12SDD: Defining & Understanding and Planning & Design

Fengxi Zhang

Defining & Understanding

Solution Overview

Program Description
The program needs to be able intake coloured bitmap image and use linear vectorisation to create a linear black and white vectorised image. Then the program must be able to intake the previous linear vector image and turn it into a regular vector image. It must also be able to intake a black and white image and directly vectorise it into a regular vector image.

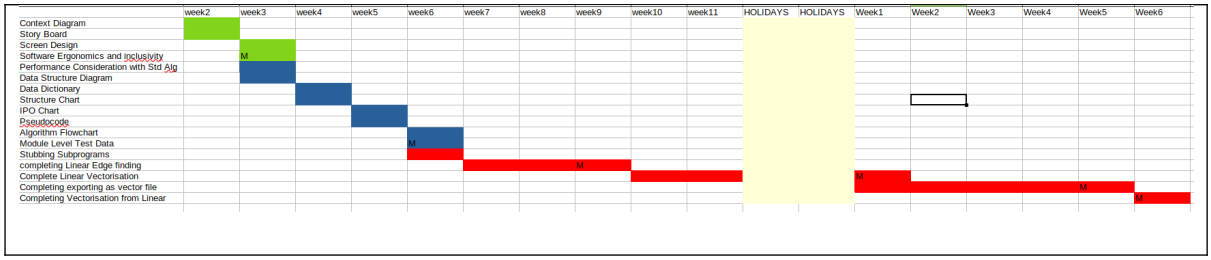
Functionality Requirements
Must be able to - Convert coloured bitmap images into black and white linear vector images - Convert black and white bitmap images into black and white vector images - Convert linear black and white vector images to black and white vector images -Accept different values for tolerence of what counts as a line

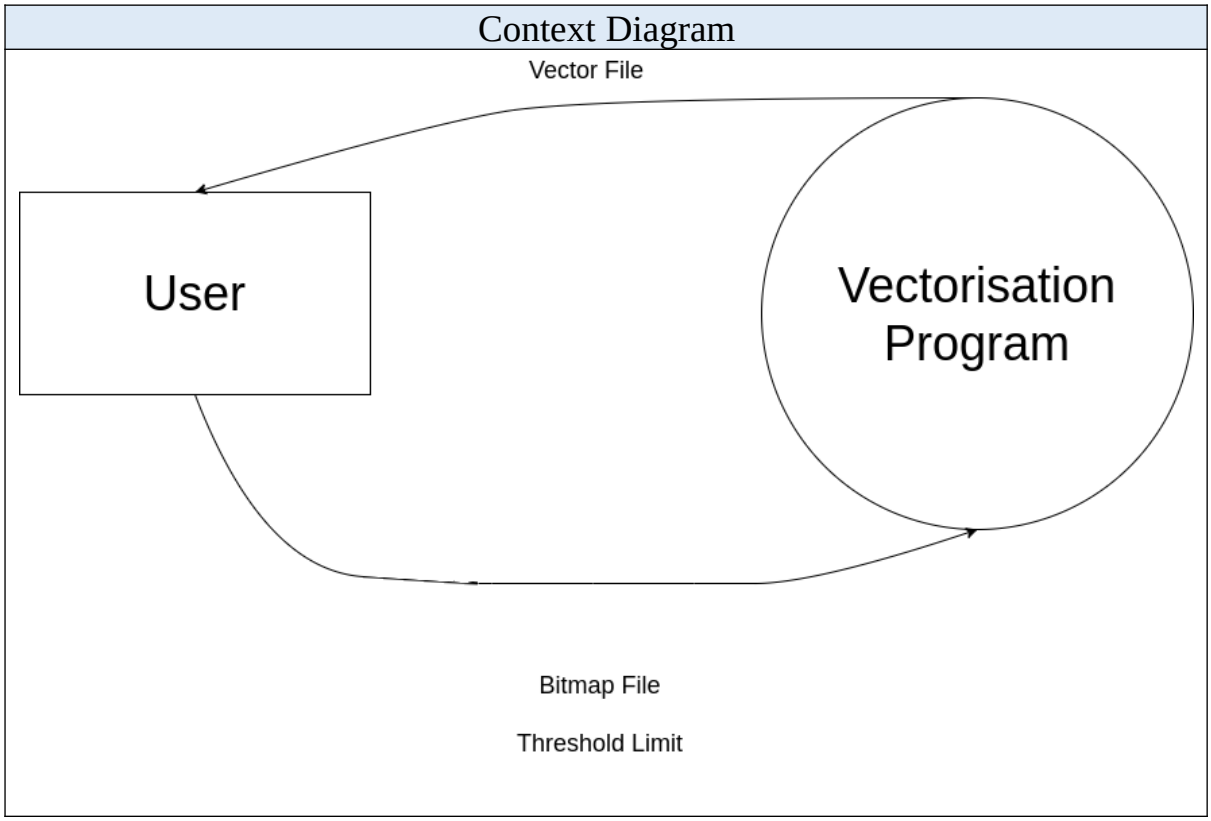
System Boundaries
Must be able to - Convert coloured bitmap images into black and white linear vector images - Convert black and white bitmap images into black and white vector images - Convert linear black and white vector images to black and white vector images - Run in the background as other processes occur Won’t be able to - Intake and recognise future file formats - Vectorise low quality images - Intake files not in the bitmap format

Planning & Design

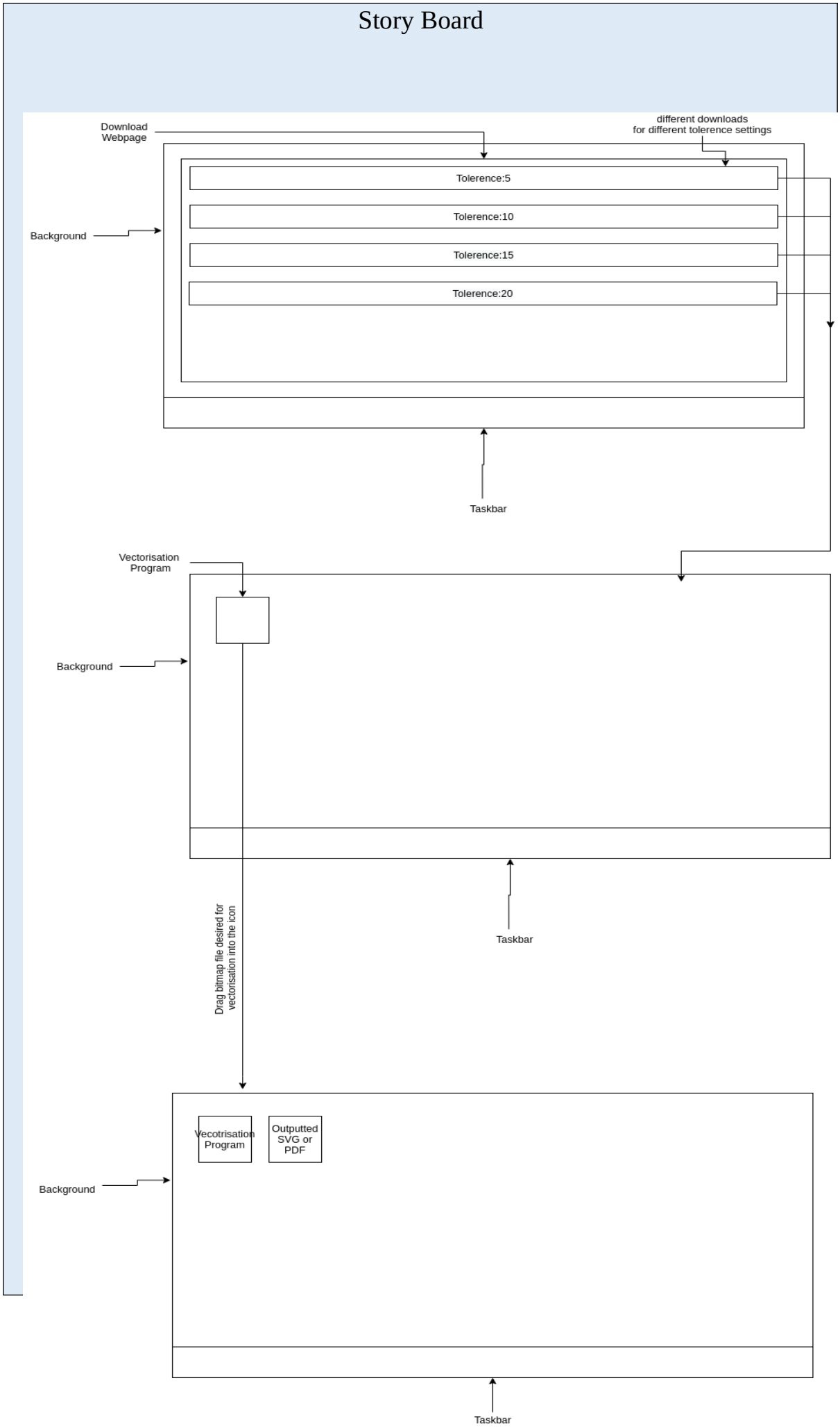
General Planning & Design

Gantt Chart

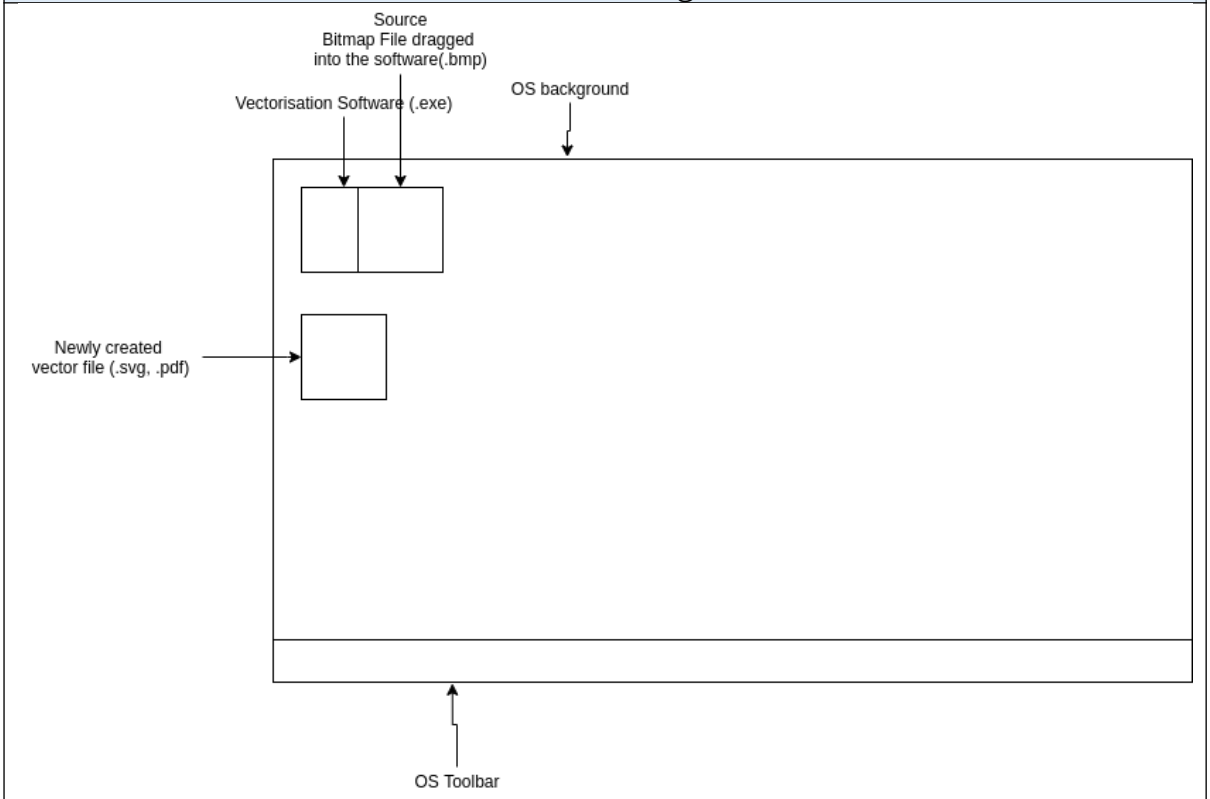

week2
week3
week4
week5
week6
week7
week8
week9
week10
week11
HOLIDAYS
HOLIDAYS
Week1
Week2
Week3
Week4
Week5
Week6
Context Diagram
Story Board
Screen Design
Software Ergonomics and inclusivity
Performance Consideration with Std Alg
Data Structure Diagram
Data Dictionary
Structure Chart
IPO Chart
Pseudocode
Algorithm Flowchart
Module Level Test Data
Stubbing Subprograms
completing Linear Edge finding
Complete Linear Vectorisation
Completing exporting as vector file
Completing Vectorisation from Linear
M


Context Diagram
Vector File
User
Vectorisation Program
Bitmap File
Threshold Limit

Story Board


Download
Webpage
different downloads
for different tolerence settings
Tolerence:5
Tolerence:10
Tolerence:15
Tolerence:20
Background
Taskbar
Vectorisation
Program
Background
Taskbar
Drag bitmap file desired for
vectorisation into the icon
Vecotrisation
Program
Outputted
SVG or
PDF
Background
Taskbar

Screen Design



Software Ergonomics & Inclusivity

The software will feature just a desktop icon with different downloads for different tolerences.

To use the user will drag a file into the icon and the result will be added on the desktop

This without many requirements for instructions and documentation with the required instructions able to be conveyed through images allows the greatest inclusivity in terms of language.

Further the program will be able to run in the background allowing other productive tasks to be done increasing ergonomics.

Ergonomics can also be increase through the design not requiring many actions from the user.

Structural Planning & Design

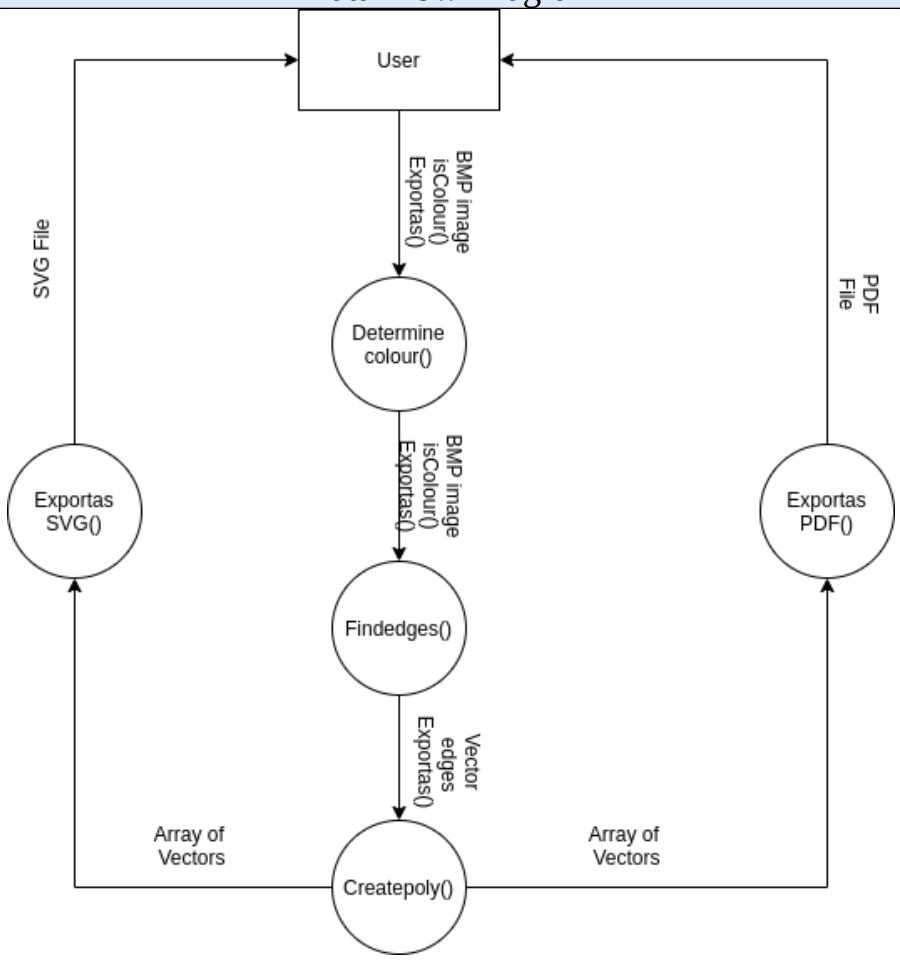
Performance Considerations with Standard Algorithms

The algorithm requires the checking of each 2x2 square that can be formed in the array of pixels that makes up the image as part of the edge tracing process. Further this array cannot be sorted as that will jumble all the pixels of the image into strips of colour, meaning that algorithms such as binary search cannot be used. This results in an efficiency of approximately O(2n) as each pixel must be analysed in the edge seeking process and then again in the exporting as a vector.

Data Structure Discussion

The main data manipulation will occur during the process of edge finding where each 2x2 adjacent square must be looped through and analysed to see if an edge exists. To find the edge random points between pixels will be chosen until a black pixel is adjacent to it. Then the line will keep the black pixel on its left and make turns accordingly. It will be easiest to use 2d array to arrange the pixels as the array will save the location of the pixels. The pixels can then be looped though using a nested “for” loop and compare whether a black pixel is on the left using the position of the pixel in the 2d array. Later the edges can be transformed into vectors using bezier curves and polygons the equations of which will be stored using 2d arrays

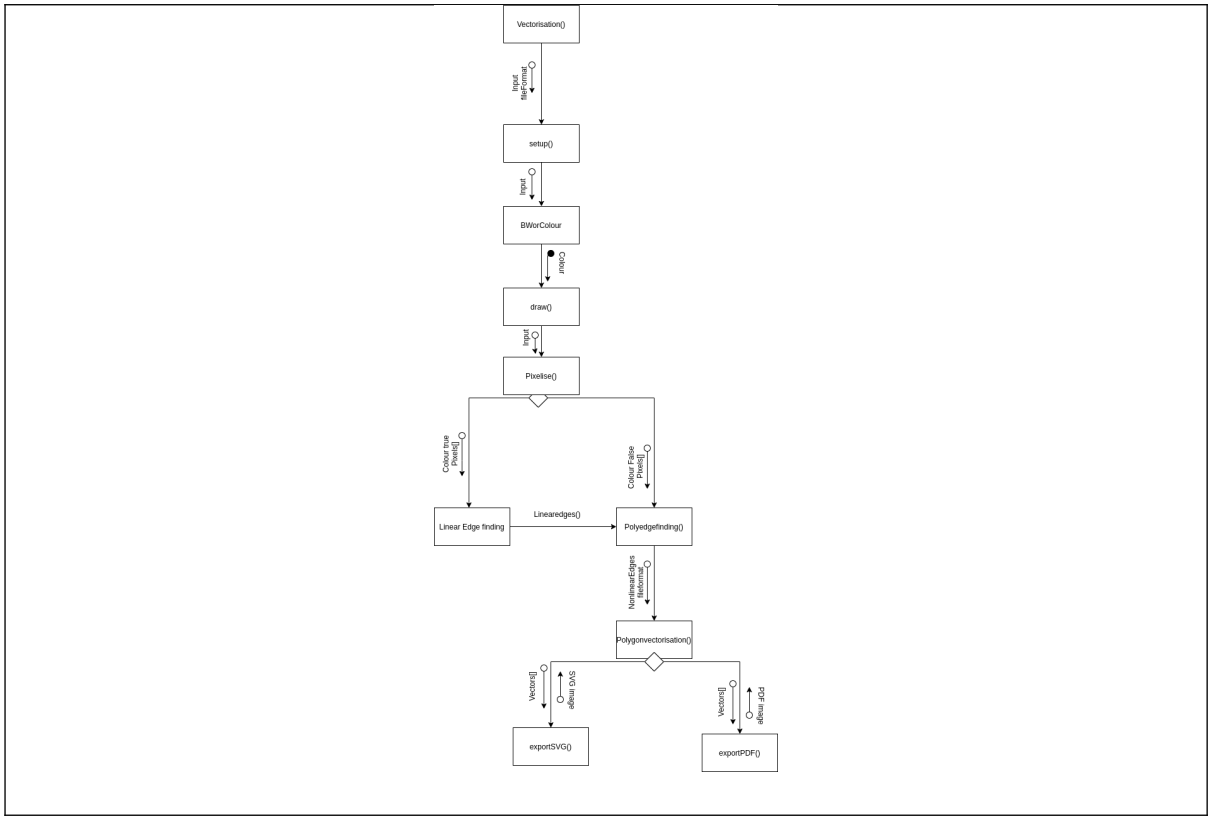
Data Flow Diagram



Data Dictionary

Name	Data Type	Scope	Description	Example
Input	Bitmap	Global	The raster file inputted by the user	Joe.bmp
Pixels[][]	Array[][] of integers	Global	Array[][] of pixels converted from the input. May be black and white or colour	[[233,3,234,4,34,21,45], [45,34,5,76,65]]
Colour	Boolean	Global	Records whether the input is coloured as different processes must be taken for coloured an black and white images	True
Threshhold	Integer	Global	Preset threshold at which a line is considered line, uses the integer darkness of the pixel	34
Vector[]	Array[] of equations	Global	An array of vector equations saved after the image has been vectorised	[2x +7]
Export	PDF or SVG	Global	A vector image of the same name as the original image. The finish produce of the porcess	Joe.svg or Joe.pdf

Structure Chart



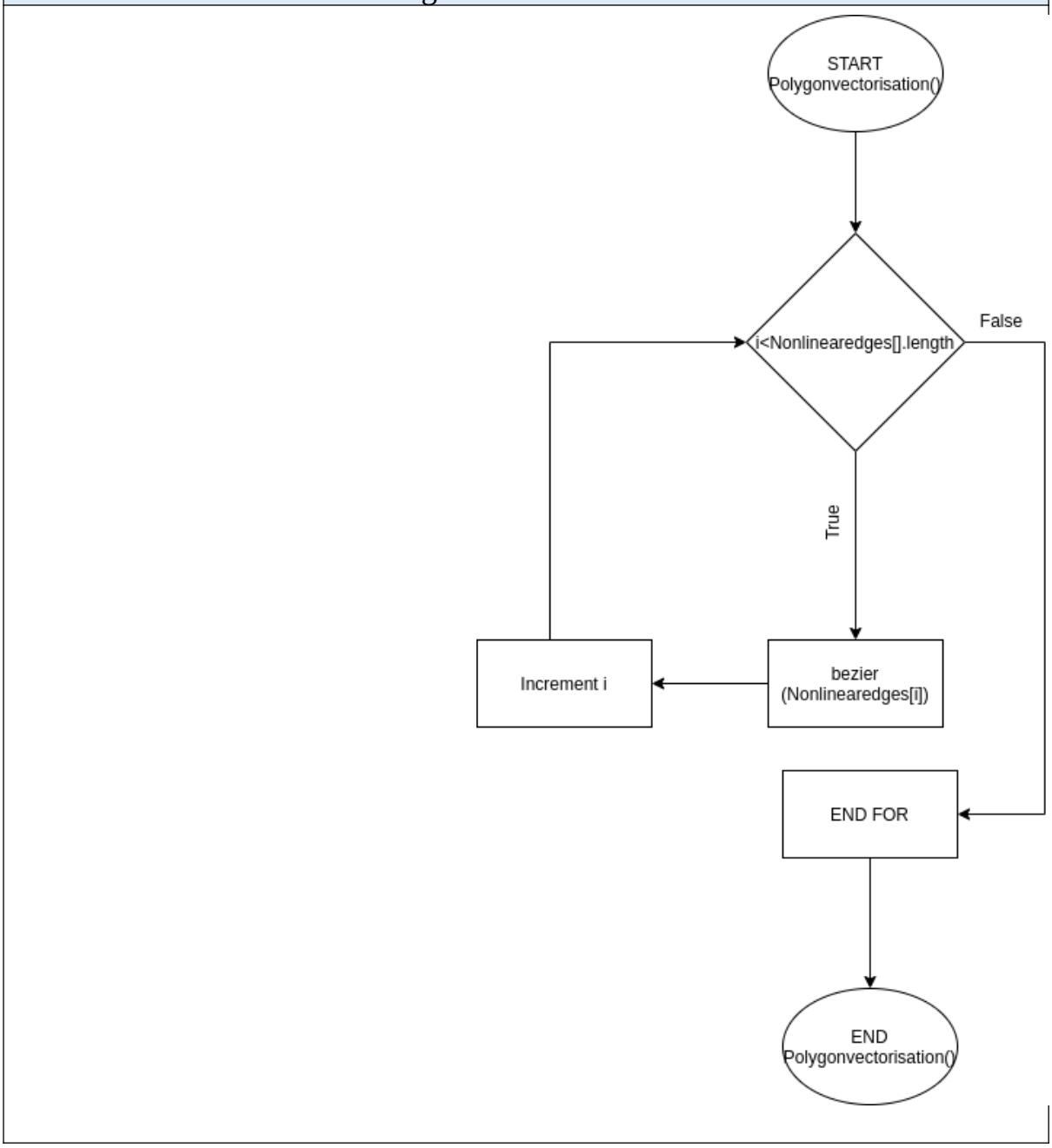
Algorithm Planning & Design

Polygonvectorisation()

IPO Chart		
Inputs	Processes	Outputs
Nonlinearedges[][] fileformat	Use Bezier functions with the inputs of the vertexes of the edges to smooth the staircasing of the image	Polygon equations of the traced edges

Pseudocode
START Polygonvectorisation(Nonlinearedges, fileformat) ----- FOR i=0 to Nonlinearedges[].length ----------bezier(Nonlinearedges[i]) -----END FOR END Polygonvectorisation()

Algorithm Flowchart



Module-Level Test Data			
Test	Expected Output	Actual Output	Reason for Inclusion
[0.5:3,4:3,5:4]	Bezier curve with anchor points 0.5:3 and 5:4 and control points 4:3	Bezier curve with anchor points 0.5:3 and 5:4 and control points 4:3	Tests if array and floats can be inputted into the bezier function
[-4:3, 3:4, -5:3]	Bezier curve that goes off the screen	Bezier curve that goes off the screen	Tests the effects of negative values
[120000003:34222; 233334:234552, 23452:2345234]	Overflow Error	Overflow Error	Tests what happens if an error occurs in previous subprograms resulting in extremely large values
[0.00000123:0.00000000245, 0.000000234:0.234235]	Extremely small bezier curve	Extremely small bezier curve	Tests what happens if an error occurs in previous subprograms resulting in extremely small values