

Executive Summary

Manufacturing job shops that provide 2D cutting (laser, plasma, waterjet) take too long to provide customers with a quote. With online shopping being so popular the desire for instant results is expected by customers. Quotes done by a person are also not consistent due to the tedious process for creating a quote that allows a lot of room for error.

By creating an automated instant quoting program, customers will be able to get an accurate quote instantly from companies that offer 2-D cutting services. This will help companies to get and retain customers as companies will have a reliable and easy to use process for getting accurate quotes for their desired product.

Background

We are going to develop a website that will automate quoting for job shops offering 2d cutting services (laser, plasma, waterjet) using a DXF file. At the moment, quoting is not consistent and takes up to 2 - 4 days.

With our Quick Quote application, we are looking to provide customers with instant, consistent, and reliable quotes. Each job shop will have their own branded landing page for customers to drop in DXF files and receive their pricing. Job shops will be allowed to customize their pricing parameters as they please.

Key Requirements

When designing Quick Quote, We produced some key requirements that we thought were crucial to the success of our application.

Table 1. Key Requirements

#	Requirement	Priority
1	Parsing DXF File	Critical
2	Instant Price Quote	Critical
3	Consistent Output Generation	Critical
4	Drag and Drop File Feature	High
5	Modifiable Quoting Parameters (JOB SHOP)	Critical
6	Modifiable Quoting Parameters (USER)	Critical
7	All Code Maintained on GitHub	High
8	Database Server	Critical

Detailed Design Phase

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 the information and data is stored.

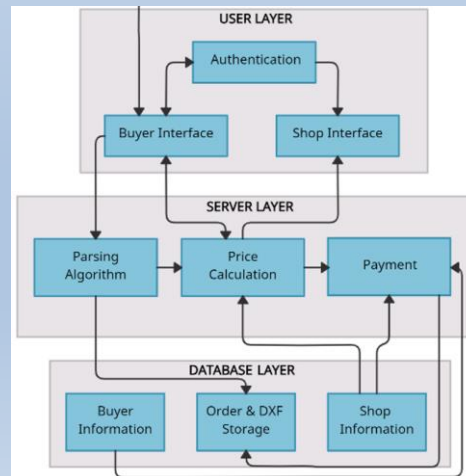


Figure 1. Dataflow Diagram

Experimental Results

In order to ensure consistency and functionality of our Quick Quote application, we ran 4 different type of tests on the DFX parsing algorithm using the drag and drop feature. Pictures of DXF files and their results are before:



Figure 2. DXF test 1

Result: Parsing FAILED
 Reason: Shapes cannot overlap or intersect.



Figure 3. DXF test 2

Result: Parsing FAILED
 Reason: If there are multiple shapes, they need to be nested within each other.



Figure 4. DXF test 3

Result: Parsing FAILED
 Reason: All loops should be closed.



Figure 5. DXF test 4

Result: Parsing SUCCEEDED
 Reason: Passed all requirements.

Conclusions

Even though we added all the key requirements, we ran out of time and were not able to implement the customer information side of the database. This requirement would have made it possible for customers to make accounts and have all their information auto filled when ordering. Storing that customer data would also help the Job shops know if a customer is new or returning, which can help them allocate resources where they see fit.

In the future, Quick-Quote also needs to create a product that allows companies to offer a competitive instant quote service. Offering quoting based off several file types will increase the market appeal for this product. Some file types to be considered: STEP (.stp, .step), STL (.stl), Parasolid, Autodesk Inventor (.ipt), and SOLIDWORKS (.sldprt).

References

What Is Amazon DynamoDB? - Amazon DynamoDB. (2018, July 3). Docs.Aws.Amazon.Com.
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>