**State University of New York at New Paltz**

**Matthew Morfea**

**Project Type: On-Campus Project Assignment, Class Key: c-s21-20, CPS485-01: Projects**

**OS Visualizations**

**PROJECT PROPOSAL**

**“Computer Science Projects”**

**Spring 2021**

**(Prof. Hanh Pham)**

**TABLE OF CONTENTS**

**1. Problem Description**

**1.1 Business Context and Goals** ….……………………………………….…………….……. page 03

**1.2 Technical Requirements** ….……………………………………………….……….………. page 04

**1.3 Your Responsibilities** ……...……………………………………………….……….………. page 04

**2. Technologies**

**2.1 Related Technologies** ….……………………………………….……………………..…….. page 05

**2.2 Newly Learned Skills/Technologies …**………………………………..………………. page 05

**3. Plan**

**3.1 Ideas for Solution (Architecture + Protocols)** …….……………………..…..….. page 06

**3.2 Programming/Coding Components** …….……………………………..………………. page 07

**3.3 Schedule** ………………………………………………………………………………………….... page 08

**1. Problem Description**

**1.1 Business Context and Goals**

Sometimes it’s hard to learn about algorithms based on concepts or equations, especially for a huge and complex field like Operating Systems. This project is designed to focus on the visualization of algorithms pertaining to different concepts in Operating Systems. The software we are creating is designed to aid teachers and students demonstrate and learn the varying algorithms of Operating Systems. This includes many algorithms from different places of the wide field of Operating Systems. This includes areas such as CPU Scheduling, Disk Scheduling, Memory Allocation, Page Replacement Methods, and others. Being attached to the SUNY New Paltz web server, Users (teachers, students, etc.) can use this software wherever they can connect to the internet. This could be in a classroom giving a demonstration, in a dorm or home studying, or simply on a mobile device to dive into new knowledge in the world of Operating Systems.

As we are in the design and beginning implementation process of the project, we can see multiple directions and features our software should be able to achieve. On a basic design our software will be hosted on the SUNY New Paltz web server and will be accessible to anyone with an internet connection. This will connect people to the front-end website where they can learn and interact with the algorithms on the web server. The algorithms themselves will take user input and be able to animate the user’s inputs based on the data set they provide and the algorithms they are currently viewing. Each algorithm has buttons and features based on which class of algorithms you would be looking at. Some of the buttons include Previous and Next features allowing you to move one step at a time to show the logic behind the algorithm. Play and Pause, which show a slowed down version of the algorithm along with highlights of what specifically the algorithm is changing as it progresses down the information the user entered. Start and end, that simply display before and after the algorithm has run its course.

Some further features have also been considered and may or may not be implemented, this includes allowing logins for users to keep track of the algorithms and maybe have access to further information. Connection to a database to load the algorithms as it may be better for efficient of the website and security. Some animations may also be placed solely on the front end of the design of the website itself to give off a more modern and flashier website to attract more users.

Overall, the software’s core if based on algorithm implementation and the Graphics User Interface. The user can input numbers and information which leads to the output and animation from the algorithm. The user can input button clicks to view certain steps of the algorithm which are outputted based on the button clicks. Based on the information provided the user can learn more about the Algorithms of Operating Systems.

**1.2 Technical Requirements**

The technical requirements are straightforward for our current design of the project. The software can be viewed through the SUNY New Paltz web server, thus, anywhere with an internet connection and search engine such as google, Firefox, etc., s its hosted on a website. The user makes clicks via mouse and keyboard (or touchscreen) to interact with the main components of the website. The main components of the website being the front-end of the website itself and the actual algorithm’s visualizations in the form of graphics user interfaces. Any standard computer of smart phone would be able to access both the website and the graphics user interface.

The processing between the website to user and algorithm to user is as follows:

1. User connects to website to display home screen.
2. User picks a specific area of algorithms to choose from, then inputs numbers and information for the algorithm via the graphics user interface.
3. As the user determines which buttons to be pressed the algorithms processes the user’s information accordingly:
   1. Play: Algorithms outputs all numbers / information in sequence of steps.
   2. Next: Algorithm outputs one number / part of information.
   3. Previous: GUI outputs previous Processes numbers / information.
   4. Pause: Pauses play feature wherever in the processing.
   5. Start: Goes to start before algorithms output.
   6. End: Goes to end after algorithms output.
4. The user can then move to another class of algorithms or another algorithm through navigation on the website.

**1.3 Your Responsibilities**

My responsibilities are specifically for the Disk Scheduling section of algorithms. This includes many different algorithms like FCFS (First Come First Serve), SSTF (Shortest Seek Time First), SCAN, CSCAN (Circular SCAN), LOOK, CLOOK (Circular LOOK), and others. I am required to implement these algorithms as well as create a smooth and straight forward visualization in the form of a graphics user interface for each. These algorithms mentioned may be subject to change as some of them may not be required or others may be required later. The best way to both implement and display these algorithms is through the process of arrays and JavaScript to draw lines that represent the paths that the read / write head would take on an actual disk drive. I am responsible for these concepts as well as making sure they work on the website setting.

**2. Technologies**

**2.1 Related Technologies**

For the software and my contributions to the project, I am using a combination of many front-end languages on our web server. I am using HTML5, CSS, and Bootstrap5 to design portions of the framework to house the algorithms. Furthermore, I am using HTML5, CSS, JavaScript (HTML) DOM, JavaScript, HTML CANVAS, and jQuery to create the visualizations for the algorithms mentioned in section 1.3. This is all done by hosting the web server through SUNY New Paltz’s Wyvern Server, which I am accessing through WinSCP. All the code being typed up in Visual Studio Code. For the design portions, I am using the Microsoft Office Suite to create flowcharts, PowerPoints, and type up reports.

If this project ends up having me work on any back end portions, I would be using both SQL and Putty to access the database.

**2.2 Newly Learned Skills/Technologies**

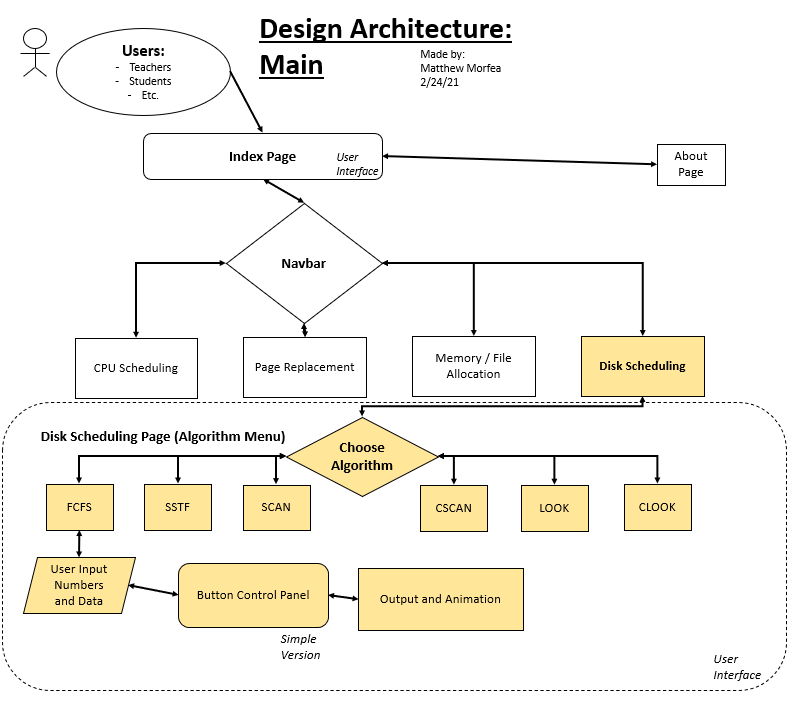
There are many new skills that I have started to work on through the creation of this project. Some of which being:

* JavaScript
* JavaScript (HTML) DOM
* HTML CANVAS
* jQuery
* Putty

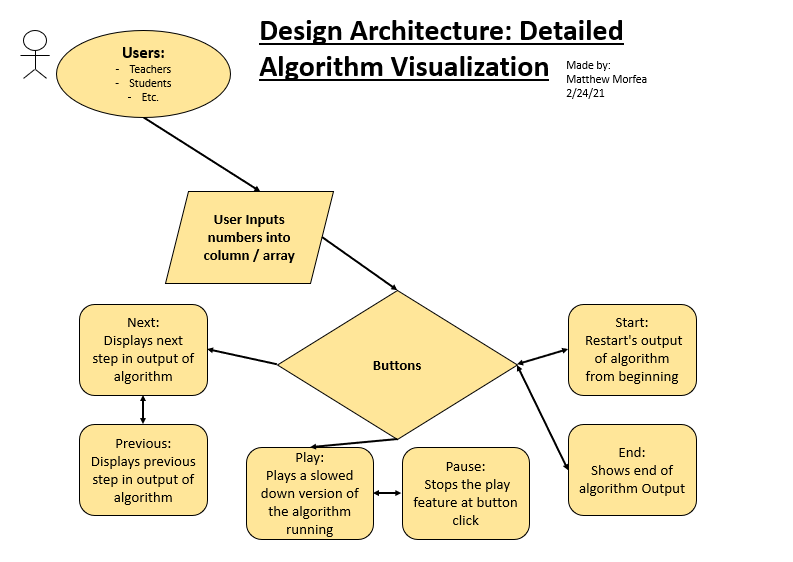
**3. Plan**

**3.1 Ideas for Solution (Architecture + Protocols)**

Architecture: Main Program



Protocol: (Buttons) Algorithm Output control



**3.2 Programming/Coding Components**

For this project I will be using this list of Languages, Packages and Tools:

* HTML5
* HTML Canvas
* CSS
* Bootstrap5
* Standard JavaScript
* JavaScript DOM
* jQuery
* WinSCP
* Visual Studio Code

If we connect to a database later in the semester:

* Putty
* SQL

**3.3 Schedule**

|  |  |  |
| --- | --- | --- |
| Date | What to be done | By whom |
|  |  |  |
| Feb 3 | Get assignments | Professor / Group |
| Feb 10 | Review in OS Algorithms and Research in JavaScript,  Setup Web Framework | Group |
| Feb 17 | Design of Algorithms, Research in JavaScript,  Setup Web Framework | Group / Matthew |
| Feb 24 | Research in JavaScript,  Midterm Report,  Preparation for Midterm Presentation | Group |
| Mar 3 | Oral Midterm Presentation,  Implementation of Code (FCFS) | Group |
| Mar 10 | Implementation of Code / Debugging (FCFS) | Group |
| Mar 17 | Implementation of more Algorithms (SSTF, …) | Group |
| Mar 24 | Code, Debugging, Analysis | Group |
| Mar 31 | Code, Debugging, Analysis | Group |
| Apr 7 | Debugging, Analysis, Testing, and Documentation | Group |
| Apr 14 | Testing, Start Final Report, Start oral Presentation Slides | Group |
| Apr 21 | Final Report, Oral Presentation Slides | Group |
| Apr 28 | Practice and Record Final Presentation Video | Group |
| May 5 | Final presentation (Job Fair, poster and equipment) | Group / Class / Professor |