

# **Software Requirements Specification**

For

Minor 1 Project - AlgoViz

B.Tech Computer Science Engineering

Semester V

Submitted by

<b>Specialization</b>	<b>SAP ID</b>	<b>Name</b>
DevOps	500076402	Pranit Gupta
DevOps	500076447	Sarthak Srivastava
OSS	500075837	Khushboo Jakhmola
OSS	500069562	Priyanshu Gautam

Department of Cybernetics

School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY  
STUDIES, DEHRADUN- 248007. Uttarakhand

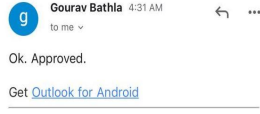
# Table of Contents

Topic		Page No
Table of Content		1
Revision History		4
1	Introduction	5
	1.1 Purpose of the Project	5
	1.2 Target Beneficiary	5
	1.3 Project Scope	5
	1.4 References	6
2	Project Description	7
	2.1 Reference Algorithm	7
	2.2 Data/ Data structure	7
	2.3 SWOT Analysis	8

	2.4 Project Features	8
	2.5 User Classes and Characteristics	9
	2.6 Design and Implementation Constraints	9
	2.7 Design diagrams	10
	2.8 Assumption and Dependencies	14
3	System Requirements	14
	3.1 User Interface	15
	3.2 Software Interface	
	3.3 Database Interface	
	3.4 Protocols	
4	Non-functional Requirements	15.
	4.1 Performance requirements	15.

	4.2 Security requirements	
	4.3 Software Quality Attributes	16.
5	Other Requirements	
Appendix A: Glossary		
Appendix B: Analysis Model		
Appendix C: Issues List		

# Revision History

Date	Change	Reason for Changes	Mentor Signature
28-10-2021	Addition of OpenGL library	For better graphics	 <p>Gourav Bathla 4:31 AM to me ▾ ← ... Ok. Approved. Get <a href="#">Outlook for Android</a></p>

# **1. INTRODUCTION**

We are now in a digitally advanced age where visual content plays a crucial role in every part of life and more than 65 percent\* of the population worldwide are visual learners. Algorithms are the core operational unit when it comes to approaching and solving any problem that requires efficiency as well as accuracy. Through the implementation of this project, we aim to assist in the ease of understandability of various algorithms. The user will get access to a Voice Over Video in the format of a mp4 file which is customized in order for them to understand and these mp4 files will be available in two different languages (Hindi/English) which the user can toggle between. We plan to use the Simple DirectMedia Layer (SDL) library which is internally written in C language and SFML library.

## **1.1 Purpose of the Project**

The purpose of this project is to provide novice learners an easy way to understand and implement algorithms easily and time efficiently. With access to a large variety of content online students have become more receptive to the visual content and which has in turn affected their attention span. Providing graphical representation for better understanding and quick learning is the objective of our project. We will be providing voice over videos in languages like English and Hindi which are the widely understood languages.

## **1.2 Target Beneficiary**

In this Project, the target groups are those students(learners) who have difficulty in understanding certain algorithms with the help of written notes or material available online. This project will give them access to Voice Over Videos along with the graphical representations for some algorithms making it way easier to understand and learn. This project will target the novice learners who don't have access to such videos or have difficulty in understanding as it is not available in their native languages.

## **1.3 Project Scope**

Through the implementation of this project, we aim to assist in the ease of understandability of various algorithms. The user will get access to a Voice Over Video in the format of a mp4 file which is customized in order for them to understand and these mp4 files will be available in two different languages (Hindi/English) which the user can toggle between. We plan to use the Simple DirectMedia Layer (SDL) library which is internally written in C language and SFML library.

The main objective of this project is :

- To provide step wise visual representation of algorithms.
- To implement Voice over Video for students in order to understand better and generate a mp4 file of the same.
- To provide Voice over Video in two different languages namely English and Hindi so that the students can learn in their preferred language.
- To create an enriching learning experience for the novice learners

## 1.4 References

- [1] [rohanakar/algorithm-visualizer](https://github.com/rohanakar/algorithm-visualizer) <https://github.com/rohanakar/algorithm-visualizer>
- [2] Singh, A. K., Jamal, D., & Aggarwal, P. (2021). *Algorithm Visualizer* (No. 5385). EasyChair.
- [3] Jain, H., Sharma, S., Arora, P., & Kumar, K. (2021). Howsort: Algorithm Visualizer Application. *Available at SSRN 3849936*.
- [4] Röbling, G., & Freisleben, B. (2002). ANIMAL: A system for supporting multiple roles in algorithm animation. *Journal of Visual Languages & Computing*, 13(3), 341-354.
- [5] [Simple DirectMedia Layer - Wikipedia](#)
- [6] [algorithm-visualizer.org](https://algorithm-visualizer.org)
- [7] S. Amershi, G. Carenini, C. Conati, A. K. Mackworth and D. Poole, "Pedagogy and usability in interactive algorithm visualizations: Designing and evaluating CIspace," in *Interacting with Computers*, vol. 20, no. 1, pp. 64-96, Jan. 2008, doi: 10.1016/j.intcom.2007.08.003.
- [8] [Sort Visualizer](https://www.sortvisualizer.com/) [<https://www.sortvisualizer.com/>]

## 2. PROJECT DESCRIPTION

### 2.1 Reference Algorithm

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

#### Algorithm

To sort an array of size  $n$  in ascending order:

- 1: Iterate from  $arr[1]$  to  $arr[n]$  over the array.
- 2: Compare the current element (key) to its predecessor.
- 3: If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.



Time Complexity:  $O(n^2)$

Auxiliary Space:  $O(1)$

Boundary Cases: Insertion sort takes maximum time to sort if elements are sorted in reverse order. And it takes minimum time (Order of  $n$ ) when elements are already sorted.

Algorithmic Paradigm: Incremental Approach

Sorting In Place: Yes

Stable: Yes

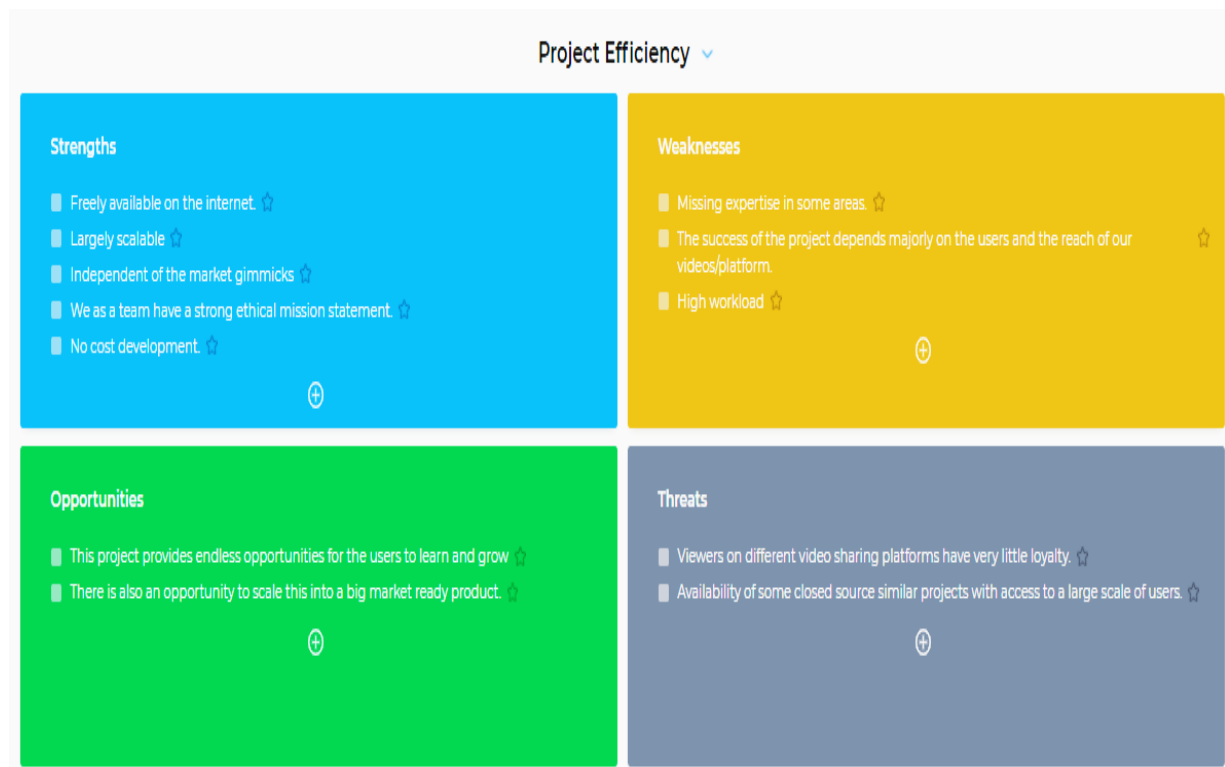
Online: Yes

Uses: Insertion sort is used when the number of elements is small. It can also be useful when the input array is almost sorted, only a few elements are misplaced in a complete big array.

## 2.2 Characteristics of Data

The Dataset we will be using is a predefined dataset which is an array set. It will be considered as input in the code and the output visualization or mp4 file will be made according to that dataset or array set that will be defined in project source code.

## 2.3 SWOT Analysis



## 2.4 Project Features

The major project features include:-

- Selecting algorithm for visualization
- Selecting an mp4 file which will have Voice Over Video content in order to explain each step of the algorithm during the visualization.
- Also the user will have an option in order to request from us the mp4 file and save it for future use.

## 2.5 User Classes and Characteristics

In this project, there are two user classes which are User and Developer.

User - This user class will have access to the mp4 files and the visualization project source code which they can execute in their respective machines or compilers after downloading the required library and having mentioned requirements but they cannot change the source code to the main repository. They will also have a choice in selecting languages which are Hindi and English.

Developer - This user class will have access to the source code. They can change the source code such as adding more algorithms or improving visualization or mp4 files quality and audio. User issues will also be addressed from this user class.

## 2.6 DESIGN AND IMPLEMENTATION CONSTRAINTS

In this Project, the basic requirements for project to be successfully executed in users machines are -

### **Hardware:**

Windows - Processor - 4 x 2.42GHz / RAM - 4GB

HDD - 40GB(free space)

MacOSX - Processor - 8 x 2.3GHz / RAM - 16GB

SSD - 20GB(free space)

Platform used for this project is different for Windows and Macintosh users since the compilers -

Ubuntu - This will be the OS we will be using for running and implementing our code.

Git and GitHub - This will be used for management of code using open source and source code management.

Xcode - This is a compiler for mac users.

Each user has to install SFML libraries in their compilers or terminal in order to successfully execute the source code. Also they need to configure their audio connections so that the Voice Over Video functionality works perfectly and they learn easily and clearly.

Source Code will be written in Cpp language using SFML & OpenGL library used for visualization and graphical representation. SFML & OpenGL are visualising libraries used in C and C++. It can also be used to add audio to projects one develops.

Other libraries that can be used for execution of project are :-

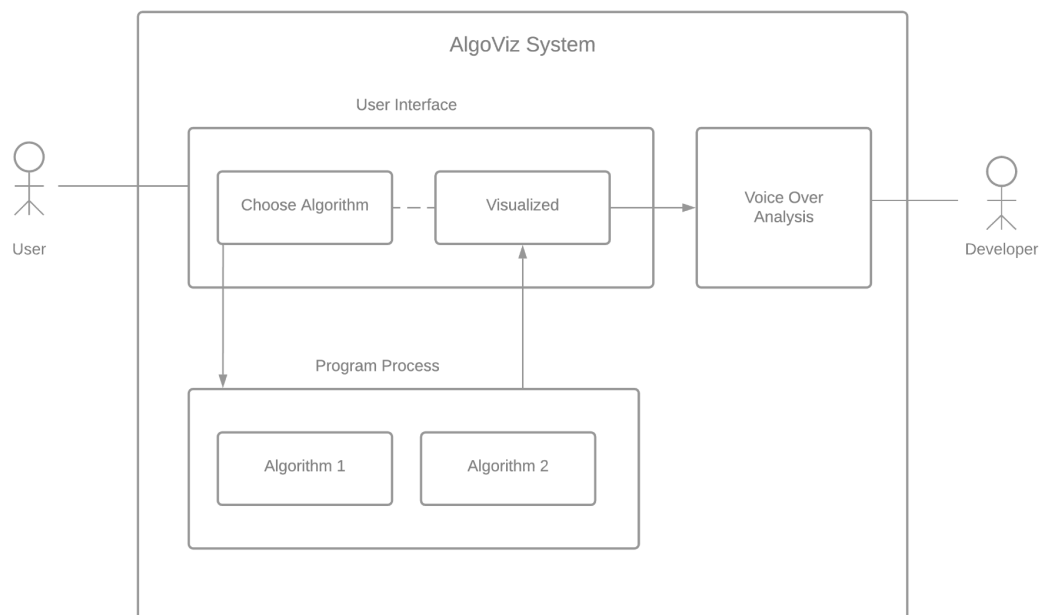
SDL2 - SDL2 Library which is a better and updated version than graphics.h for visualizing the algorithm that the user will opt to learn

SDL – SDL Library is a combined library with SDL2 for visualizing in C and C++.

OpenGL - OpenGL Library is used for rendering 2D and 3D vector graphics.

## 2.7 DESIGN DIAGRAMS

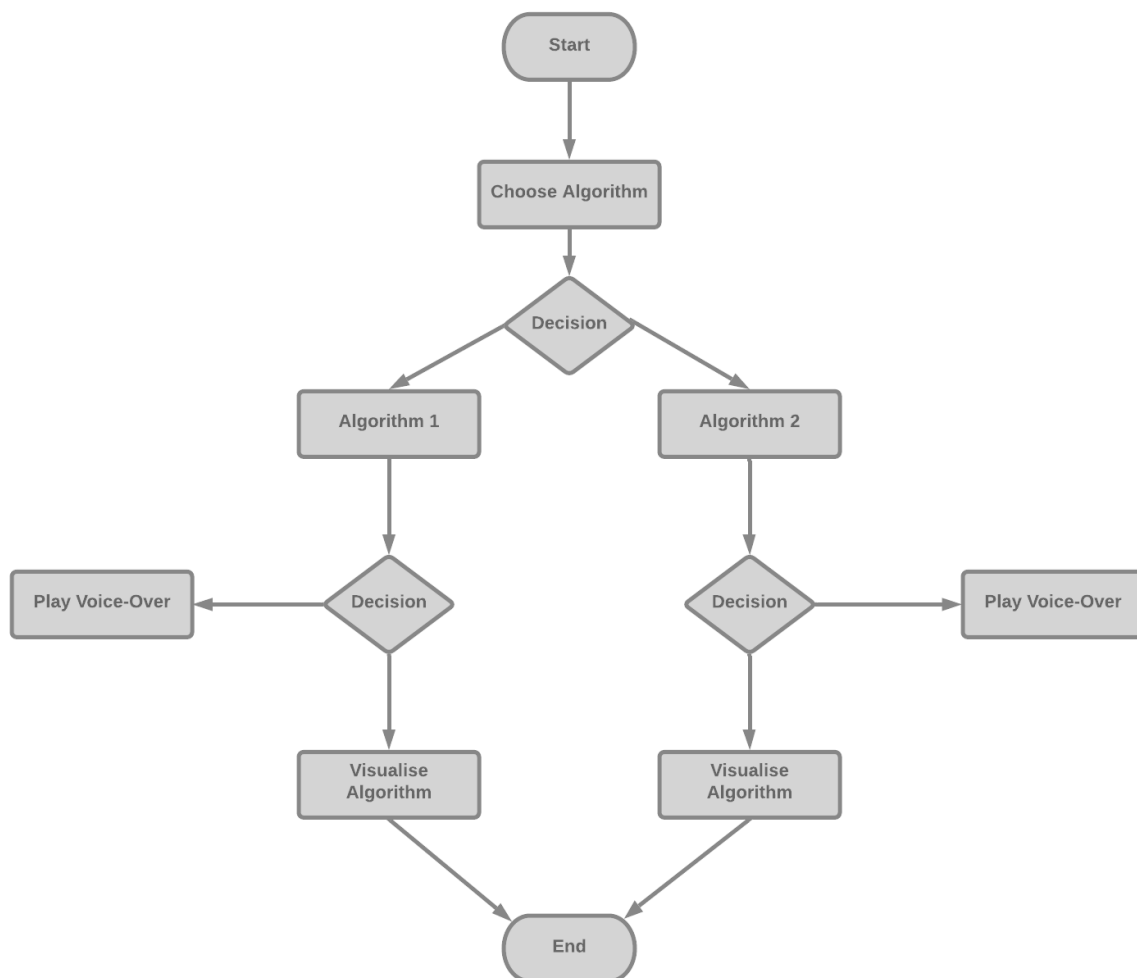
### USE-CASE DIAGRAM



In Fig.1, the USE-Case Diagram of our project is depicted. In this Project, we provide the students a choice to choose the algorithm they want to study and visualize. After the user chooses the algorithm the particular function runs in order to visualize the algorithm in an iterative manner or step-by-step manner. Also the user will be provided with a Voice Over Video option wherein each step of the algorithm will be explained and the user can choose to learn in their preferred language.

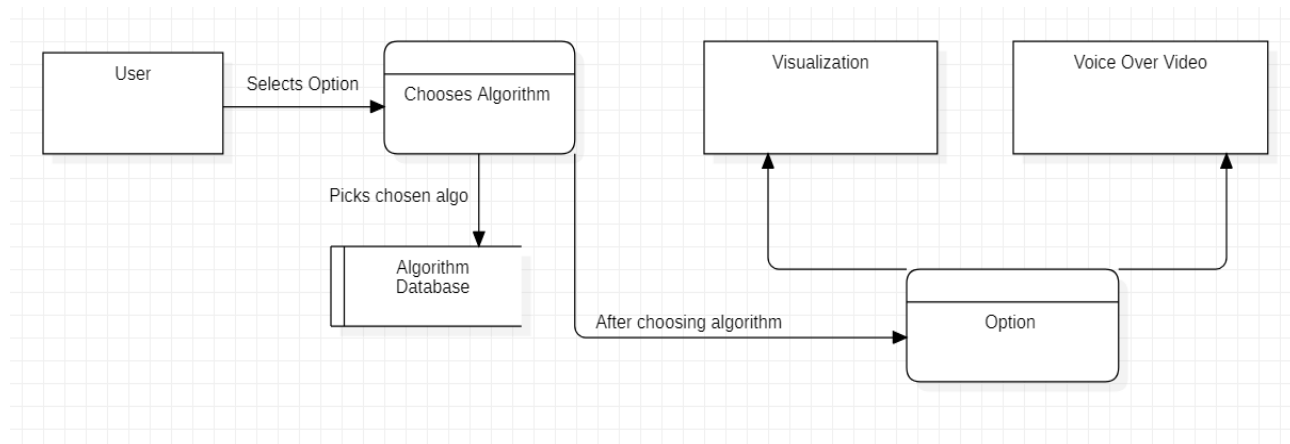
The functions in this project will be the same name as the algorithm which we will be implementing such as Algorithm1(), Algorithm2(),...,AlgorithmN(). In the initial part of the project we will define the visual setup such as nodes or bars which will have fixed values and hence explain the algorithm accordingly. The other functions will include the display function, size function, iteration function in order to alter the visualizing and speed at which the user wants to learn the algorithm. We will be using the SDL\_Renderer property to control the speed of the algorithm. Finally, in the main() function there will be function calling and in the compilation time the user needs to enter his/her choice as per the options mentioned.

## ACTIVITY DIAGRAM



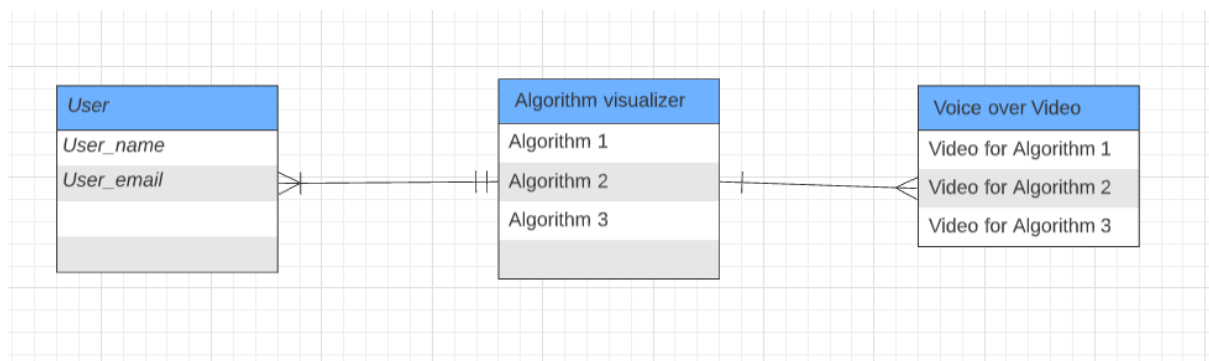
In Fig.2, the activity diagram of our project is depicted. In this diagram it is simply explained the workflow of our project. Starting is done by selecting the algorithm the user would like to study. Then after selecting the algorithm, the user has two options either to visualize the algorithm or to see the Voice Over Video(mp4 file) with each step explained in either Hindi or English language. After the visualization or the mp4 file is played and the user understands the algorithm window closes and then you can select another algorithm from the list mentioned to study and improve your algorithm and DSA skills.

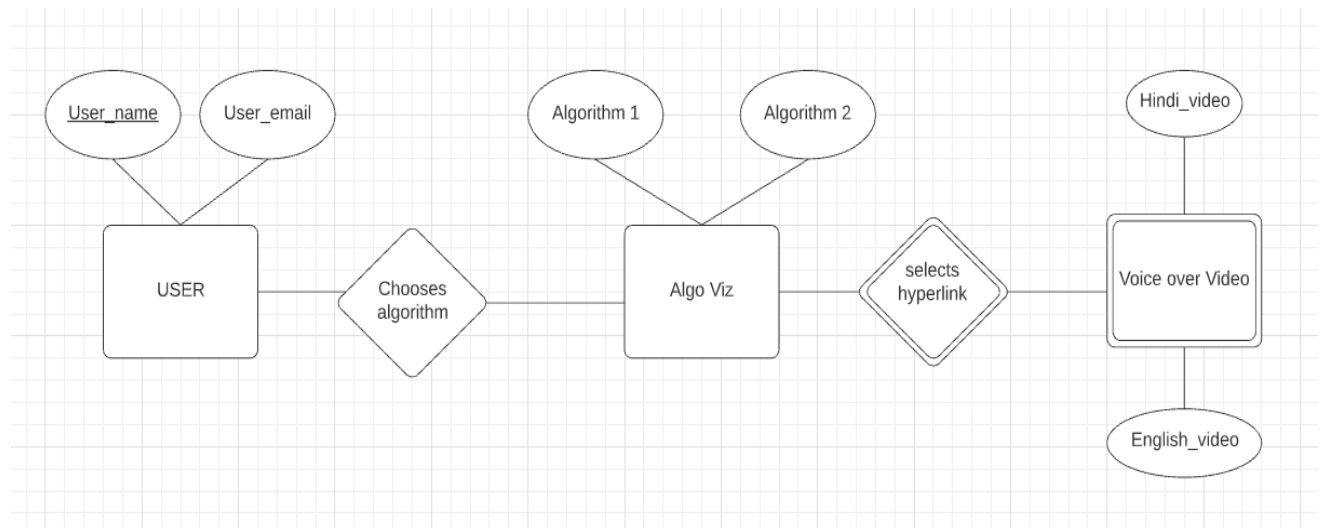
## DATA FLOW DIAGRAM



In fig.3, the Data Flow Diagram of our project is depicted. In this diagram the user will first have an option to choose the algorithm which they need to visualize or get a Voice Over Video from the algorithms that we have stored in our project memory or algorithm database for the code to fetch the algorithm selected and then ask for further functioning. After selecting the algorithm the user will have another to choose from the two options whether they want to visualize the algorithm or get a Voice Over Video. After the selected option and completion of the task the code will end and the user can re-run the code for more algorithms.

## ER DIAGRAM

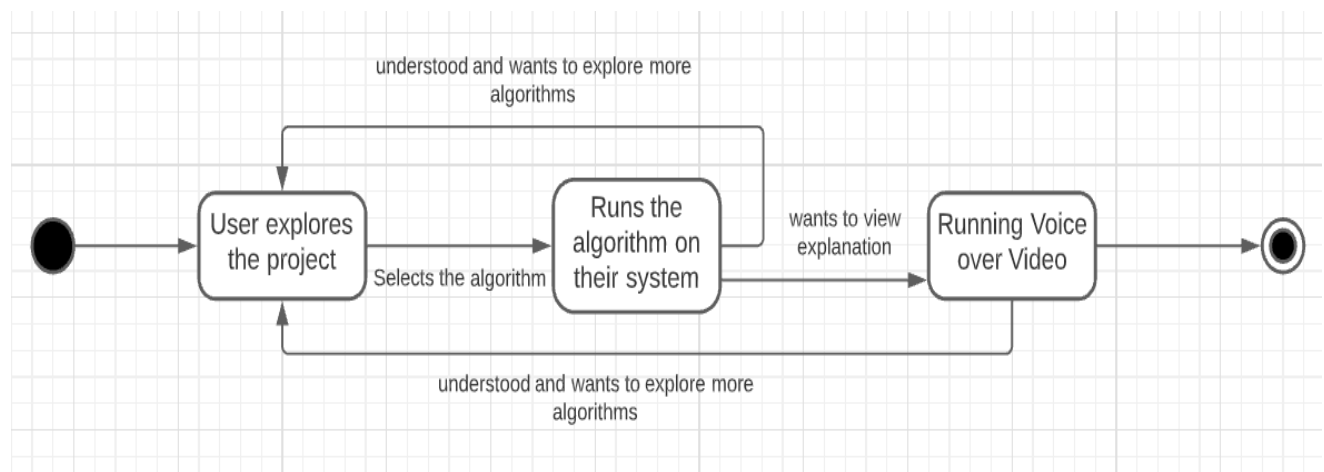




In fig.4, the ER Diagram of our project depicts the relationship between the user(entity), AlgoViz(entity) and Voice over Video(weak entity). The relationship between the user and our project is defined as “chooses algorithm” and the relationship between the project and voice over video is a weak relationship defined as”selects hyperlink”(to the video). The user entity has the key attribute as “User\_name” and base attribute as “user\_email”. Algo viz contains attributes namely the algorithms 1 and 2.

Voice over Video is a weak entity as it depends on the relationship, “selects hyperlink”, which is a choice for the user. The user has a choice between Hindi and English explanation.

## STATE DIAGRAM



In fig. 5, the state diagram explains the various stages in which the project is running. The start symbol leads to the state of the user exploring the project and deciding which algorithm he wants to visualize. Then he selects the algorithm and runs the algorithm on their system . After that, if the user understood the algorithm and its working he can go back to the previous state of exploring the other algorithms or he can run the voice over video of the explanation

of that algorithm. Completing this state, the user has the choice to terminate the process or go back to the first state.

## **2.8 ASSUMPTIONS AND DEPENDENCIES**

The assumptions made from our side are that the user has basic knowledge of programming and is able to understand and follow simple steps to run our project if he/she desires. Also, the user is capable of fulfilling basic requirements as mentioned above.

This project, to a large extent, is dependent on the learners(users) who don't have proper resources or opportunities to study computer algorithms or have some difficulty in understanding them. The main hindrance which most students face is that they don't have access to explanations and illustrations in their native language or a language more familiar to them. Hence, we are providing Voice Over Videos in the most common and most widely used languages namely, Hindi and English. This will help the learners to study and understand certain algorithms easily and they could also re-watch the mp4 files as per their convenience and pace.

## **3. SYSTEM REQUIREMENTS**

### **3.1 User Interface**

The user will have a command line application where there will be an option to select an algorithm that the user needs to visualize or get a Voice Over Video.

### **3.2 Software Interface**

SFML - SFML is a visualising library used in C and C++. It can also be used to add audio to projects one develops.

SDL2 - SDL2 Library which is a better and updated version than graphics.h for visualizing the algorithm that the user will opt to learn

SDL – SDL Library is a combined library with SDL2 for visualizing in C and C++.

### **3.3 Database Interface**

We will be taking a predefined Array set as input. And the visualization will be performed on the following array set.

For eg:

```
int numbers[] = { 66, 83, 343, 111, 500, 182, 46, 370, 480, 527, 266, 167, 163, 551, 462, 101, 143, 549, 82, 100, 69, 482, 491, 388, 588, 424, 138, 564, 519, 504, 52, 97, 419, 171, 275, 48, 247, 23, 441, 98, 240, 420, 203, 582, 365, 415, 267, 301, 512, 210, 16, 278, 463, 139, 392, 581, 338, 74, 282, 211, 296, 518, 468, 54, 204, 363, 375, 33, 130, 99, 31, 59, 575, 37, 361, 39, 312, 434, 200, 106, 389, 221, 220, 255, 196, 68, 474, 292, 231, 513, 369, 283, 201, 445, 449, 194, 251, 294, 206, 274 }
```

### **3.4 Protocols**

Windows - Processor - 4 x 2.42GHz / RAM - 4GB

HDD - 40GB(free space)

MacOS - Processor - 8 x 2.3GHz / RAM - 16GB

SSD - 20GB(free space)

## **4. NON-FUNCTIONAL REQUIREMENTS**

### **4.1 Performance requirements**

This Project will be very easy to operate and the user needs only some basic hardware and software requirements in order to successfully compile the source code through GitHub.

The user runs the code on command prompt and selects the algorithm as per his/her choice. The algorithm is then visualized on a window , with different colours and animations using the two libraries. If the user wishes to view the voice over video, they need to open the youtube window through the link provided which will play the video for them.

### **4.3 Software Quality Attributes**

**Adaptability** - This project is adaptable to those novice learners who don't get opportunity or required resources so that they can understand and study difficult algorithms and implement them. This project will be scaled according to the user needs where languages will be added



as well as algorithms will be added without any external intervention because it is solely the developers role to change or add functionalities in the source code.

**Availability** - This project is available over the internet for all the learners at very easy. The algorithms that we will be considering will be a bit rare and these will be a bit difficult to understand. Also this version will be a bit heavy because their will be two functionalities that are Visualising and Voice Over Video features. Also the source code will be easily available on GitHub for users to run it in their local machines.

**Correctness** - This project's correctness depends entirely on the user because it is very user friendly and the user has simple steps in order to get the product. Also the user will also have an option to re-run the code and again operate the same way in order to see another visualization or get another Voice Over Video.

**Flexibility** - This project is very flexible as it will be available on the internet for different Operating System users such as Windows and Macintosh. The output screen will also be different in both the scenarios like in Windows the user will get a console output whereas a Macintosh user will get Terminal output.

**Interoperability** - This basic interoperability of this project from taking user input as the the choice for which algorithm the user wants to see the visualization or get the Voice Over Video. After the selected functionality is done the code will terminate and the user can re-run the code accordingly.

**Maintainability** - The maintainability of this project will lie only on the developers who will have permissions to edit the source code according to the issue which will be raised by the users. Also the developers will be adding the asked or trending algorithms by the users.

**Portability** - This project is portable to many other softwares or platforms. The only requirement that the user needs to fulfill is the need to download all the libraries and dependencies that we must have used after looking into the errors that might come up while running in other platforms or softwares.

**Reliability** - This project is really reliable because it is very user friendly as well as the user just needs some empty space as the project is a bit big in size due to its functionalities.

**Reusability** - This project is very re-usable since it will be scaled up for many more versions which will contain more algorithms and functionalities. Also the project can be re-run for a more user friendly interface.

**Robustness** - This project is robust to error since the user will not encounter them the only need is to download and install the required libraries and dependencies for successfully compiling or running the source code.

**Testability** - Our project will be tested on each user machine as the user will need to run the source code in order to understand or study the algorithm.

**Usability** - Our project will be used by novice learners who do not get proper resources to study and understand coding related algorithms. This project will be very efficient and effective as this will help them study within their native language also as the newer version will be made after scaling the project. Users will be fully satisfied since it is very user friendly and helpful for students of many age groups.