Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Summer 2021

Laminar Workflow
Quick Quote

Nishan Thapa
Alisha Kunwar
Laith Marzouq
Eric Keng
Sindhu Parajuli

Revision History

Revision	Date	Author(s)	Description
0.1	06.15.2021	LM, EK	document creation
0.2	06.15.2021	EK	Created all Sections and Headers. Uploaded all Images. Filled general descriptions
0.3	06.25.2021	AK, NT, SP	Description of each layer and subsystem
1.0	06.27.2021	EK, NT, AK, SP, LM	Final Draft Edits
2.1	08.05.2021	SP, AK, NT, LM, AK	Close Out Review

Contents

List of Figures

List of Tables

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 JavaScript as the User Layer, Spring Boot JavaScript as the Server Layer, and Amazon Web Services (AWS) as the database. React will host the view for all users of the application, Spring Boot 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 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.

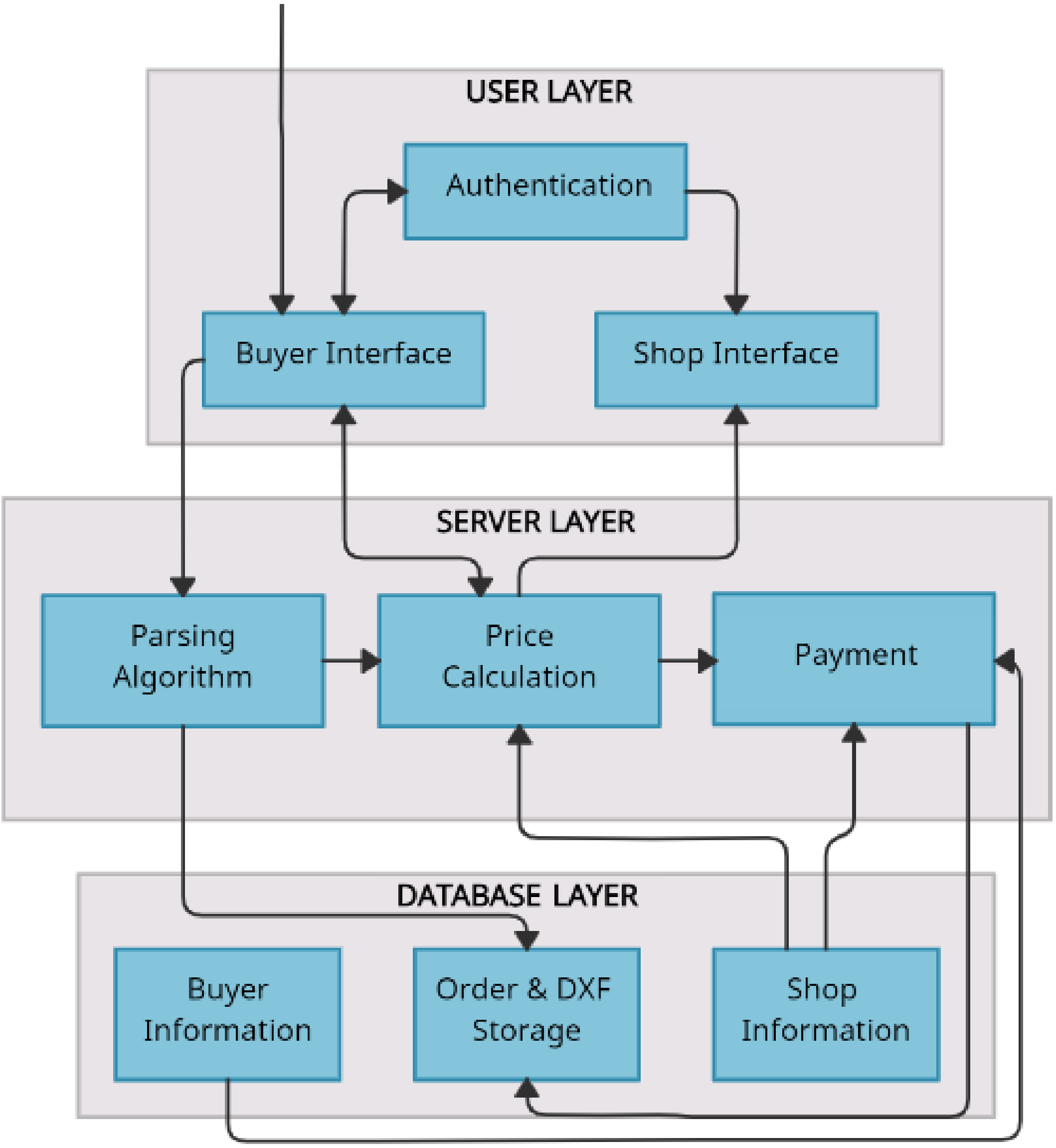

Figure 1: System architecture

3 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.

3.1 Layer Hardware

Laminar workflow project is a software application, there will not be hardware layer in any aspect of the project.

3.2 Layer Operating System

This application will be able to run in any operating system.

3.3 Layer Software Dependencies

The user layer is mostly built using Reactjs(Javascript libraries). The sign-in/login page is created using Reactjs and Amazon cognito.

3.4 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.

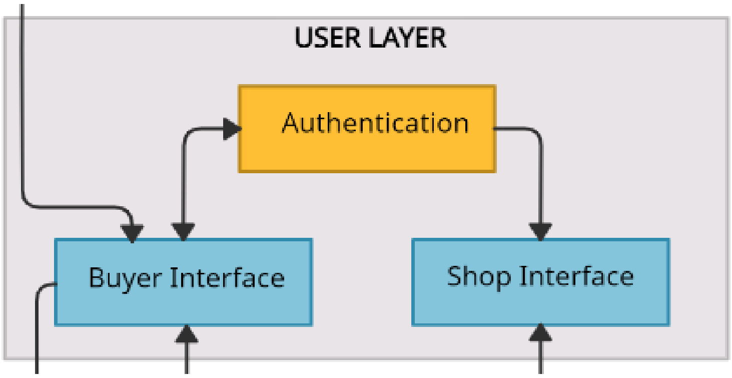

Figure 2: Authentication Description Diagram

3.4.1 Authentication Hardware

The subsystem is associated to software only, there will not be any hardware components.

3.4.2 Authentication Operating System

The subsystem will be running on Windows and Mac operating system.

3.4.3 Authentication Software Dependencies

Amazon Cognito is used for authentication and user management for this application.

3.4.4 Authentication Programming Languages

The language used is JavaScript.

3.4.5 Authentication Data Structures

The subsystem will get user's information and authenticate the user, and grant access to the respective dashboard.

3.4.6 Authentication Data Processing

The subsystem will use the stored data i.e user information from the database to verify the identity of a user.

3.5 Buyer Interface

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

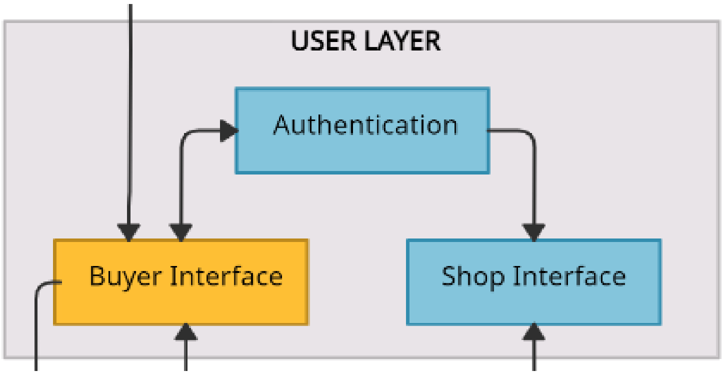

Figure 3: Buyer Interface Description Diagram

3.5.1 Buyer Interface Hardware

The subsystem is associated to software only, there will not be any hardware components.

3.5.2 Buyer Interface Operating System

The subsystem will be running on Windows and Mac operating systems.

3.5.3 Buyer Interface Software Dependencies

Reactjs(Javascript library) is used to create buyer interface.

3.5.4 Buyer Interface Programming Languages

The language used is Javascript.

3.5.5 Buyer Interface Data Structures

The subsystem allows users to upload the DXF files.

3.5.6 Buyer Interface Data Processing

The subsystem communicates to the parsing layer, receives the quotes and displays it.

3.6 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.

3.6.1 Shop Interface Hardware

The subsystem is associated to software only, there will not be any hardware components.

3.6.2 Shop Interface Operating System

The subsystem will be running on Windows and Mac operating systems.

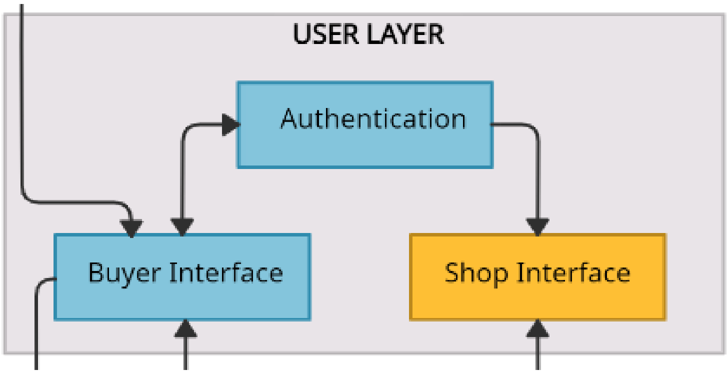

Figure 4: Shop Interface Description Diagram

3.6.3 Shop Interface Software Dependencies

Reactjs(Javascript library) is used to create buyer interface.

3.6.4 Shop Interface Programming Languages

The language used is JavaScript.

3.6.5 Shop Interface Data Structures

The subsystem communicates price interface of the server layer.

3.6.6 Shop Interface Data Processing

The subsystem will provide quoting parameters to the price calculation interface that will change the amount quoted for a 2D cutting request based on line distance, pierce points, and material.

4 Server Layer Subsystems

4.1 Layer Hardware

The subsystem is associated to software only, there will not be any hardware components.

4.2 Layer Operating System

All the subsystem in the backend will be running on AWS cloud. The server will be hosted on AWS.

4.3 Layer Software Dependencies

Spring Boot framework is used to communicate with the database and the user interface.

4.4 Parsing Algorithm

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.

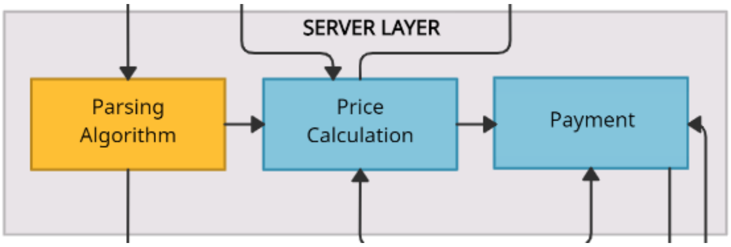

Figure 5: Parsing Algorithm Description Diagram

4.4.1 Parsing Algorithm Hardware

The subsystem is associated to software only so there will not be any hardware components.

4.4.2 Parsing Algorithm Operating System

The subsystem in the backend will be running on AWS cloud and the server will be hosted on AWS.

4.4.3 Parsing Algorithm Software Dependencies

Spring Boot framework is used to communicate with the database and the user interface.

4.4.4 Parsing Algorithm Programming Languages

The subsystem will be programmed in Java using the Java Spring Boot.

4.4.5 Parsing Algorithm Data Structures

The DXF file from the user interface is parsed into the parameters like area, number of pierces, which is used for price calculation.

4.4.6 Parsing Algorithm Data Processing

The DXF file is passed as a parameter to the parsing layer. The parsing algorithm is used to parse the DXF file.

4.5 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.

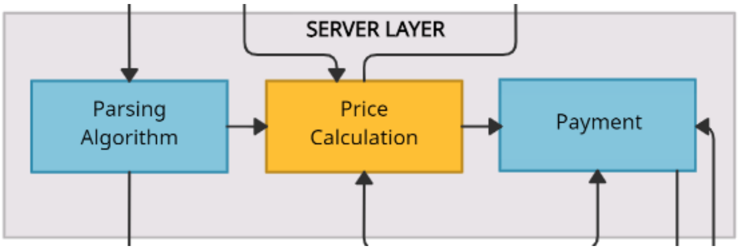

Figure 6: Price Calculation Description Diagram

4.5.1 Price Calculation Hardware

The subsystem is associated to software only, there will not be any hardware components.

4.5.2 Price Calculation Operating System

The subsystem in the backend will be running on AWS cloud and the server will be hosted on AWS.

4.5.3 Price Calculation Software Dependencies

Spring Boot framework is used to communicate with the database and the user interface.

4.5.4 Price Calculation Programming Languages

The subsystem will be programmed in Java using the Java Spring Boot.

4.5.5 Price Calculation Data Structures

The data is obtained from the parsing layer. The current cost is retrieved from the database for the dimension of the materials and price is calculated.

4.5.6 Price Calculation Data Processing

The subsystem performs price calculation algorithm using the arithmetic formulas to find the total cost.

4.6 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.

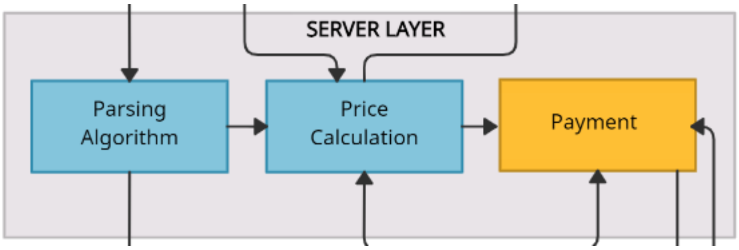

Figure 7: Payment Description Diagram

4.6.1 Payment Hardware

The subsystem is associated to software only, there will not be any hardware components.

4.6.2 Payment Operating System

The subsystem in the backend will be running on AWS cloud and the server will be hosted on AWS.

4.6.3 Payment Software Dependencies

Spring Boot framework is used to communicate with database and the user interface.

4.6.4 Payment Programming Languages

The subsystem will be programmed in Java using the Java Spring Boot.

4.6.5 Payment Data Structures

If the user proceeds for payment this subsystem extracts all the information entered by the user such as credit card information.

4.6.6 Payment Data Processing

The subsystem performs in-house validation of the provided payment information using third party API for handling payment and sends response accordingly.

5 Database Layer Subsystems

5.1 Layer Hardware

The subsystem is associated to software only, there will not be any hardware components.

5.2 Layer Operating System

All subsystem will be running on AWS cloud.

5.3 Layer Software Dependencies

The Amazon DynamoDb will be used as database for storing, updating and retrieving all the information required for server and user layer.

5.4 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.

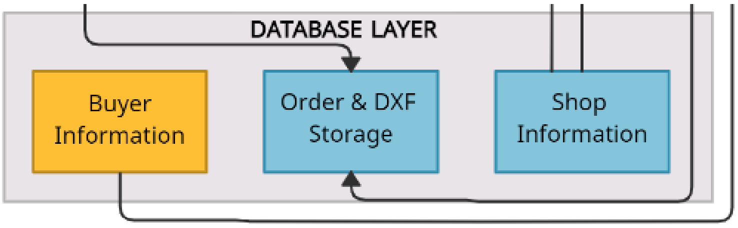

Figure 8: Buyer Information Description Diagram

5.4.1 Buyer Information Hardware

The subsystem is associated to software only, there will not be any hardware components.

5.4.2 Buyer Information Operating System

The subsystem will be running on AWS cloud.

5.4.3 Buyer Information Software Dependencies

The Amazon Dynamodb will be used as database for storing or updating the buyer information.

5.4.4 Buyer Information Programming Languages

The programming language used is Java with Spring Boot framework.

5.4.5 Buyer Information Data Structures

After buyer creates an account, the subsystem will obtain and store the buyer information. The data stored can be retrieved by the server layer whenever required.

5.4.6 Buyer Information Data Processing

The buyer information is sent to server layer. The server layer interacts with this subsystem to store the data. When the user logs in, the server layer checks the credentials with the data stored by retrieving from this subsystem.

5.5 Order 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.

5.5.1 Order and DXF Storage Hardware

The subsystem is associated to software only, there will not be any hardware components.

5.5.2 Order and DXF Storage Operating System

The subsystem will be running on AWS cloud.

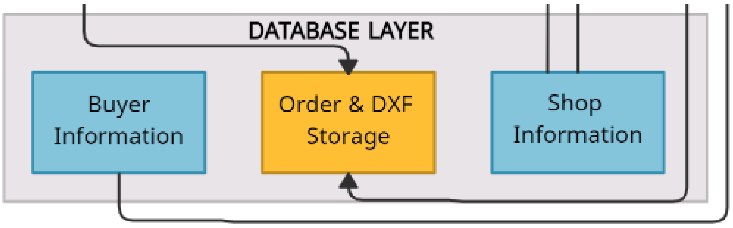

Figure 9: Order and DXF Storage Description Diagram

5.5.3 Order and DXF Storage Software Dependencies

Amazon Dynamodb will be used as the database for storing the orders placed by the customers.

5.5.4 Order and DXF Storage Programming Languages

The programming language used is Java with Spring Boot framework.

5.5.5 Order and DXF Storage Structures

After approval of the payment, the order details and payment information is stored in this subsystem.

5.5.6 Order and DXF Storage Data Processing

The order details and payment for every order is stored in this subsystem. The order details can be retrieved with queries and sent to server layer.

5.6 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.

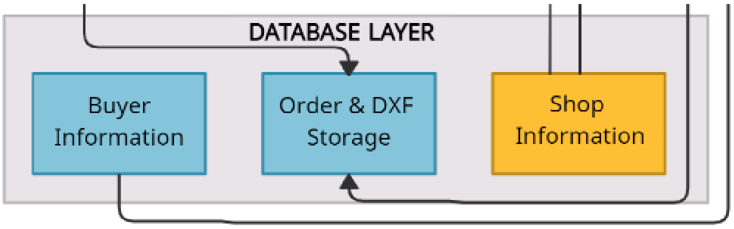

Figure 10: Shop Information Description Diagram

5.6.1 Shop Information Hardware

The subsystem is associated to software only, there will not be any hardware components.

5.6.2 Shop Information Operating System

The subsystem will be running on AWS cloud.

5.6.3 Shop Information Software Dependencies

The Amazon Dynamodb will be used as database for storing information related to shop/seller.

5.6.4 Shop Information Programming Languages

The programming language used is Java with Spring Boot framework.

5.6.5 Shop Information Data Structures

The subsystem stores shop information like name, email,address. It also stores cost of each materials.

5.6.6 Shop Information Data Processing

The subsystem retrieves material cost and passed to the server layer to perform the price estimation based on the parsed parameters.

6 Appendix A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

References