. The user layer must communicate with the server layer to reach the database layer in order to store and provide all necessary information for sellers and buyers and manage their accounts.

This layer is built and maintained using React JavaScript. It allows us to instantly and easily change the website formatting which helps with the customizable landing page and debugging.

## **2.2 SERVER LAYER DESCRIPTION**

The server layer is the layer which perform all the application logic. It acts as an interface between user and database layer. This layer perform all of the core functions of the application. It handles all the user request and respond it back. In our application, the server layer has three sub-systems.

The server layer is responsible for receiving HTTP request to parse DXF file. It retrieves DXF file as a parameter and implements parsing algorithm to get the parsed data. The data is used as an entity to perform price calculation operation. The price calculation sub-system fetch data from database layer needed to calculate the price. The server layer generates the quote and responds back to the user interface within a few seconds. If user proceeds with payment, the payment layer is invoked which extracts user information, validates card information, invoke third party API for payment handling, and responds accordingly.

## **2.3 DATABASE LAYER DESCRIPTION**

The database layer is the layer where all of the information, data and files are stored. The database layer interacts with the server layer to store or retrieve the needed data. Based on the scope of our application, we need to store Buyer (Customer) information, Seller (Shop) information and parameters, order placed by buyer along with DXF files. All of the data storage and retrieval on database layer is done through the server layer.

The database layer stores information of the buyer and seller/shops once an account is created. These credential needs to be retrieved and checked while performing login. Similarly, We also need to store and retrieve the quoting parameters of the shop/sellers, which is the basis for the quote. In addition, the DXF file uploaded by buyers, and order placed by them has to be stored in the database and retrieved based on the needs.

### 3 SUBSYSTEM DEFINITIONS & DATA FLOW

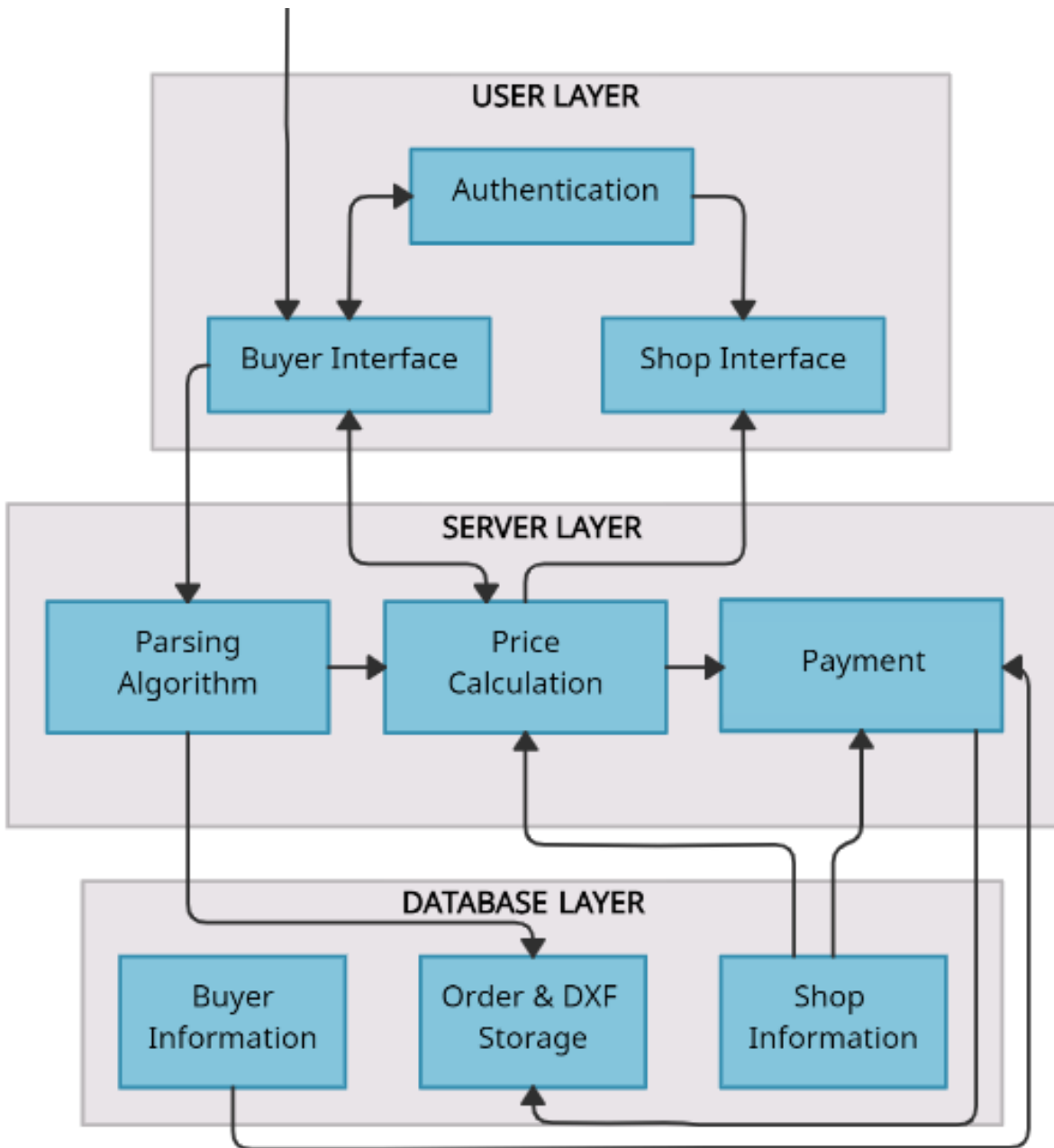


Figure 2: Data Flow Diagram



## 4 USER LAYER SUBSYSTEMS

The User Layer is the primary interface for buyers and sellers using the Quick Quote application. It allows easy access for sellers to customize their companies' landing page and keep up with orders. It allows buyers to upload DXF files to easily request instant quotes from sellers and submit orders. The user layer must communicate with the database layer to store and provide all necessary information for sellers and buyers to manage their accounts and orders as necessary. The subsystem layers include: authentication, buyer-interface, and shop-interface.

### 4.1 AUTHENTICATION

The Authentication layer will be an option for both sellers and buyers. Both sellers and buyers will have the same initial view in this layer. A valid email and password will be required for those who already have an account. For those who do not have an account there will be an option to create an account. Once logged in through this layer the seller will be able to navigate to the seller interface and the buyer will have access to their order history and account information.

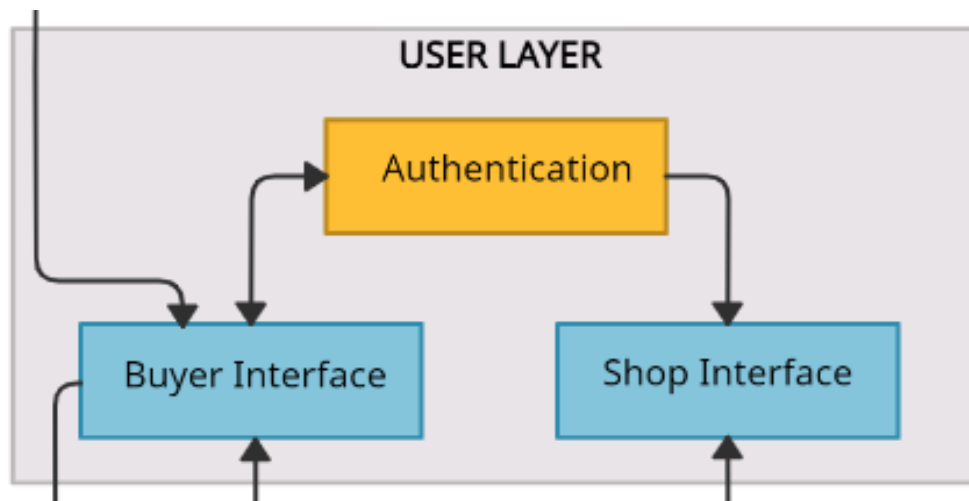


Figure 3: Authentication Subsystem

#### 4.1.1 ASSUMPTIONS

The seller login is the most important aspect of this layer as it will allow customers of the Quick Quote application to customize their landing page to fit their companies' needs. Sellers will be able to customize quoting parameters, view order requests, and edit the view of the buyer-side layer. Buyers will only need this layer if they would like to store their information and view order history.

#### 4.1.2 RESPONSIBILITIES

The login layer is responsible for verifying if users have a valid account and then navigating them to the appropriate view after successful login. This layer also holds the responsibility of adding new users to the database.

#### 4.1.3 AUTHENTICATION INTERFACES

The login and authentication layer will need to communicate with the database server to check the credentials of sellers and buyers who try to login. It must also be able to add new users who create an account through this layer.

Table 2: Authentication interfaces

ID	Description	Inputs	Outputs
#01	Authentication	email password	Object to server
#02	Create an account	name email password	valid/invalid Object to server

## 4.2 BUYER INTERFACE

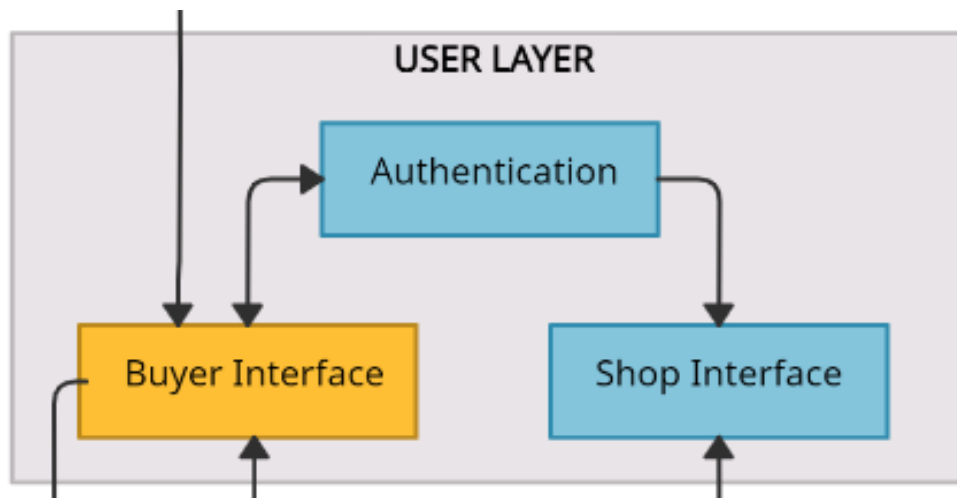


Figure 4: Buyer Interface Subsystem

### 4.2.1 ASSUMPTIONS

The buyer interface will be the default view when entering the application. This will give quick access to dragging and dropping or uploading DXF files, receiving quotes, and logging in. Everyone who opens the application will have the same initial view as if entering a store to purchase merchandise.

### 4.2.2 RESPONSIBILITIES

The buyer layer will be responsible for giving navigation access points to all users of the Quick Quote product. The focus of the page should be the request quote option. There will be a menu for other navigation options like the login feature for the Shop owners or buyers who already have a saved account.

### 4.2.3 BUYER INTERFACE

The buyer interface is the main entry point to the application. It hosts the access point for the Authentication subsystem. It also communicates with the parsing algorithm and price calculation to provide instant quotes for buyers.

Table 3: Buyer interface

ID	Description	Inputs	Outputs
#01	Request a quote	DXF file	Upload to Parsing Algorithm
#01	Price Calculation Display	Price Calculation	Display Cost to buyer

### 4.3 SHOP INTERFACE

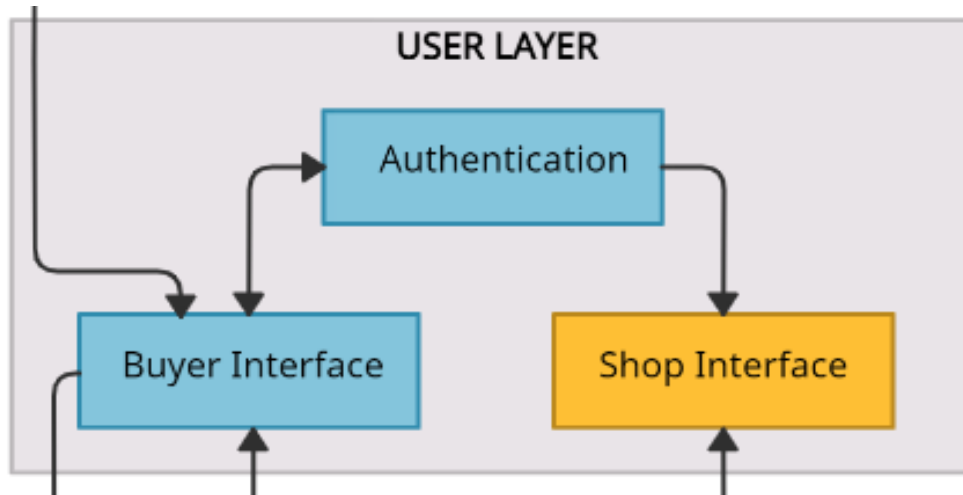


Figure 5: Shop Interface Subsystem

#### 4.3.1 ASSUMPTIONS

The shop interface is the marketable aspect of our application to 2D cutting shops who will be able to customize their buyer interface design. This feature will be on a subscription base for shop owners.

#### 4.3.2 RESPONSIBILITIES

The shop interface will allow shop owners to customize the quoting parameters that provide quotes in the buyer interface. This interface will also allow shop owners to see useful information regarding their shop like pending orders.

#### 4.3.3 SHOP INTERFACE

The shop interface will need to communicate directly with the price calculation interface and the buyer interface. It will provide quoting parameters to the price calculation interface that will change the amount quoted for a 2D cutting request based off line distance, pierce points, and material. It will also provide view changes like pictures and color schemes to the buyer interface to allow the shop owners to customize the buyer's view of their services.

Table 4: Shop Interface

ID	Description	Inputs	Outputs
#01	Price quoting parameters	Cutting costs Cost per pierce point	Update to price calculation
#02	Buyer interface customization	Photo Color Scheme Text	Update to buyer interface

## 5 SERVER LAYER SUBSYSTEMS

The server layer is the layer which perform all the application logic. It acts as an interface between user and database layer. It receives HTTP request made from the user interface, communicates with the database retrieving the necessary data to perform the business operation, and sends the response to the user layer.

### 5.1 PARSING

Once the DXF file is uploaded in the user interface layer, using POST request the file is passed as parameter to the parsing layer. In this layer the file can be read and parse. It contains logic (parsing algorithm) for parsing the file such that all the necessary parameters like total linear distance of cut, length, width, total area of material used, number of pierce/plunge points are parsed out.

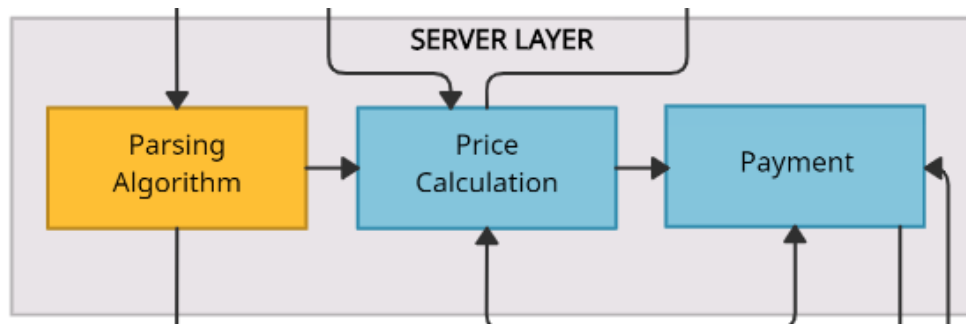


Figure 6: Parsing Subsystem

#### 5.1.1 ASSUMPTIONS

This subsystem is for parsing the DXF file. After the file is passed to the parsing layer all the necessary data is parsed out within a few seconds.

#### 5.1.2 RESPONSIBILITIES

The responsibility of this layer is that it should handle HTTP request and get the file as a parameter. It should be able to parse the file and fetch all the necessary parameters.

#### 5.1.3 PARSING INTERFACES

Table 5: Parsing interfaces

ID	Description	Inputs	Outputs
#01	Parsing DXF file	DXF file	Parameters (Parsed data)

## 5.2 PRICE CALCULATION

After parsing the DXF files, data is obtained about the material that is ordered. Using the data, price is calculated using the current cost, dimensions of materials obtained by the DXF parsing. The price calculation involves arithmetic formulas and is server based. Server side calculations results in faster results and easier to integrate.

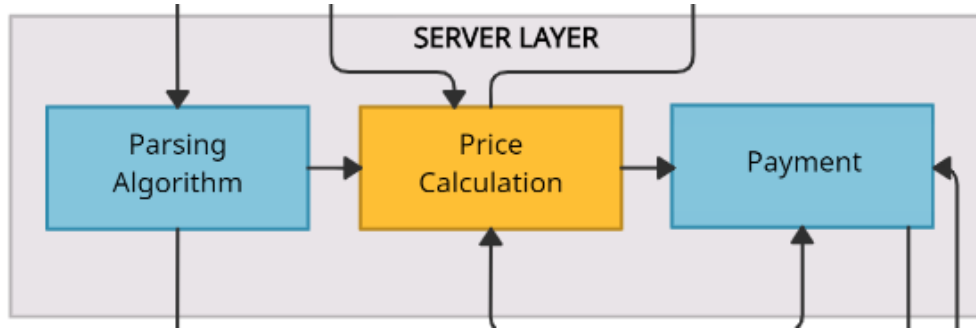


Figure 7: Price Calculation Subsystem

### 5.2.1 ASSUMPTIONS

This subsystem is for calculating the price. So it is assumed that the DXF file is already parsed and data can be fetched.

### 5.2.2 RESPONSIBILITIES

The responsibility of this layer is that it should be able to get the data after being parsed, perform meaningful calculations based on the dimensions obtained from the DXF files and cost of the material fetched from the input request. It is responsible for pulling referential data from external service or database if needed to calculate pricing. This layer is responsible for providing price quote as quickly as possible.

### 5.2.3 PRICE CALCULATION INTERFACES

Table 6: Price calculation interfaces

ID	Description	Inputs	Outputs
#01	Perform price calculation	Parsed data material data	Calculated price

## 5.3 PAYMENT

In this layer, payment process is handled. After the quote is generated, if the user proceeds for payment, payment layer is invoked which extracts all information entered by user such as credit-card information, user information and so on in the server layer. Afterwards, it performs some in-house validation like valid credit card details, etc and upon successful validation a third party API for handling payments is invoked and sends response accordingly.

### 5.3.1 ASSUMPTIONS

This layer assumes that quote for payment is already generated and is valid.

### 5.3.2 RESPONSIBILITIES

The responsibility of this layer is to validate the information about payment provided by the user, invoke the third party payment API to process the payment and return response to the user-interface accordingly.

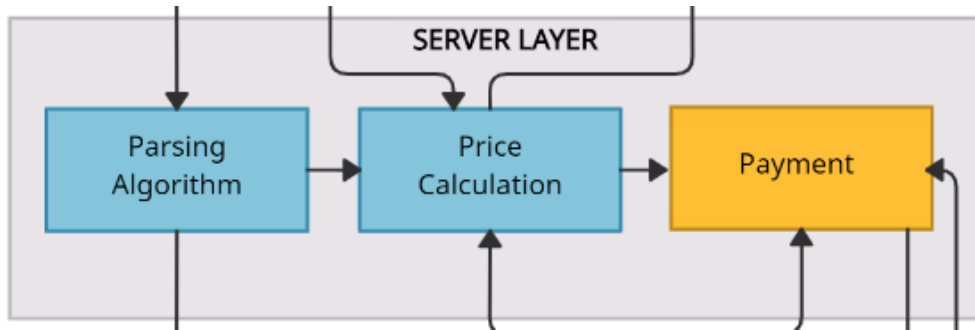


Figure 8: Payment Subsystem

### 5.3.3 PAYMENT INTERFACES

Table 7: Payment Interfaces

ID	Description	Inputs	Outputs
#01	Process Payment	Validated payment information	Boolean response

## 6 DATABASE LAYER SUBSYSTEMS

The database layer is the layer where all of the information and data are stored. The database layer is divided into three different subsystem based on the types and nature of data stored.

### 6.1 BUYER INFORMATION

In this database subsystem, information and credentials related to Buyers are stored. Buyers provide information like email, password, address, phone numbers etc once they create an account. These information are sent to the server layer. The server layer interacts with this subsystem to store the data. Moreover, the login credentials of the buyers will also be checked by the server layer based on the information retrieved from this subsystem.

Other than that, Buyer information like name, email, address, payment etc. can be retrieved and sent to the server layer subsystems whenever required.

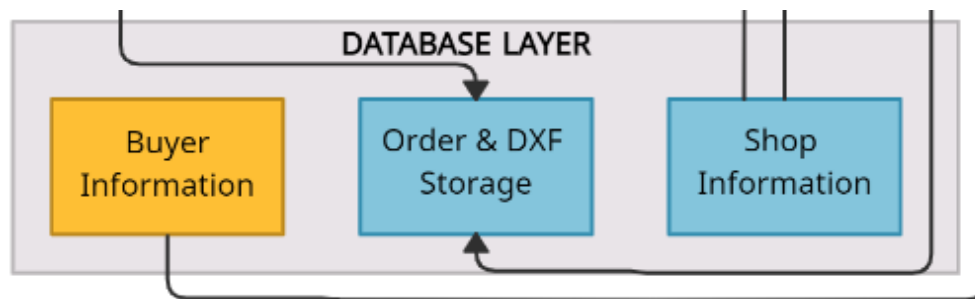


Figure 9: Buyer information Subsystem

#### 6.1.1 ASSUMPTIONS

Buyer Sign up is already done and Buyer data is already stored in this layer.

The buyer login and authentication is done by authentication function of Amazon Web Services (AWS) on server layer.

#### 6.1.2 RESPONSIBILITIES

The major responsibility of this subsystem is to store and manage the information of the buyer. This subsystem also retrieve buyer data, payment information and sends to the server layer based on the query done by server.

#### 6.1.3 BUYER INFORMATION INTERFACES

Table 8: Buyer Information interfaces

ID	Description	Inputs	Outputs
#01	Save buyer's data	Buyer Information	saves data
#02	Retrieve buyer's data	db Query	data objects

## 6.2 SHOP INFORMATION

This database subsystem stores information related to shop/seller. The shop uses quoting parameters like material density, cut speeds, price/inch/process, Material cost which are stored in this subsystem.



These parameters are required by server layer, while quoting price or placing order. This subsystem retrieves these quoting parameters and send it to the server layer for price estimation.

Other than that, this subsystem also stores shop credentials like name, email, address etc while signing up. Shop credentials stored in this layer can be retrieved and can be used for authentication.

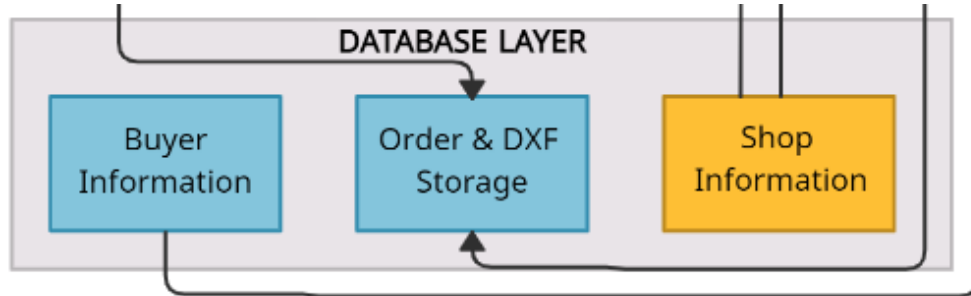


Figure 10: Shop information Subsystem

### 6.2.1 ASSUMPTIONS

Seller Sign up is already done and data is already stored in this layer The seller login and authentication is done by authentication function of Amazon Web Services (AWS) on server layer.

### 6.2.2 RESPONSIBILITIES

The major responsibility of this layer is to store quoting parameters like material density, cut speeds, price/inch/process, Material cost of every shops. This stored information can be retrieved and used by server layer for price estimation.

This subsystem also stores seller credentials which can be used for login and authentication.

### 6.2.3 SHOP INFORMATION INTERFACES

Table 9: Shop Information interfaces

ID	Description	Inputs	Outputs
#01	Saves shop information	shop information name, address email etc.	saves the data
#02	Retrieves shop information and credentials	DB query	data objects
#03	Stores quoting parameters for shops	Quoting parameters like: material cost, density, price/inch/process etc.	saves the parameters
#04	Retrieves quoting parameters	DB query	data objects of parameters

### 6.3 ORDERS AND DXF STORAGE

This subsystem is for storing the orders placed by the customers. Once payment for the order is approved, the server layer stores the order details, payment and DXF files for every order in this subsystem. Similarly, The order details can be retrieved with some query and sent to server layer. This order detailed is displayed to the buyer and seller along with the DXF files.

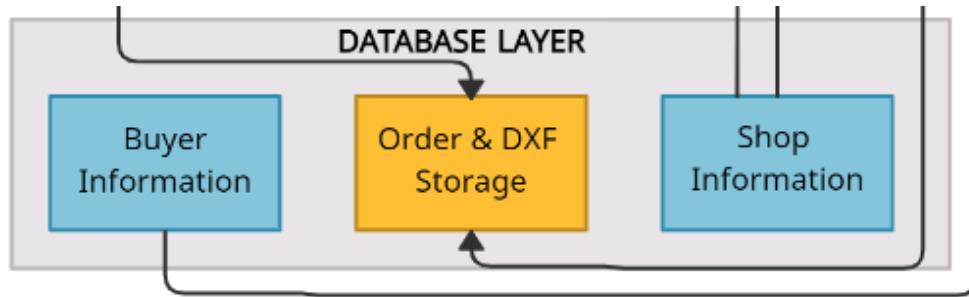


Figure 11: Order and DXF storage subsystem

#### 6.3.1 ASSUMPTIONS

The payment for the order is approved.

#### 6.3.2 RESPONSIBILITIES

The major responsibility of this layer is to store the order details, buyer information and DXF files for that order. This stored order details and DXF files can be retrieved and sent to the server layer which can be viewed by Buyer and the shop.

#### 6.3.3 ORDERS AND DXF STORAGE INTERFACES

Table 10: Orders and DXF Storage interfaces

ID	Description	Inputs	Outputs
#01	Store order details and DXF files	Order information Payment information DXF files	Stores order details and DXF files
#02	Retrieves order details and DXF files	DB query	Objects of File and order details

## REFERENCES