University of Waterloo

Faculty of Engineering

Department of Electrical and Computer Engineering

StreamingOS: Low Cost Education System

Progress Report

Group 2020.15

Prepared by:

Matthew Milne – 20626854

Vidit Soni – 20627647

Vinayak Sharma – 20585279

Anurag Joshi – 20604210

Surag Sudesh – 20636861

Consultant:

Wojciech Golab

July 29th, 2019

**Table of Contents**

[1.0 Overview of the project 3](#_Toc15246100)

[1.1 Revised Project Abstract 3](#_Toc15246101)

[1.2 Original Project Timeline 3](#_Toc15246102)

[2.0 Current Status of Project 4](#_Toc15246103)

[2.1 Prototype Completion 4](#_Toc15246104)

[2.2 Student Hours 5](#_Toc15246105)

[3.0 Discussion 6](#_Toc15246106)

[Appendix A: Student Logs 7](#_Toc15246107)

[1) Anurag Joshi 7](#_Toc15246108)

[2) Matthew Milne 10](#_Toc15246109)

[3) Surag Sudesh 13](#_Toc15246110)

[4) Vidit Soni 16](#_Toc15246111)

[5) Vinayak Sharma 19](#_Toc15246112)

[Appendix B: Initial Prototype Demonstration Feedback Sheet 22](#_Toc15246113)

# **Overview of the project**

## **1.1 Revised Project Abstract**

As technology improves in the 21st century, using mobile devices for educational purposes is becoming more common in primary and secondary schools. In addition to the cost, this technology becomes quickly outdated and needs to be replaced, forcing schools to spend continuous amounts of money on maintenance. StreamingOS is a system that provides students and teachers with inexpensive thin endpoint devices, with the resource-heavy OS being streamed to these devices from a backend server using container virtualization. The objective of this project is to design a powerful, inexpensive device and streaming system that enhances the learning experience. StreamingOS uses an inexpensive endpoint device and container virtualization to visually render and stream the execution of applications from a server or the teacher’s computer to these devices used by the students. The system design leverages concepts learned in distributed computing, operating systems, database theory, and networking courses. The advantage of this design over current alternatives is that it is scalable while enabling the teacher full control of what software each student views. The inexpensive hardware helps break down the barrier of the lack of technology in school settings and empowers teachers to incorporate more modern-day means of learning in their classrooms.

## **1.2 Original Project Timeline**

Due to minimal dependencies between the different subcomponents, they can be developed in parallel if a virtual test environment has been setup. Figure 1 shows the project timeline.

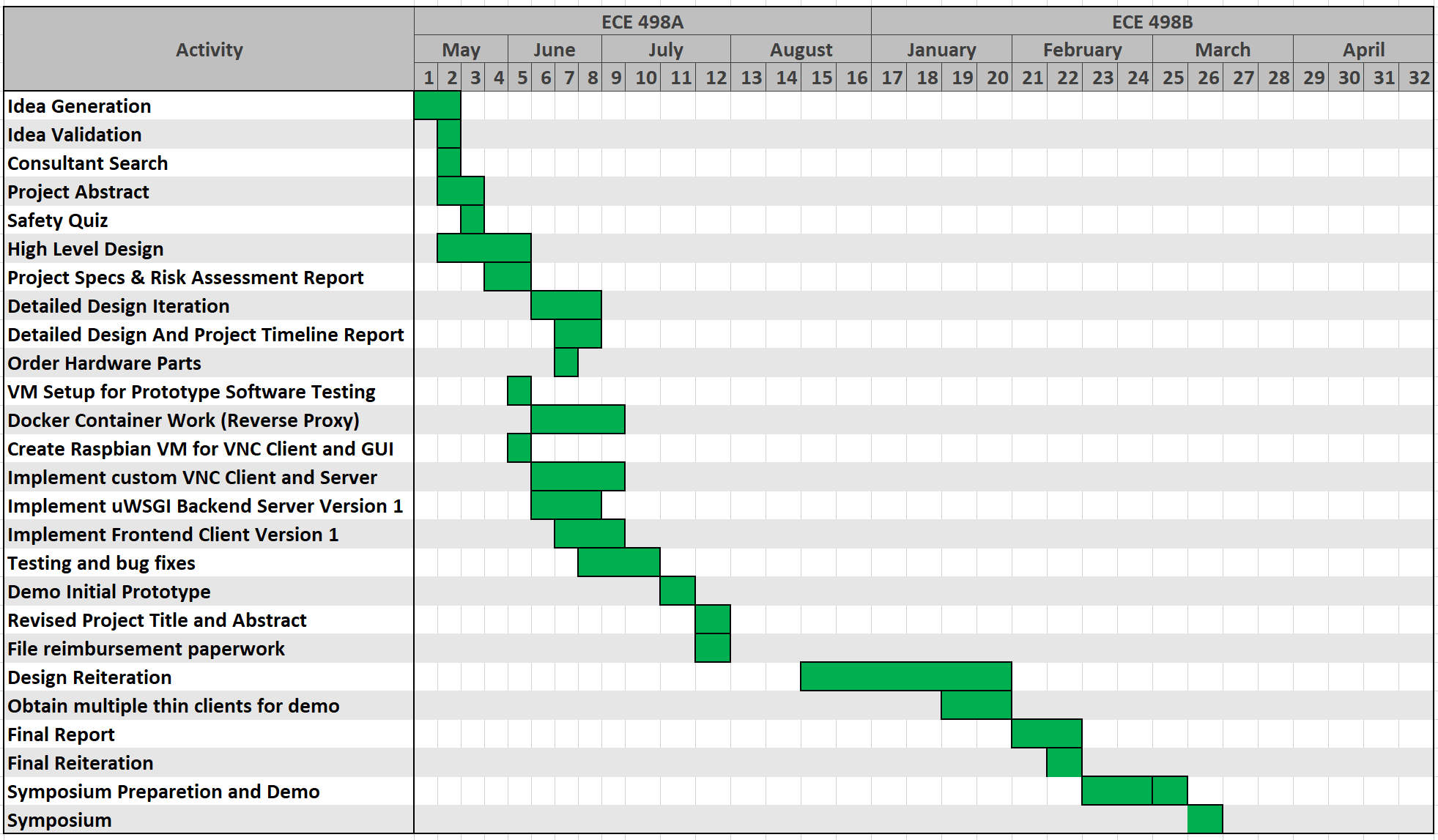


Figure 1 – Project Timeline

# **Current Status of Project**

## **2.1 Prototype Completion**

We estimate that our prototype model is roughly around 60% complete compared to the actual full model we envisioned. The first functional property we mentioned was latency and we had said that the end to end delay in our system should be less than 300ms. When we gathered data by running tests on our model, we found that the current end to end delay was roughly around 150ms. Therefore, we have met the requirements for this functional property. The next one was Authentication System. We believe we have completed this property as well since we use an OAuth 2.0 based authentication protocol for user authentication. The “Teacher Control” functionality is partially completed since we have the resources required to grant and revoke access to applications to students. However, we are currently missing on the functionality that allows the teacher to view live data from the student’s screen on their screen. The next property was Multi User support. Currently, we have the feature where one teacher and around 5-6 students can be using our systems concurrently in a mini lecture room environment in an efficient manner. We believe we have partially completed this feature as well.

The next property was Network Connection. We are satisfying this property by making use of the HTTP methods for REST API calls between the client and server. Our next property was high performance and we believe we are already achieving this property since our system uses approximately 3.75 GB RAM and 11% CPU load for 3 OS containers, a reverse proxy and a database container which is quite reasonable compared to our initial estimates. We are currently missing the Students Bird Eye View functionality where the teacher will be able to see student specific data on their screens such as their marks, class attendance etc. The next non- functional property we mentioned was security which we accomplish by using Bcrypt and OAuth 2.0 for user authentication. The cost of the device currently is around $27 which is lower than our initial estimate of $75. Therefore, we satisfy this property as well. We achieve the property of reliability by making sure that the Electron application for the teacher's portal can reliably transmit commands to grant and revoke access of applications to each student. We also read database flags for containers before assigning them. We also perform some health check on endpoints for each container to ensure reliability. We have not met the property of device portability yet since our raspberry pi solution still requires a monitor, keyboard and mouse set to work. This will be a major task to solve in the upcoming term.

Lastly, we believe we have partially achieved heterogeneity since our Electron application runs on all platforms but there is a decent chunk of work left in the VNC Client part of this non-functional property. Overall, we believe that we have achieved around 60% of the functional and non-functional properties. This is also in conjunction with the estimates provided by our consultant which state that the prototype model is around 50 to 74% complete with respect to the full version.

## **2.2 Student Hours**

Table 1 below shows the number of hours spent each student has worked on so far for the project.

Table 1 - Number of hours invested by each team member in 4A

|  |  |
| --- | --- |
| **Student** | **# of Hours Invested** |
| Anurag Joshi | 123 |
| Matthew Milne | 123 |
| Surag Sudesh | 124 |
| Vidit Soni | 125 |
| Vinayak Sharma | 123 |
| Total | 618 |

# **3.0 Discussion**

Based on the list of functional and non-functional specifications listed in the detailed design document, the prototype is ~60% complete. The group is highly confident in the ability to finish the project before March 2020.

This is a list of tasks that need to be performed before the project can be considered complete along with expected time required.

* Improvements to latency by implementing quic protocol for VNC client server communication (~2 weeks)
* Investigate webRTC (~ 1 week)
* Implement Student Bird’s Eye view using noVNC (~ 3 weeks)
* Implement software stack on a low-cost tablet (potentially Fire 7 HD) to make the solution portable (~ 5 weeks)
* Improvements to the UI/UX of the Teacher and Student Electron apps (~ 2-3 weeks)
* Implement user specific storage (~ 2 weeks)

The time estimates are calculated based on the amount of time spent during ECE 498A to perform similar tasks. Since the prototype is 60% done, most of the above-mentioned tasks can be performed in parallel with 1-2 members taking the lead for each.

During the coop term, the team plans to conduct a 1-hour meeting using Skype (or similar video conferencing software) every Sunday at 1PM EST to discuss current status, blocking issues and triage any work that needs to be done. All issues are tracked on our Trello Kanban board. Each member will spend approximately 3-5 hours per week on project work.

Any work that is not completed in the coop term, and the final end-to-end system test will be completed in the first few weeks of ECE 498B.

# **Appendix A: Student Logs**

## **Anurag Joshi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ECE498A: Student Log** | | | | | |
| **Name: Anurag Joshi Group: 2020. 15 Signature:**  **By signing above, I am stating that this is an accurate account of the tasks, dates, and times that I worked on my capstone design project.** | | | | | |
| **Task** | **Date** | **Start time** | **Finish time** | **Hours** | **Running total of hours** |
| Idea Generation | May 8 | 11:30 | 14:30 | 3 | 3 |
| Safety Quiz Reading | May 11 | 2:00 | 3:00 | 1 | 4 |
| Idea Generation | May 14 | 11:00 | 13:00 | 2 | 6 |
| Idea Generation | May 15 | 11:30 | 14:30 | 3 | 9 |
| Researching and meeting with profs | May 16 | 9:00 | 12:30 | 3.5 | 12.5 |
| meeting with profs | May 17 | 10:00 | 13:00 | 3 | 15.5 |
| Researching Electron vs PWA | May 19 | 13:00 | 18:00 | 3 | 18.5 |
| Team Meet | May 20 | 16:00 | 17:30 | 1.5 | 20 |
| Writing Project Abstract | May 20 | 20:00 | 21:00 | 1 | 21 |
| Create Prototype of Hello World | May 22 | 15:00 | 18:00 | 3 | 24 |
|  | May 23 | 15:00 | 16:00 | 1 | 25 |
| Safety Quiz | May 24 | 15:45 | 16:15 | 0.5 | 25.5 |
| Team Meet | May 27 | 16:00 | 17:30 | 1.5 | 27 |
| Start building actual app | May 29 | 15:00 | 17:00 | 2 | 29 |
| Building Teachers App | May 30 | 12:00 | 14:00 | 2 | 31 |
| Team Meet | May 30 | 15:00 | 16:00 | 1 | 32 |
| Write my part in Project Spec doc | June 1 | 12:00 | 14:00 | 2 | 34 |
| Finding suitable bootstrap templates | June 2 | 12:00 | 14:00 | 2 | 36 |
| Team Meet | June 3 | 16:00 | 17:30 | 1.5 | 37.5 |
| Play with bootstrap | June 4 | 9:00 | 10:00 | 1 | 38.5 |
| Finalize Project Spec Doc | June 5 | 11:30 | 14:30 | 3 | 41.5 |
| Team Meet | June 6 | 15:00 | 16:00 | 1 | 42.5 |
| Creating actual HTML templates | June 12 | 16:00 | 19:00 | 3 | 45.5 |
| Hardcoding actual page to show to team | June 13 | 12:00 | 15:00 | 3 | 48.5 |
| Team Meet | June 13 | 15:00 | 16:00 | 1 | 49.5 |
| Write my part in the Design Doc | June 23 | 10:00 | 12:00 | 2 | 51.5 |
| Team Meet | June 24 | 16:00 | 17:30 | 1.5 | 53 |
| Team meet to finalize design doc | June 26 | 11:30 | 14:30 | 3 | 56 |
| Implement changes in the hardcoded HTML page after discussion | June 29 | 12:00 | 17:00 | 5 | 61 |
| CSS modifications | July 2 | 9:00 | 14:00 | 5 | 66 |
| Remove hardcoding and point to JS methods | July 3 | 15:00 | 20:00 | 5 | 71 |
| Fill in the JS Methods to point to user data lists. | July 4 | 10:00 | 14:00 | 4 | 75 |
| Implement granting and revoking application access code | July 6 | 9:00 | 13:00 | 4 | 79 |
| Get the HTTP REST API calls working | July 10 | 13:00 | 20:00 | 7 | 86 |
| Bug fixes with Status table | July 11 | 13:00 | 16:00 | 3 | 89 |
| Fix bugs related to AJAX call ordering | July 15 | 11:00 | 14:00 | 3 | 92 |
| Solve CORS issue when running on Electron | July 15 | 12:00 | 14:00 | 2 | 94 |
| Create a template Student Electron Application | July 16 | 14:00 | 19:00 | 5 | 99 |
| Create JS methods for Student App | July 16 | 16:00 | 18:00 | 2 | 101 |
| Meeting | July 17 | 18:00 | 21:00 | 4 | 105 |
| Meeting | July 18 | 11:30 | 14:00 | 2.5 | 107.5 |
| Prepare for Demo | July 19 | 11:30 | 14:00 | 2.5 | 110 |
| Put the whole system together and test end to end functionality | July 23 | 10:00 | 15:00 | 5 | 115 |
| Minor bug fixes in Student App | July 24 | 11:30 | 14:00 | 2.5 | 117.5 |
| Demo Setup and Demo | July 25 | 13:00 | 15:30 | 2.5 | 120 |
| Work on the Progress Report | July 28 | 20:00 | 23:00 | 3 | 123 |

## **Matthew Milne**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ECE498A: Student Log** | | | | | |
| **Name: Matthew Milne Group: 2020. 15 Signature:**  **By signing above, I am stating that this is an accurate account of the tasks, dates, and times that I worked on my capstone design project.** | | | | | |
| **Task** | **Date** | **Start time** | **Finish time** | **Hours** | **Running total of hours** |
| Idea Generation | May 8 | 11:30 | 14:30 | 3 | 3 |
| Safety Quiz Reading | May 11 | 2:00 | 3:00 | 1 | 4 |
| Idea Generation | May 14 | 11:00 | 13:00 | 2 | 6 |
| Idea Generation | May 15 | 11:30 | 14:30 | 3 | 9 |
| Researching and meeting with profs | May 16 | 9:00 | 12:30 | 3.5 | 12.5 |
| Meeting with profs | May 17 | 10:00 | 13:00 | 3 | 15.5 |
| Researching uWSGI vs Django | May 19 | 13:00 | 16:30 | 3.5 | 19 |
| Team Meet | May 20 | 16:00 | 17:30 | 1.5 | 20.5 |
| Writing Project Abstract | May 20 | 20:00 | 21:00 | 1 | 21.5 |
| Get uWSGI running, and a single endpoint working | May 22 | 16:00 | 18:30 | 2.5 | 24 |
| Work on REST API documentation | May 23 | 15:00 | 18:00 | 3 | 27 |
| Get uWSGI running, and a single endpoint working | May 24 | 13:00 | 15:00 | 2 | 29 |
| Safety Quiz | May 24 | 15:30 | 16:00 | 0.5 | 29.5 |
| Team Meet | May 27 | 16:00 | 17:30 | 1.5 | 31 |
| Creating authentication endpoints | May 30 | 12:00 | 15:00 | 3 | 34 |
|  | May 30 | 16:00 | 19:00 | 3 | 37 |
| Team Meet | May 30 | 15:00 | 16:00 | 1 | 38 |
| Write my part in Project Spec doc | June 1 | 9:00 | 11:00 | 2 | 40 |
| Create SQL statements to initialize the database | June 2 | 9:00 | 11:00 | 2 | 42 |
| Team Meet | June 3 | 16:00 | 17:30 | 1.5 | 43.5 |
| Research OAuth 2.0 | June 4 | 9:00 | 10:00 | 1 | 44.5 |
| Finalize Project Spec Doc | June 5 | 11:30 | 14:30 | 3 | 47.5 |
| Team Meet | June 6 | 15:00 | 16:00 | 1 | 48.5 |
| Work on OAuth 2.0 Implementation | June 12 | 16:00 | 19:00 | 3 | 51.5 |
|  | June 13 | 12:00 | 18:00 | 6 | 57.5 |
| Team Meet | June 13 | 15:00 | 16:00 | 1 | 58.5 |
| Write my part in the Design Doc | June 23 | 9:00 | 13:00 | 4 | 62.5 |
| Team Meet | June 24 | 16:00 | 17:30 | 1.5 | 64 |
| Team meet to finalize design doc | June 26 | 11:30 | 14:30 | 3 | 67 |
| Implement the create and delete user | June 29 | 10:00 | 14:00 | 4 | 71 |
|  | July 2 | 9:00 | 12:00 | 3 | 74 |
| Implement user information related calls | July 3 | 12:00 | 13:00 | 1 | 75 |
|  | July 4 | 9:00 | 12:00 | 3 | 78 |
| Implement granting and revoking calls | July 6 | 9:00 | 13:00 | 4 | 82 |
| Testing and cleaning up calls | July 10 | 9:00 | 12:00 | 3 | 85 |
|  | July 11 | 13:00 | 16:00 | 2 | 88 |
|  | July 15 | 11:00 | 14:00 | 3 | 91 |
| Document the REST APIs | July 15 | 12:00 | 14:00 | 2 | 93 |
| Fixed the HTTP response code for add and revoke | July 16 | 12:00 | 14:00 | 2 | 95 |
|  | July 16 | 16:00 | 18:00 | 2 | 97 |
| Meeting | July 17 | 18:00 | 21:00 | 4 | 101 |
| Meeting | July 18 | 11:30 | 14:00 | 2.5 | 103.5 |
| Prepare for Demo | July 19 | 11:30 | 14:00 | 2.5 | 105 |
|  | July 22 | 13:00 | 17:00 | 4 | 110 |
| Put the whole system together and test end to end functionality | July 23 | 10:00 | 15:00 | 5 | 115 |
|  | July 24 | 11:30 | 14:00 | 2.5 | 117.5 |
| Demo Setup and Demo | July 25 | 13:00 | 15:30 | 2.5 | 120 |
| Progress Report | July 29 | 9:00 | 12:00 | 3 | 123 |

## **Surag Sudesh**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ECE498A:  Student Log** | | | | | |
| **Name:      Surag Sudesh               Group:        2020. 15                                  Signature:**  **By signing above, I am stating that this is an accurate account of the tasks, dates, and times that I worked on my capstone design project.** | | | | | |
| **Task** | **Date** | **Start time** | **Finish time** | **Hours** | **Running total of hours** |
| Idea Generation | May 8 | 11:30 | 14:30 | 3 | 3 |
| Safety Quiz Reading | May 11 | 2:00 | 3:00 | 1 | 4 |
| Idea Generation | May 14 | 11:00 | 13:00 | 2 | 6 |
| Idea Generation | May 15 | 11:30 | 14:30 | 3 | 9 |
| Researching and meeting with profs | May 16 | 9:00 | 12:30 | 3.5 | 12.5 |
| Meeting with profs | May 17 | 10:00 | 13:00 | 3 | 15.5 |
| Research X vs RDP vs VNC | May 19 | 14:00 | 18:30 | 4.5 | 20 |
| Team Meet | May 20 | 16:00 | 17:30 | 1.5 | 21.5 |
| Writing Project Abstract | May 20 | 20:00 | 21:00 | 1 | 22.5 |
| Additional research on X vs VNC | May 22 | 15:00 | 18:30 | 3.5 | 26 |
| Safety Quiz | May 24 | 15:30 | 16:00 | 0.5 | 26.5 |
| Team Meet | May 27 | 16:00 | 17:30 | 1.5 | 28 |
| Work on libVNC research and documentation | May 28 | 12:00 | 16:00 | 4 | 32 |
|  | May 39 | 16:00 | 18:00 | 2 | 34 |
| Team Meet | May 30 | 15:00 | 16:00 | 1 | 35 |
| Write my part in Project Spec doc | June 1 | 9:00 | 11:00 | 3 | 38 |
| Setup dev environment and run example vnc client | June 2 | 9:00 | 12:00 | 3 | 41 |
| Team Meet | June 3 | 16:00 | 17:30 | 1.5 | 42.5 |
| Research VeNcrypt and disable it | June 4 | 9:00 | 10:00 | 1.5 | 44 |
| Finalize Project Spec Doc | June 5 | 11:30 | 15:00 | 3.5 | 47.5 |
| Team Meet | June 6 | 15:00 | 16:00 | 1 | 48.5 |
| Implement basic VNC client using raw encoding | June 12 | 16:00 | 19:00 | 3 | 51.5 |
|  | June 13 | 12:00 | 19:00 | 7 | 58.5 |
| Team Meet | June 13 | 15:00 | 16:00 | 1 | 59.5 |
| Write my part in the Design Doc | June 23 | 11:00 | 15:00 | 4 | 63.5 |
| Team Meet | June 24 | 16:00 | 17:30 | 1.5 | 65 |
| Team meet to finalize design doc | June 26 | 11:30 | 14:30 | 3 | 68 |
| Implement copyrect encoding | June 29 | 11:30 | 16:00 | 4.5 | 72.5 |
|  | July 3 | 12:00 | 13:00 | 1 | 73.5 |
| Implement tight encoding | July 6 | 10:00 | 14:00 | 4 | 77.5 |
|  | July 9 | 10:30 | 13:00 | 2.5 | 80 |
| Enable zlib compression | July 11 | 13:00 | 16:00 | 3 | 83 |
|  | July 15 | 10:00 | 13:00 | 3 | 86 |
| Update Student Electron App to invoke proper binaries | July 15 | 12:00 | 14:00 | 2 | 88 |
| Setup required development libraries on pi and run the vnc client | July 16 | 16:00 | 23:00 | 7 | 95 |
| Meeting | July 17 | 18:00 | 21:00 | 4 | 99 |
| Meeting | July 18 | 11:30 | 14:00 | 2.5 | 101.5 |
| Performance optimizations | July 19 | 11:30 | 14:00 | 2.5 | 104 |
| Rewrite pi client using alternate rendering engine and perf optimizations | July 21 | 21:00 | 05:00 | 7 | 111 |
| Put the whole system together and test end to end functionality | July 23 | 10:00 | 15:00 | 5 | 116 |
|  | July 24 | 11:30 | 14:00 | 2.5 | 118.5 |
| Demo Setup and Demo | July 25 | 13:00 | 15:30 | 2.5 | 121 |
| Progress Report | July 28 | 18:00 | 21:00 | 3 | 124 |

## **Vidit Soni**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ECE498A: Student Log** | | | | | |
| **Name: Vidit Soni Group: 2020. 15 Signature:**  **By signing above, I am stating that this is an accurate account of the tasks, dates, and times that I worked on my capstone design project.** | | | | | |
| **Task** | **Date** | **Start time** | **Finish time** | **Hours** | **Running total of hours** |
| Idea Generation | May 8, 2019 | 11:30 | 14:30 | 3 | 3 |
| Safety Quiz Reading | May 12, 2019 | 10:00 | 11:00 | 1 | 4 |
| Idea Generation | May 14, 2019 | 11:00 | 13:00 | 2 | 6 |
| Idea Generation | May 15, 2019 | 11:30 | 14:30 | 3 | 9 |
| Researching and Meeting with Profs | May 16, 2019 | 9:00 | 12:30 | 3.5 | 12.5 |
| Meeting with Profs | May 17, 2019 | 10:00 | 13:00 | 3 | 15.5 |
| Research Server Stack | May 17, 2019 | 15:00 | 19:00 | 3 | 18.5 |
| Research UWSGI and Setup Basic app | May 18, 2019 | 9:00 | 12:00 | 3 | 21.5 |
| Team Meeting | May 20, 2019 | 16:00 | 17:30 | 1.5 | 23 |
| Writing Project Abstract | May 20, 2019 | 20:00 | 21:00 | 1 | 24 |
| Researched Different Databases | May 21, 2019 | 9:00 | 11:00 | 2 | 26 |
| Setup Basic MySQL database | May 21, 2019 | 18:00 | 20:00 | 2 | 28 |
| Researched Docker | May 22, 2019 | 18:00 | 22:00 | 4 | 32 |
| Finish Safety Quiz | May 24, 2019 | 12:00 | 12:30 | 0.5 | 32.5 |
| Research IPtables | May 25, 2019 | 14:00 | 18:00 | 4 | 36.5 |
| Setup UWSGI in a Docker container | May 26, 2019 | 13:00 | 15:30 | 2.5 | 39 |
| Team Meeting | ￼May 27, 2019 | 16:00 | 17:30 | 1.5 | 40.5 |
| Research IPTables | May 30, 2019 | 15:00 | 16:00 | 1 | 41.5 |
| Work on Project Spec Doc | June 1, 2019 | 10:00 | 12:00 | 2 | 43.5 |
| Team Meet | June 3, 2019 | 16:00 | 17:30 | 1.5 | 45 |
| Finalize Project Specification Doc | June 5, 2019 | 11:30 | 14:30 | 3 | 48 |
| Team Meet | June 6, 2019 | 15:00 | 16:00 | 1 | 49 |
| Team Meet | June 13, 2019 | 15:00 | 16:00 | 1 | 50 |
| Work on Design Document | June 22, 2019 | 12:00 | 16:00 | 4.5 | 54.5 |
| Team Meet | June 24, 2019 | 16:00 | 17:30 | 1.5 | 56 |
| Team Meet to Finish Design Document | June 26, 2019 | 11:30 | 14:30 | 3 | 59 |
| Building RP, and Creating cleaning up endpoints for Routes | June 26-27, 2019 | 22:00 | 1:00 | 3 | 62 |
| Cleaning up RP routes calls and Setting up OS Container. Testing route setup to OS container. | June 27-28, 2019 | 20:00 | 1:00 | 5 | 67 |
| Setup Setup MySQL DB in Docker container, and setup files to create necessary tables | July 2, 2019 | 16:30 | 18:00 | 1.5 | 68.5 |
| Setup peewee in RP codebase, and create data objects for database access/communication | July 2-3, 2019 | 23:00 | 1:00 | 2 | 70.5 |
| Research OAuth2.0 | July 3, 2019 | 17:00 | 18:00 | 1 | 71.5 |
| Setup username and password authentication with BCrypt | July 4, 2019 | 19:00 | 21:00 | 2 | 73.5 |
| Setup OAuth2.0 | July 5, 2019 | 11:30 | 15:00 | 3.5 | 77 |
| Setup UWSGI Permissions Server in Docker | July 6, 2019 | 12:00 | 16:00 | 4 | 81 |
| Create Health Check and Permission Grant/Revoke Endpoints in Permission Server, and change RP code to invoke permissions server endpoints | July 7-8, 2019 | 18:00 | 1:00 | 7 | 88 |
| Setup grant and revoke App permissions (using Shell Commands) in Permissions server, and clean up bugs within Permissions server. | July 9, 2019 | 9:00 | 14:00 | 5 | 93 |
| Remove SQL scripts for Database initialization, and setup Functionality directly inside RP codebase. Create fake data to use. | July 10, 2019 | 12:00 | 16:00 | 4 | 97 |
| Setup Backend (Reverse Proxy, Database, and OS Container) on Windows Test Machine with Hamachi for teammates to connect to. | July 11, 2019 | 18:00 | 21:00 | 3 | 100 |
| Team Meeting | July 17, 2019 | 18:00 | 21:00 | 3 | 103 |
| Team Meeting | July 18, 2019 | 11:30 | 14:00 | 2.5 | 105.5 |
| Prepare for Demo | July 19, 2019 | 11:30 | 14:00 | 2.5 | 108 |
| Prepare for Demo | July 22, 2019 | 13:00 | 17:00 | 4 | 112 |
| Setup complete End-to-End system, and test Functionality | July 23, 2019 | 10:00 | 15:00 | 5 | 117 |
| Setup complete End-to-End system, and test Functionality | July 24, 2019 | 11:30 | 14:00 | 2.5 | 119.5 |
| Setup System and Demo | July 25, 2019 | 13:00 | 15:30 | 2.5 | 122 |
| Progress Report Work | July 28, 2019 | 12:00 | 15:00 | 3 | 125 |

## **Vinayak Sharma**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ECE498A: Student Log** | | | | | |
| **Name: Vinayak Sharma Group: 2020. 15 Signature:**    **By signing above, I am stating that this is an accurate account of the tasks, dates, and times that I worked on my capstone design project.** | | | | | |
| **Task** | **Date** | **Start time** | **Finish time** | **Hours** | **Running total of hours** |
| Idea Generation | May 8 | 11:30 | 14:30 | 3 | 3 |
| Safety Quiz Reading | May 11 | 2:00 | 3:00 | 1 | 4 |
| Idea Generation | May 14 | 11:00 | 13:00 | 2 | 6 |
| Idea Generation | May 15 | 11:30 | 14:30 | 3 | 9 |
| Researching and meeting with profs | May 16 | 9:00 | 12:30 | 3.5 | 12.5 |
| Meeting with profs | May 17 | 10:00 | 13:00 | 3 | 15.5 |
| Research on hardware:  Raspberry pi vs  Orange Pi Prime zero | May 19 | 20:00 | 23:30 | 3.5 | 19 |
| Team Meet | May 20 | 16:00 | 17:30 | 1.5 | 20.5 |
| Writing Project Abstract | May 20 | 20:00 | 21:00 | 1 | 21.5 |
| Research about the database design | May 22 | 18:00 | 20:30 | 2.5 | 24 |
| Work on Database documentation | May 23 | 15:00 | 18:00 | 3 | 27 |
| Work on Hardware documentation | May 24 | 13:00 | 15:00 | 2 | 29 |
| Safety Quiz | May 24 | 15:30 | 16:00 | 0.5 | 29.5 |
| Team Meet | May 27 | 16:00 | 17:30 | 1.5 | 31 |
| Researching about VNC client and learning about Docker | May 30 | 18:00 | 21:00 | 3 | 34 |
|  | May 31 | 16:00 | 19:00 | 3 | 37 |
| Team Meet | May 30 | 15:00 | 16:00 | 1 | 38 |
| Writing my part in Project Spec doc | June 1 | 18:00 | 21:00 | 3 | 40 |
|  | June 2 | 12:00 | 2:00 | 2 | 42 |
| Team Meet | June 3 | 16:00 | 17:30 | 1.5 | 43.5 |
| Research on electron app and armv6l architecture | June 4 | 18:00 | 19:00 | 1 | 44.5 |
| Finalize Project Spec Doc | June 5 | 11:30 | 14:30 | 3 | 47.5 |
| Team Meet | June 6 | 15:00 | 16:00 | 1 | 48.5 |
| Research about Debian OS, Raspbian OS and understanding the armv6l architecture | June 12 | 16:00 | 19:00 | 3 | 51.5 |
|  | June 13 | 16:00 | 22:00 | 6 | 57.5 |
| Team Meet | June 13 | 15:00 | 16:00 | 1 | 58.5 |
| Write my part in the Design Doc | June 23 | 10:00 | 14:00 | 4 | 62.5 |
| Team Meet | June 24 | 16:00 | 17:30 | 1.5 | 64 |
| Team meet to finalize design doc | June 26 | 11:30 | 14:30 | 3 | 67 |
| Setting up local environment to run the backend system: OS Container | June 29 | 17:00 | 21:00 | 4 | 71 |
| Ordering the raspberry PI | July 4 | 18:00 | 19:00 | 1 | 72 |
| Finding a way to install electron on armv6l architecture | July 6 | 20:00 | 2:00 | 5 | 77 |
| Assembling the raspberry PI and buying the mouse and keyboard | July 10 | 19:00 | 23:00 | 5 | 82 |
| Writing the scripts to run the electron app in kiosk mode | July 15 | 12:00 | 15:00 | 3 | 85 |
|  | July 15 | 18:00 | 23:00 | 5 | 90 |
| Integrating the VNC client, electron app on Raspberry PI | July 16 | 16:00 | 23:00 | 7 | 97 |
| Meeting | July 17 | 18:00 | 21:00 | 4 | 101 |
| Meeting | July 18 | 11:30 | 14:00 | 2.5 | 103.5 |
| Prepare for Demo | July 19 | 11:30 | 14:00 | 2.5 | 106 |
|  | July 22 | 13:00 | 17:00 | 4 | 110 |
| Put the whole system together and test end to end functionality | July 23 | 10:00 | 15:00 | 5 | 115 |
|  | July 24 | 11:30 | 14:00 | 2.5 | 117.5 |
| Demo Setup and Demo | July 25 | 13:00 | 15:30 | 2.5 | 120 |
| Progress Report | July 28 | 18:00 | 21:00 | 3 | 123 |

# **Appendix B: Initial Prototype Demonstration Feedback Sheet**

**A screenshot of a cell phone

Description automatically generated**