End-Of-Studies Project

SawMill++

How to help small and medium-sized sawmills increase their efficiency while minimizing their losses and waste.

By LEMOINE Arthur, ALGOSUP (Vierzon, France), June 2024

Table Of Content

Glossary

Lumber:

Lumber is wood that has been processed into uniform and useful sizes (dimensional lumber), including beams and planks or boards. Lumber is mainly used for construction framing, as well as finishing (floors, wall panels, window frames).

Batten:

A batten is a long flat strip of squared lumber, small in width and height.

Wood Falls:

Wood falls define part of the log that is unusable as lumber after cutting mainly the rounded part of the log.

Sawdust:

Wood dust produced by the blade when cutting.

Wood Chips:

Wood chips are basically small parts of wood produced by the action of cutting, depending on the blade and method, wood chips can be bigger or smaller, or eventually just sawdust.

Cubic metre:

The cubic metre or cubic meter is the unit of volume in the International System of Units. Its symbol is m³. It is the volume of a cube with edges one metre in length.

Stere:

A stere is a measurement unit for wood quantities, it is remarkable by its changing way of calculations depending on the length of the wood logs, for instance for logs that are 1m long 1 stere = 1 cubic metre but if the wood logs are 50 cm long, then 1 stere = 0.8 cubic metre.

UI:

UI refers to the user interface. It is the interface of software through which the user can interact/communicate with the software.

UX:

UX refers to user experience. It is a broad term defining the experience of the user throughout the usage of software (It is usually related to UI).

Operating System

The operating system is the low-level software that supports a computer's most basic functions.

Unity

Unity is a game engine that comes with its own editor, facilitating the creation of software (usualy but not exclusively video-games)

C#

C#, pronounced "C-sharp," is an object-oriented programming language derived from C (C -> C++ -> C#)

Long Term Support (LTS)

Long Term Support describes a version of a software for which the support is guaranteed for longuer than usual.

CSV extension

A csv file is text document red as a table with values separated with comas (Coma Separated Values)

I. Introduction

1.1 Problem:

Small sawmills are less efficient than their bigger counterparts, partially due to the fact that they use smaller and usually manual machinery which is more sensitive to human errors. This leads to a loss of potential income as more cuts mean more energy consumed (along with tool durability) and can also lead to the loss of part of the log which could have been used as lumber.

Beside the financial losses this issue can cause, there is also an environmental issue as the unused wood will be further transformed (which means more energy costs) to be used elsewhere while it could have been used as construction wood or any other lumber.

1.2 Solution:

My solution would be a software capable of saying which is one of the most efficient methods to saw a particular wood log given the required lumber, the size of the log and other parameters described below. That software would be compatible with desktop machines but also with a mobile phones and tablets to be portable and useable directly on the field.

1.3 Target Audience:

My target audiences are small and medium sized sawmills that work with small and usually manual machines. Bigger sawmills already have similar software linked to heavy machinery and produced by the manufacturer.

1.4 The Software:

The software will take the following inputs and outputs:

- Inputs:
  - Log size (Length & Diameter[Foot & Head])
  - Wanted piece of lumber (type & size | example: Plank — 18x150mm)
  - Blade thickness (needed to calculate losses due to cutting and quantity of dust and chips produced)
  - Wood type (example: Oak, Spruce, …)
  - Wood density
  - Bark's thickeness
- Outputs:
  - Pieces of work possible with:
    - Size
    - Quantity

        - Possible selling price
    - Quantity of wood dust produced
    - Quantity of wood falls (bark and round sides)
    - Graph of possible lumber
    - Multiple choices of valuation (beside the required piece, you'll have the choice of what to do with the rest of the log)
    - Description of the procedure to saw the log according to the choice made (with schematics)

Additionaly the calculation should show solutions with the least possible saw cuts in order to be as efficient as possible to save energy and time.

II. Functional Specifications

2.1 Features

- Give the process to saw a specified lumber in a given wood log (schematics & measurements)
    - Give the quantity of Sawdust and wood chips produced in the process (in cubic metres)
    - Give the quantity of wood falls produced in the process (in cubic metres)
    - Optimise the process of sawing to reduce the number of saw cuts and the quantity of wood falls as much as possible
- Keep track of the quantity of wood falls and wood chips / Sawdust
    - Add
    - Delete
    - Modify
    - Display
- Stock Management
    - Add
    - Delete
    - Modify
    - Display
- Client's orders Management
    - Add order
    - Delete order
    - Modify order
    - Display order (with order fulfilment using comparison to stock)
    - Use client's orders to propose additional lumber^{[1](#lumber)} choices in the sawing
- Give statistics
    - Different usages of wood log in percentages
    - Different lumbers in stock
    - Different types of wood used/proposed
- Display "Alerts" for any stock (raw material, blades, wood falls, etc.) depending on a threshold set by the user.

- Parameters
  - Light / Dark mode
  - Language selection
    - French
    - English
  - Metric / Imperial system

2.2 The Optimisation Algorithm

The following algorithm is only one way of doing the calculation, it may not be the best nor the most efficient of all as it is impossible to take into account the infinite number of variables in the shape of a wood log. People sawing wood should always verify that the output of this algorithm is possible in the specific case of the log they are sawing.

Introduction:

To create an optimised and as accurate as possible algorithm, we need to understand how wood sawing works. It can be divided into four steps which represent the four sides of the log you're sawing. These steps can differ because there is more than one way to saw wood. We will limit ourselves to this specific method.

Step 1:
Your log is still round meaning you can't saw at precise lengths. You need to do a first cut which will be the first wood fall (or loss)
Depending on the log, you may need to do a second (and eventually more) cut to be able to later turn the log to a straight face (to have a 90° angle on your lumber) this (or theses) cut(-s) can be resewed later to avoid throwing them out and get more lumber out of the log

Step 2:
Now that you have a straight face on your log, you can rotate it 90° and do the same again until you have an angle with no remaining bark.

Step 3:
Now you can rotate the log 90° again and you can start to cut at precise measurements since you have a straight face below.
Cut lumber until you're left with the size of the last piece of lumber that you want (usually the bigger one)

Step 4:
Now you can rotate the log 90° one last time and cut your final sections until you have the piece you wanted

Exemple:

You want to make a piece of lumber^{[1](#lumber)} that is 150x150mm in a log of diameter 600mm

Step 1: You cut the first pieces until you reach about 75 to 80% of the log (depending on the log's shape)

Step 2: You rotate the log 90°. You cut until you have an angle with no bark left

Step 3: You rotate the log 90°. You have let's say 480mm, you'll need 150mm for the last piece you want, which leaves you with 330mm to work with to do other pieces (bark on top still remaining), you can divide it to see what you can do with it. In our example, we can do up to 15 planks of 18mm (remember to take into account the thickness of the blade in this example 4mm). Now you're left with the log 150mm high and still one side with bark (around 480mm)

Step 4: You rotate the log 90°. You have again 480mm in height so you do the same as before and your planks are now directly cut to their final form and you're left with your 150x150mm in the end

So in this example we were able to saw 1 150x150mm and up to 30 18x150mm in one log but with big losses in the corners.

The Algorithm:

The algorithm will take the eight following values as inputs:

- The log's length (in mm)
- The log's diameter at the foot (in mm)
- The log's diameter at the head (in mm)
- The wood type
- The height of the lumber we want (in mm)
- The width of the lumber we want (in mm)
- The thickness of the blade (in mm)
- The discovery plank height (in mm) alongside the table of wanted lumber (used to determine what to do with the remaining parts of the log).

It is important to note that the table of wanted lumber will be replaced as an input by clients' orders in the second version of the software.

The last parameter will be a checkbox in the case that the user want to do one and only one piece of lumber in the log (e.g. when the user wants battens he will do maybe 20 or 30 in one log doing only that). In that specific case we will just divide the log with the dimension of the wanted lumber repeatedly for the sides 3 and 4 instead of using a secondary lumber.

With the three first values we can represent the log as a cylinder.

Using the height and width of the wanted piece, we need to determine the procedure to saw that specific piece of lumber with the minimum number of cuts AND the least losses taking into account the thickness of the blade, the bark's thickness, the percentage of the log usable and the table of wanted lumber. The remaining wood which isn't usable will be saved as wood falls.

Using the wood density, the blade thickness and the number of cuts, we will determine the quantity of Sawdust produced by the operation.

As a reference, the log is always on even ground which means that if there is a difference between the two diameters, the two sides will be aligned on the floor as follows:

In order to have a high quality and straight lumber, we will do the calculation from the smaller side.

The wood type is needed to pass on the piece(-s) of lumber produced.

The output of the algorithm will be the following:

- The process of sawing with the cuts their number and their position (detailed steps)
- A graphic representation of the said process
- The total number of cuts
- The outputted lumber (list with size in mm and number)
- The Sawdust produced (in cubic metres)
- The wood falls produced (in cubic metres)

2.3 Stock & Client's Orders Management

Each stock will contain the following information:

- Name of the stock ("wood falls", "Sawdust", ...)
- Quantity (with the unit of measurement)

In the case of lumber, each different lumber will be defined by their:

- Width
- Height
- Length
- Quantity
- Wood type

In the case of client's orders, they will be defined by their:

- Name of the client
- List of Lumber ordered
- Date Ordered
- Delivery Date

2.4 Mobile & Desktop Application

The desktop application will contain every feature listed in this document and will be the main window for the user.

Regarding the mobile version, it will only contain the one core feature of V 1.0. The point of the mobile application is to have the possibility to have the optimised sawing process without the need for a computer. It will be a smaller version of software, portable and easier to use.

The two versions of the software may be able to communicate through a system of encrypted file exchange that could contain the following:

- Sawing procedures
- Stocks values
- Client's orders

Concerning the supported Operating systems, the software will run on Windows and MacOS for the desktop version, and on Android and iOS for the mobile version.

2.5 Version Planning

V 1.0:

The first version of the software will only contain the computer version and the following core feature: Give the process to saw a specified lumber in a given wood log (schematics & measurements)

V 2.0:

The second version of the software will contain the mobile app version of the software with improvements and bug fixes for the core feature of the computer version along with the following secondary features:

- Give statistics
- Keep track of the quantity of wood falls and wood chips / Sawdust
- Keep track of stock
- Keep track of client's orders

V 3.0 & +:

As of today, there are no plans for versions of the software above version 2

2.6 User Interface / Experience

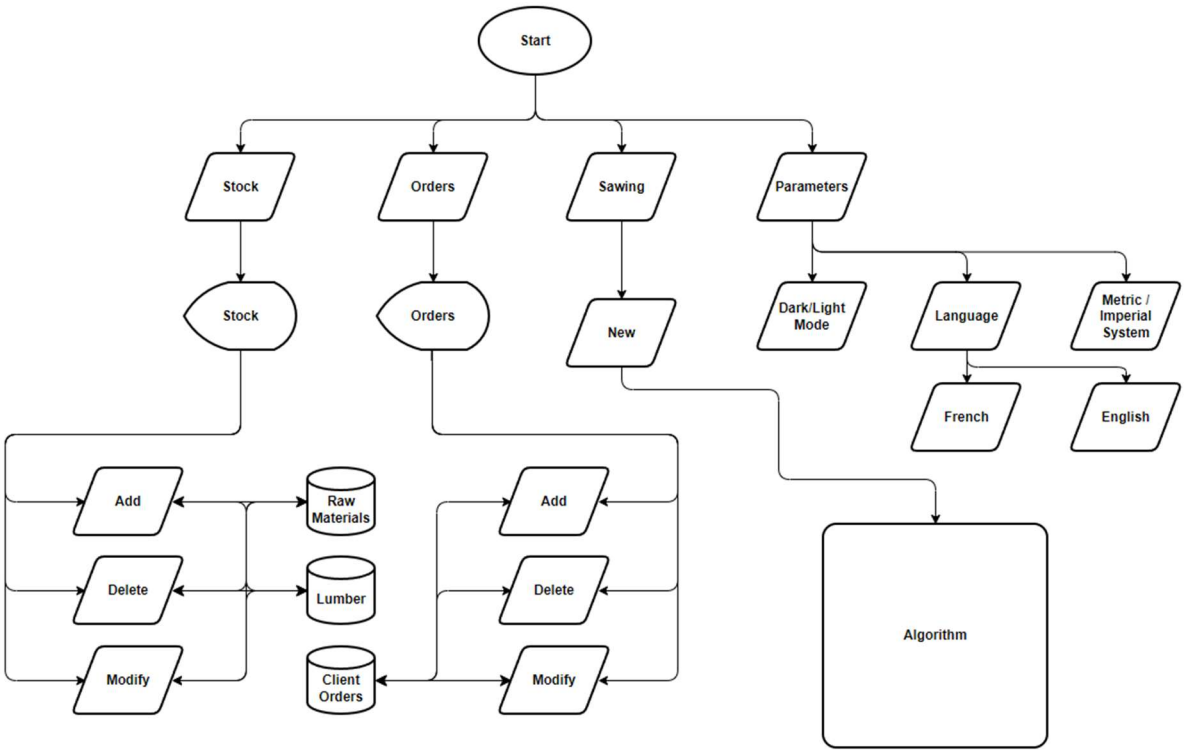
Interface

Every aspect of the UI is subject to changes as user surveys will be conducted and used to improve the design and the UX.

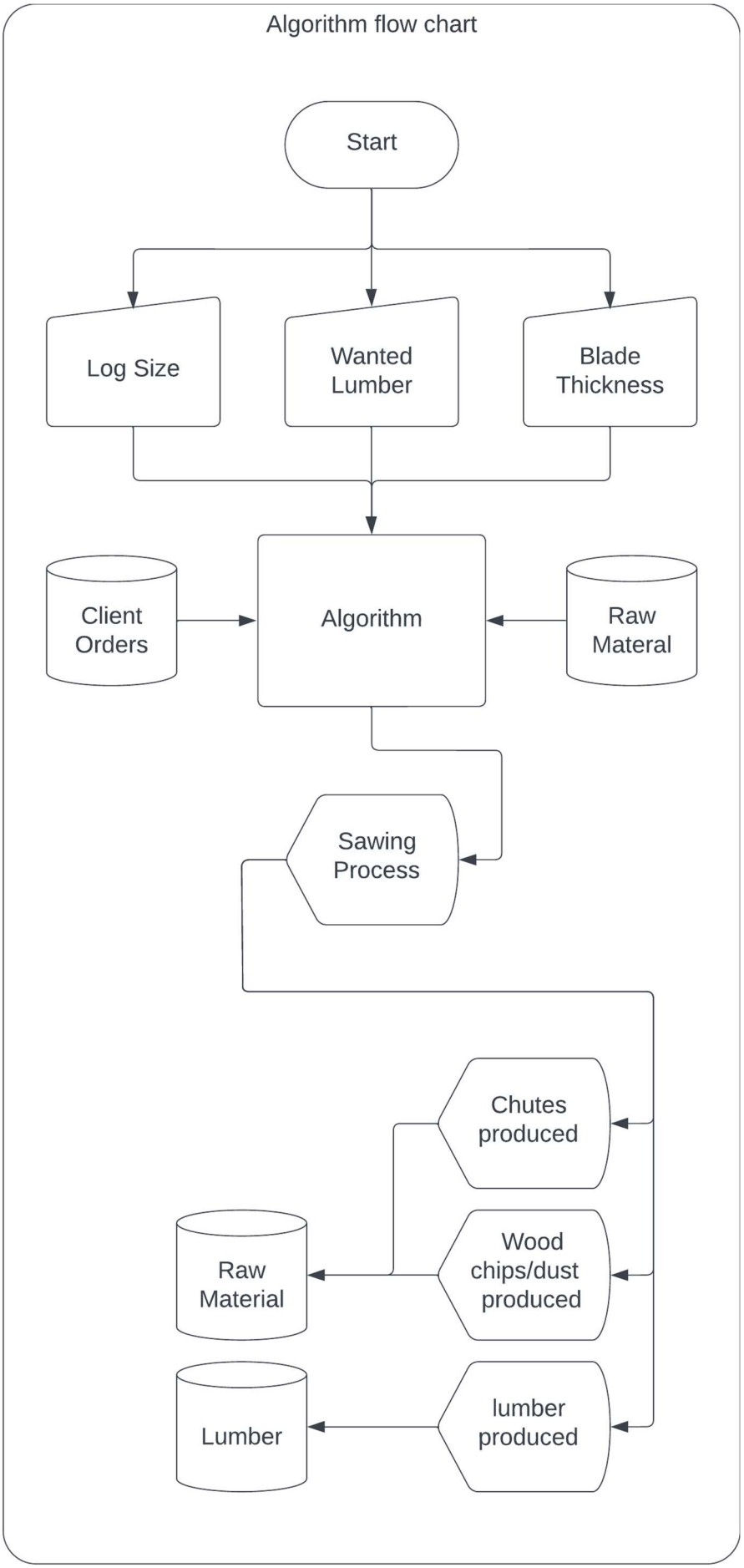
Needs:
- Overall aspect of the application (desktop and mobile version)
- Aspects of parameters
- Aspects of sawing process (outputted by the algorithm)
- Aspects of stocks
- Aspects of orders
- Aspects of the main screen

FlowCharts

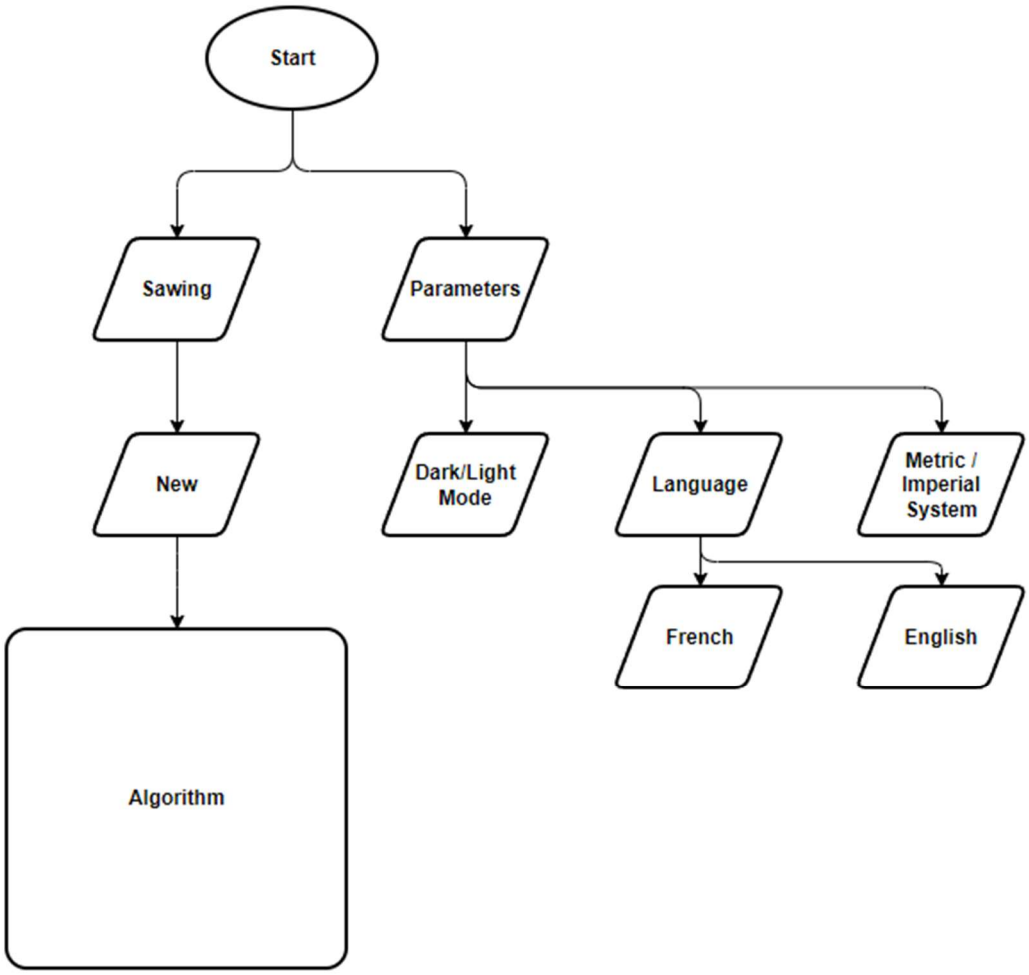
Desktop:



Algorithm:


Algorithm flow chart
Start
Log Size
Wanted Lumber
Blade Thickness
Client Orders
Algorithm
Raw Materal
Sawing Process
Chutes produced
Raw Material
Wood chips/dust produced
Lumber
lumber produced

Mobile:



2.7 Security

Regarding the security of the software, all data and calculations will be stored locally and will be encrypted to avoid any leaks. This means that the software won't need the internet and will be able to operate without access to a network.

In the event of a leak, the data stored are the following:

- Stocks
- Raw material
- lumber
- Sawdust
- wood falls
- Client's orders

Since the client's/customer's information will only be a name and the quantity of product ordered, the risk stays very low as those two pieces of information are of no use without further context.

Concerning the stocks, this information is of no use to anyone without the context of the company/sawmill owning the stock, and even with that specific context, that information is no threat to the company since their stock would most likely be visible by their potential clients or accessible through any contact to the company (phone, website, …).

2.8 Error Handling

The software will contain a solution for error handling that will, in defined and common cases, display the source of the error and tips to restore the software functionality so that the user can resolve the issue themself and avoid calling a technician.
Said error will include:
- Wrong inputs
- Known potential errors in the Algorithm

2.9 Alerts

This section is related to the stock management feature.

The software will display alerts when certain stocks get above or below the threshold set by the user. The user will have access to parameters to set what is considered a low (or high) threshold for each stock (a bigger sawmill may consider 20 blades as low when a smaller one will never have that much). When the stock gets above or below the parameter set by the user, the software will simply display an alert “Low blade stock” or “High wood-dust stock”.

The alerts will be an optional feature that the user can activate on specific stocks, represented as a check box on the stock management page. Alerts will be displayed on the main screen of the software.

III. Technical Specifications

3.1 Technical Requirements

To ensure that our software will be widely used, it will be compatible with the most common operating systems both on desktop computers and mobile phones.

Desktop

OS	OS Version
MacOs	14+
Windows	7+

Mobile

OS	OS Version
Andoid	13+
IOS	17+

2.2 Development Environment

Considering the size of this project and the timeframe in which it will be conducted, there will be two different environments in which it will be developed.

	Configuration 1	Configuration 2
OS	MacOS Sonoma 14.2.1	Windows 10
Processor	Apple M1	Intel i7 6700
RAM	8Go	16Go
Graphic Processor	Apple M1	Nvidia GTX 1660ti

2.3 Constraints

Our main constaint is time as the timeframe of the project maybe wide but as always we need to stay alert on deadlines and milestone to ensure good progression and that the delivery will be on time.

There are no specific constraints other than time on this project

2.4 Technologies

For this project, we will use Unity and C# to create our software. The choice came down to Unity because it allows us to easily integrate a User Interface and export the software in different formats compatible with our required operating systems (Computer or Mobile).

Furthermore, I personally have quite a bit of experience with Unity, thus avoiding the need to learn how to use it, saving time and allowing faster development.

We will use Unity version `2021.3.24f1` as it is an LTS (Long Term Support) version compatible with both MacOS and Windows.

We will use Visual Studio Code and Visual Studio Community as IDEs. Using whichever one is better for the specific operating system we are working on.

2.5 Conventions

Files:
folder structure:
- Folder for all scripts
- Folder for all UI
- File names in PascalCase

Coding:

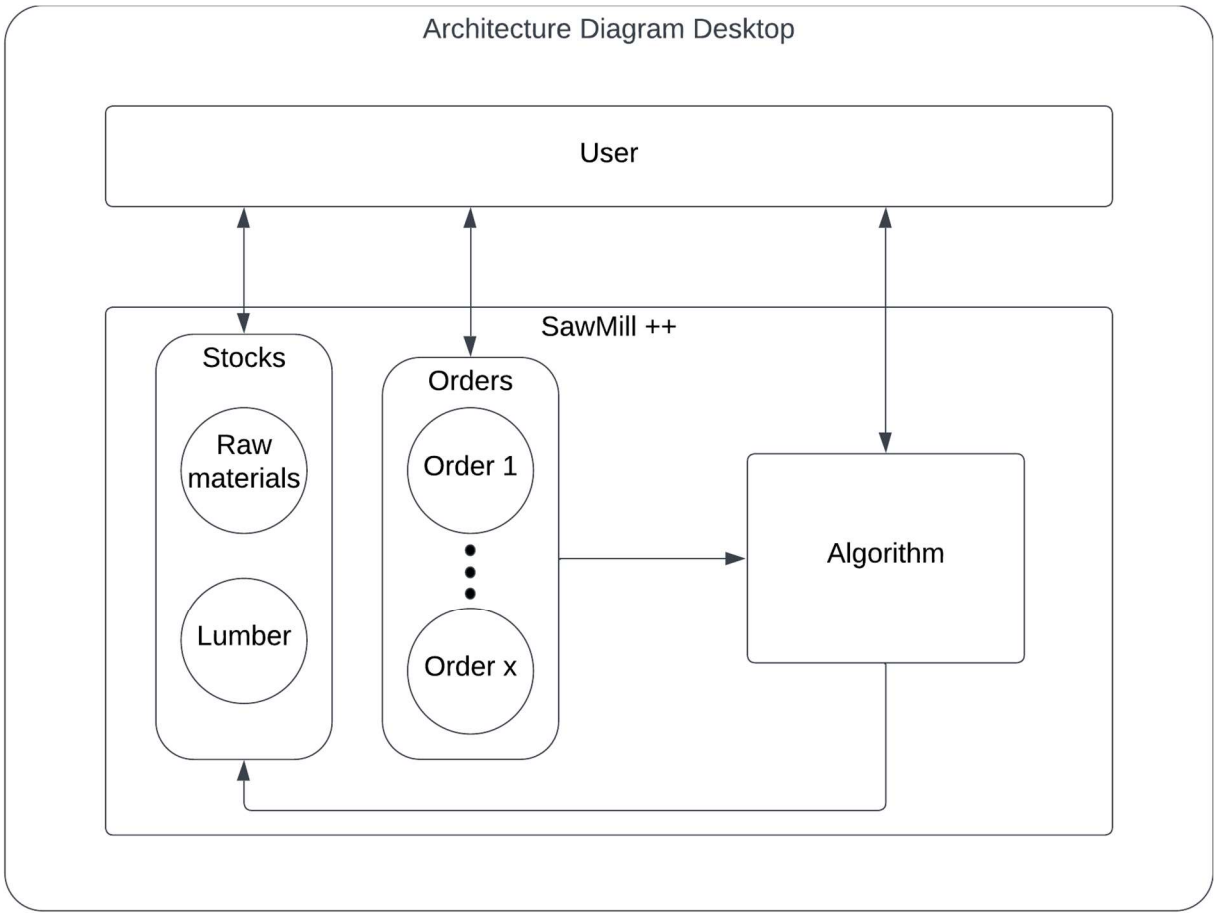
variables -> snake_case
Functions -> PascalCase

Comments:

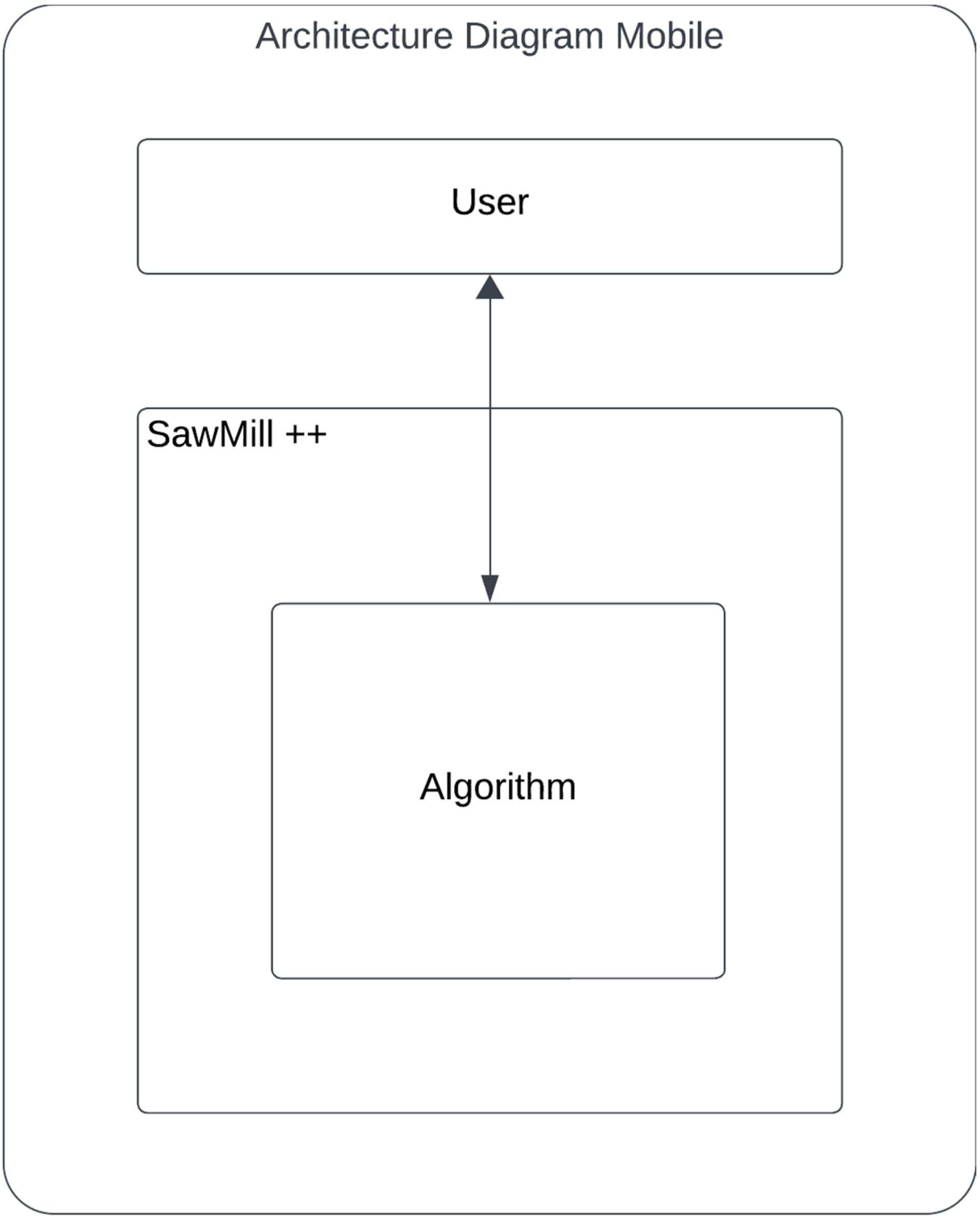
- Each function needs at least one comment explaining its purpose.
- Each complex formula must have a simplified version in comment along with purpose explaination.

2.6 Architecture Diagram

Desktop:



Mobile:



2.7 Algorithm

Classes:

Name	Attributes
Lumber	- Float: Width - Float: Height - Enum: WoodType
SawCut	- Vector2: PointA - Vector2: PointB

Inputs:

Name	Unit	Type of variable
Log's Length	mm	Float
Log's Diameter at the foor	mm	Float
Log's Diameter at the head	mm	Float
Wood Type	Name	Enum
Wanted Lumber Height	mm	Float
Wanted Lumber Width	mm	Float
Balde Thickness	mm	Float
Additional Lumber table	Table of Lumber	List<Lumber>
Discovery plank height	mm	Float
Do we repeat the wanted Lumber	Checkbix	Bool

Fonctionning

We will use a list of SawCuts (`List<SawCut>`) to store the result of the algorithm throughout its execution. The execution will be devided into the four sides of the log (as described in the functional specifications).

As stated in the functional specifications if the log's diameter at the foot and head are different we will use the smallest one.

- side 1: Starting from the top, we make saw cuts the height of the discovery plank until we reach 80% of the log's diameter.
- side 2: Starting from the right, we make saw cuts the width of the discovery plank until we reach the last cut from side 1.
- side 3:
    - We calculate what is remaining between the bottom (here bottom refers to the angle with the last cut from side 2, the round end at the bottom will be passed as discovery planks) and the last cut from side 1.
    - If the user wants to repeat the wanted lumber, then we divide the remains by its height.

    - Else we substract the height of the wanted lumber from the top (last cut from side 1) and divide by the best possible lumber height available in the additional lumber table. We also save the additional lumber used for next side.
- side 4:
    - We calculate what is remaining between the left (here left refers to the actual maximum left part of the log) and the last cut from side 2.
    - If the user wants to repeat the wanted lumber, then we divide the remains by its width.
    - Else:
        - We divide the part bellow the wanted lumber height by the width of the saved additional lumber.
        - We substract the width of the wanted lumber from the right (last cut from side 2) and divide by the best possible lumber width available in the additional lumber table (where the height is the same as the wanted lumber).

Finaly we determine how many of each additional lumber we were able to produce in that log along with the one wanted lumber. (each corner of the additional lumber needs to be in the "circle" representing the log for it to be considered complete and added to the count).

To determine the best possible lumber in the additional lumber table you look at what would remain after the division(on width or height). For the second version of the software the client's orders will also determine what additional lumber is needed.

- To calculate the saw dust produced, each saw cut can be represented as a 3D shape with those values:
    - Width -> Distance between the two points of the cut
    - Height -> blade thickness
    - Depth -> log's lenght

With this we calculate the quantity (volume in cubic meter) of saw dust produced.

To calculate the wood falls produced, we will calculate the total volume of the log and substract from it the volume of lumber sawed and saw dust produced.

Outputs:

Name	Unit	Type of variable
The sawing process	Graphical	List<SawCut>
The total number of cuts	Decimal number	Int
The outputed lumber	Lumber	Lumber
The outputed additional lumber	Lumber table	List<Tuple(Lumber, Int)
The produced wood falls	Cubic meter	Float
The produced sawdust	Cubic meter	Float

2.8 Stock / Clients' orders Management

To store the clients' orders we will use files as it is easier and faster to implement. We will also use a .csv file.

2.9 Error Handling

The software will be able to detect wrong inputs in text fields. In that case the software will display a window with the error and its cause while stoping the algorithm form running further calculations. The following errors will be handled:

Name	Cause	How to fix it
Wrong Input	An input is outside its range or in the wrong format	Change input
Algorithm failure	The algorithm has encountered a failure due to unexpected behaviour	Try again with a different input

New errors may be added in the future according to user testing and resturns.

2.10 Alerts

The alert system will work with a parameter that can be set by the user in the settings. Each stock type should have a corresponding alert. Once the stock value goes above or bellow the threeshold set by the user, an alert will be displayed on the main screen in the reserved area.

2.11 Further Considerations

Accessibility

The software will be available for mac and window device for desktop and for android and ios devices for mobile.

Risks & Assumptions

- We may have issues with time and deadlines in the future as this project is to be done on free time and this alone may not be enough.
- We may have issues with the algorithm, especially its complexity considerating the number of inputs and possibilities.

IV. Production Launch

4.1 Work

The work will be done on my free time and prioritized as follows:

1. algorithm
2. UI
3. Parameters
4. Stock managment
5. Alerts
6. Statistics

4.2 Milestones

There multiple milestones in this project, the main ones are the following:

1. Algorithm(simplified)
2. First Desktop Version
3. Algorithm(Advanced) working
4. First Mobile version

4.3 Deadlines

There are two deadlines for this project:

- Monday the 10th of june 2024 for the V1.0 and documentation
- End of august 2024 for the V2.0 and changes accodring to user feedback

V. Current and Future Changes

5.1 User feedback

As of today, we already have a few user feedback which are indicating a few missing features:

- A graphical reference for measurements on the sawing process
- A graphical reference to know which cut is the current one on the sawing process
- A way for the software to bring the virtual keybord on tactile devices (such as phones or tablets)
- A new parameter to add a “first cut” different from the discovery plank

There are still users that are testing and using our software on the field and new changes and ideas will come in the future.

5.2 Needed changes

With ideas to improve the software, the user feedbacks contains also a few bug reports. Those bugs needs to be fixed as soon as possible as some of them render the use of the software harder if not impossible.

The source code in its current state will also need a few updates for it to be compliant with the documentation of this project.

Another feedback was the current lack of a user manual, which will be a priority during the next phase of development.

VI. Conclusion

To conclude on this project, there are a few things I would do differently if I were to do it again. First, will try to be more consistent with my work, maybe allocating a specific evening in the week to the sole purpose of this project. Then, secondly, I would keep better track of where I am, creating a list of tasks and checking my progress (with the deadlines approaching, I rushed some aspects of the project, something that could have been avoided with a better organisation)

In the end, this project was a great opportunity to showcase my skills and try to push myself past my current limits while solving a real problem with which I was personally involved.