CS 25-330 Commonwealth Center for Advanced Computing (CCAC) Mainframe Online Curriculum

Prepared for

Dr. Dahlberg

By

Arbab Arif

Long Le

Osman Zafar

Under the supervision of

Dr. Dahlberg

10/11/24

Executive Summary

The executive summary highlights the key points of the document. While your advisor(s) and sponsor are expected to read the document in detail, others may only read the summary looking for a brief overview of the report. Casual readers may look at the summary to decide if they would like to continue reading. Some, more senior decision makers (e.g. executives), may read the summary to help make decisions regarding the future of the project (e.g. continuation, financing, resource allocation, etc.). It is important that all readers get a complete sense of the project, including purpose, primary objectives, design requirements, deliverables, work done to date, and timeline, among other required components provided in a table of contents. Summaries should be considered as “stand-alone” containing a complete account of the essential points of the document in chronological order of the document. Particular focus should be placed on the first sentence in order to draw readers in and should explicitly include the “who, what, and why” of the project. The executive summary is usually between half a page and a full page.

Note: The Executive Summary should be updated between major reports as more knowledge is acquired and understanding of the project expands. For example, when submitting Preliminary Design Report in December 2024, make sure you update this page to reflect the progress on the project since the submission of Project Proposal in early October 2024.

Table of Contents

Section A. Problem Statement

At their core mainframes are high-speed computers with vast amounts of memory and data processing capabilities. Today, mainframes handle roughly 70% of the world's production IT workloads ranging from healthcare to financial services and even government services. The issue these industries are facing is that these systems have been in place for years and legacy systems do not have enough developers and engineers to maintain them, particularly now because of the aging workforce and lack of education for new developers on older technologies such as z/OS and COBOL. Our goal is to develop a curriculum that gives developers and managers enough information to have a fundamental understanding to build upon and be capable of doing regular mainframe work.

A study by Reuters found that "43 percent of banking systems are built on COBOL, 80 percent of in-person transactions use COBOL, [and] 95 percent of ATM swipes rely on COBOL code," (Hartman) while the average age of COBOL and mainframe engineers is 45-55 years old. Moreover, the average retirement age in the United States is 66-67 years which makes it such that many of the current Mainframe engineers will be retiring over the next 10 years. With such a critical technology and the aging workforce, many would believe that universities would teach Mainframe skills to their students but that is not the case. The Open Mainframe Project reported in a blog that out of the roughly 4300 Universities in the United States only 10-15 offer semester-long courses on z/OS and Mainframes (Open Mainframe Project, 2021). These combined factors have resulted in a skills gap in the industry that is currently untenable.

The negative effects of this skill gap are already being felt in many industries that rely on mainframes. An example of this happened in 2012 when the Commonwealth Bank of Australia had a 12-hour outage "that left millions of Australians unable to directly transfer or receive funds on Thursday has been traced to an errant infrastructure upgrade that hobbled the institution's real-time transactions platform" (Bajkowski, 2019). This issue can largely be attributed to the bank's reliance on aging legacy systems and lack of engineers capable of maintaining them.

Our goal in CS 25-330 Commonwealth Center for Advanced Computing (CCAC) Mainframe Online Curriculum is to develop a curriculum that will give new and old engineers the education needed to fill this skills gap and hopefully make it so that companies relying on mainframe systems do not suffer from issues stemming from them.

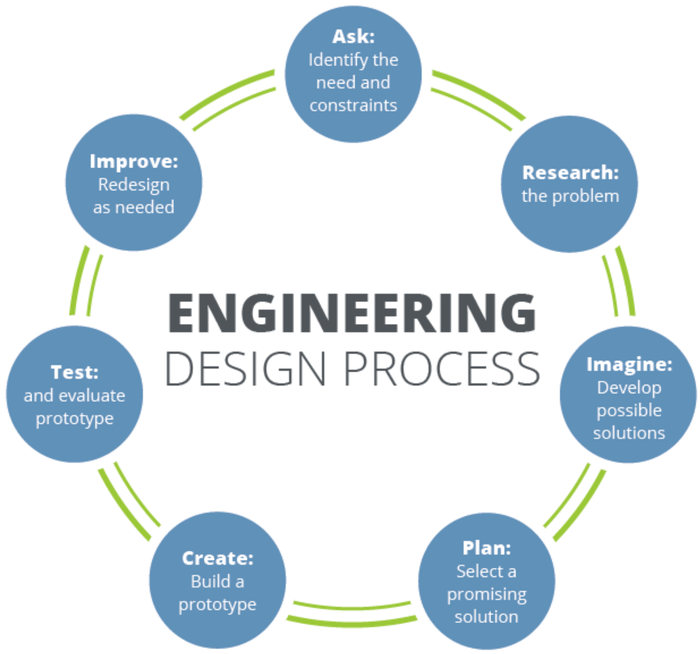

Figure 1. The iterative nature of the engineering design process [2].

Section B. Engineering Design Requirements

This section describes the goals, objectives, and constraints of the mainframe curriculum design. These requirements are derived from the needs identified of industries that rely on mainframes, particularly z/OS and COBOL. The design is structured to address the current skills gap and ensure scalability, usability, and industry relevance.

B.1 Project Goals (i.e. Client Needs)

The overall goals of this project are to address the industry wide gap in mainframe skills and provide a comprehensive, accessible, entry level education on legacy technologies. From the client's perspective, the curriculum must:

- Bridge the knowledge gap for developers inexperienced with z/OS and COBOL.
- Provide a flexible learning structure, to allow learners to understand the fundamentals of mainframe technologies and build upon them.
- Be adaptable for multiple levels of expertise, from beginners to seasoned developers that are transitioning into mainframe work.
- Be easily scalable for broader adoption across different industries and academic institutions.

B.2 Design Objectives

The curriculum design must achieve the following SMART (Specific, Measurable, Achievable, Reasistic, and Time bound) objectives:

- The curriculum will provide specific learning modules covering z/OS and COBOL programming, focusing on fundamental concepts, key commands, and system interaction.
- Success will be measured by learner completion rates, module progress, and coding exercises.
- The curriculum is designed for online, self paced learning, allowing participants to complete the course on their own schedule, which makes it accessible to both current professionals and students.
- The content will be created using readily available resources provided by Dr. Dahlberg, leveraging existing online platforms and our faculty advisors' input.
- The curriculum will be developed over the course of 12 weeks, with regular meetings and updates to ensure quality and relevance.

B.3 Design Specifications and Constraints

The following specifications and constraints apply to the curriculum development process:
- The curriculum will cover the basics of z/OS, COBOL, and mainframe system management. Advanced topics will be included in later modules (potentially optional) ensuring flexibility for different skill levels.
- User data, such as quiz results and course progress will be handled according to relevant academic data protection standards.
- Course content must be structured in a way that allows for regular updates as reorganization is necessary following feedback. This ensures the curriculum optimizes over time.
- This course should not need any further purchases from both the user perspective as well as the educators to maximize accessibility to the curriculum.
- The curriculum will be designed to work with industry standard software and platforms. It must integrate smoothly when integrating with Learning management systems like Canvas.

B.4 Codes and Standards

The development of the curriculum must adhere to the following codes and standards to ensure quality, interoperability, and legal compliance:

- IEEE Standard No. 610 - The curriculum content will align with software engineering standards to ensure proper development of technical educational materials.
- Potentially WCAG 2.1 - The curriculum may meet accessibility standards for individual with disabilities, ensuring accessibility to learning resources.

This concludes Section B, outlining the goals, objectives, and constraints needed for developing the mainframe curriculum.

Section C. Scope of Work

The project scope defines the boundaries of the project encompassing the key objectives, timeline, milestones and deliverables. It clearly defines the responsibility of the team and the process by which the proposed work will be verified and approved. A clear scope helps to facilitate understanding of the project, reduce ambiguities and risk, and manage expectations. In addition to stating the responsibilities of the team, it should also explicitly state those tasks which fall *outside* of the team's responsibilities. *Explicit bounds* on the project timeline, available funds, and promised deliverables should be clearly stated. These boundaries help to avoid *scope creep*, or changes to the scope of the project without any control. This section also defines the project approach, the development methodology used in developing the solution, such as waterfall or agile (shall be chosen in concert with the faculty advisor and/or project sponsor). Good communication with the project sponsor and faculty advisor is the most effective way to stay within scope and make sure all objectives and deliverables are met on time and on budget.

C.1 Deliverables

- None of the deliverables are needed through campus access
- All of the work can be done online. The only resources that are needed to work are shared drives/folders and certain software licenses to help learn about the mainframe
- The only deliverable that is needed from a third party-vendor is the software license. To mitigate supply chain disruptions we can persistently work on getting the license.

C.2 Milestones

Milestone	Estimated Time	Completion Date
Research on the mainframe	2-3 weeks	[Insert date]
Select important focus areas	1-2 weeks	[Insert date]
Plan out the course	2 weeks	[Insert date]
Create the course	3-4 weeks	[Insert date]

C.3 Resources

For the successful completion of the project, several key resources will be required. First, **mainframe guides** are essential for gathering the necessary information on the mainframe system. Additionally, **software licenses** will provide access to platforms that enable a deeper understanding of the mainframe, ensuring comprehensive learning. Finally, a **program** is needed to facilitate the creation of the course content. These resources are crucial for carrying out the research, learning, and course development phases of the project effectively. Should further tools or platforms be needed, they can be acquired as the project progresses.

Section D. Concept Generation

A number of methods can be used to help generate design concepts from simple reflection and brainstorming, to working the problem backwards, using reverse thinking techniques, and looking to nature for inspiration (i.e. biomimicry). Existing solutions, or components of existing solutions, can be substituted, combined, adapted, modified, put to other uses, eliminated, or rearranged to meet new design objectives and specifications. A minimum of 3 overall design concepts is required for this section although more are welcome. Provide a brief description of how each design concept addresses the design problem. Discuss the potential pros and cons, including and potential risks of failure, of each of these concepts.

It is likely that each design concept may consist of several components. In this case, one or more of these components may offer a sub-problem that can be further explored, modified, or otherwise improved upon. These sub-problems may lead to the addition of several additional design concepts and may require the inclusion of a design concept chart or matrix to organize all ideas and potential solutions.

Provide any initial design sketches, drawings, 3D renderings, or conceptual models such as dataflow diagrams, process flows, etc. developed during the concept ideation phase. All hand drawings should be drawn to scale using basic engineering drafting tools (i.e. ruler, protractor, and compass). Geometric stencils can also be used to help produce quality hand drawings. Drawings should be presented in a profession manner, preferably done on engineering graph paper and using a high-quality scan. All sketches should be labeled to identify major components and different drawing views or projections if applicable. Basic dimensions should be provided to give a general sense of scale. Label each sketch or drawing with the name of the team member responsible for the sketch, the date it was drawn, and the drawing scale.

Section E. Concept Evaluation and Selection

Using a systematic decision-making process, evaluate each of the design concepts and choose the one that is most likely to succeed in meeting the design objectives and constraints. A Decision Matrix, or Pugh Matrix, helps to analyze alternatives, eliminate biases, and make rational decisions through thought and structure. First, work to develop a set of selection criteria for which to evaluate the previously generated design concepts. Selection criteria often include concepts of performance, cost, safety, reliability, risk, etc. Note that the selection criteria developed here will likely be more general than the project design objectives. As with the design objectives, conversations with the client help define appropriate selection criteria.

In many cases, the client may value the selection criteria differently, preferring that more emphasis be placed on some than others. In this case, weighting factors may be used to place more or less importance on the various criteria in the decision making process. Again, conversations with the client can be used to define criteria weighting factors. Often times, these conversations must be analyzed and interpreted by the team to determine which criteria are more important to the client and by how much. Feel free to discuss the assigned weighting factors with the client to see if they seem accurate.

Next, define an associated metric to represent each criteria. Metrics should be specific and quantifiable, providing numerical values that quantify the often vague concepts of the selection criteria. Metrics can be obtained, generated, or estimated through a number of methods including simple background research, preliminary design calculations, or basic analyses. Note that these metrics do not need to specifically align with the design specifications although there may be some commonality between the two. Provide a brief discussion of the rationale for selecting each of the assigned metrics.

Using the defined metrics, evaluated each design concept against all selection criteria by filling out a Decision Matrix. Design concepts can be compared by using simple rank scoring, raw scoring, or weighted scoring techniques and design concept with which to move forward can be selected. This type of process provides a meaningful, unbiased means for choosing a preliminary design concept prior to moving forward with more comprehensive, detailed analyses as provided in the design methodology section below. The results of this process should be discussed with the project client prior to moving forward with the selected design. Table 1 provides an example of a simple decision matrix.

Table 1. Example of a Decision Matrix.

	Design Concept A	Design Concept B	Design Concept C	Design Concept D
Criteria 1				
Criteria 2				
Criteria 3				
Criteria 4				
Criteria 5				
Total Score				

Note: Weights can be assigned to each criterion if desired.

Section F. Design Methodology

Provide a detailed explanation of the methods that will be used to help evaluate, improve, and evolve the design through the iterative engineering design process. Consider that ultimately, the final design must be verified and validated to ensure that it meets all of the previously developed and listed design objectives and specifications. Verification ensures that the design meets all specification, while validation confirms that the design functions as intended such to meet the client's needs. While it is common for initial design concepts to first be evaluated using simplified design criteria and metrics, the chosen design should be advanced, and later verified, using engineering calculations, computational models, experimental data, and/or testing procedures.

Use this section to describe any underlying physical principles and mathematical equations that govern the design. Provide details of any computer-aided modeling techniques used to evaluate the design including the software used, prescribed boundary conditions, and assumptions. Include a detailed description of any experimental testing methods including required testing equipment, test set-up layout, data acquisition and instrumentation, and testing procedures. If one or more prototypes is to be produced and tested, provide a detailed description of how each will be evaluated.

Note: The contents of this section are expected to vary from project to project. Subsections may be appropriate for providing details of analytical, computational, experimental, and/or testing methods. Some potential subsections that may be included in this section are provided. While critical design equations may be provided here, lengthy mathematical derivations may be included in an appendix. Validation procedures are critical and all projects should address such topic.

F.1 Computational Methods (e.g. FEA or CFD Modeling, example sub-section)

F.2 Experimental Methods (example subsection)

F.3 Architecture/High-level Design (example subsection)

F.5 Validation Procedure

Describe how the design team will validate that the final design meets the client's needs. This section should include a plan to meet with the client towards the end of the project to discuss final design details and demonstrate a prototype, experimental test, and/or simulation results. Provide a relative time frame for this validation to occur (e.g. "mid-March" or "early-April"). Include a brief discussion on how client feedback will be captured, such as a formal survey, interview, or observation notes of the client using the prototype. It may also include plans to solicit feedback from other stakeholders and/or potential users.

Section G. Results and Design Details

Use this section to highlight the major results of the design methodology described above including important analytical, computational, experimental, modeling, assembly, and testing results. This section should be one of the most substantial sections of the report showcasing all of the hard work and effort that went into the completion of the final design and delivery of the project deliverables. Show how the identified problem was solved.

Highlight the prominent features of the final design through analysis results, modeling, drawings, renderings, circuit schematics, instrumentation diagrams, flow and piping diagrams, etc. to show that the design functions as intended and meets all design objectives and constraints. Overview designs such as dataflow diagrams, process flow, swim lane diagrams, as well as presentation-layer designs (e.g. storyboards for front-ends) should be included here. Detailed designs such as database designs, software designs, procedure flowcharts, or pseudocode should be included here. Support computational and experimental results with key plots and figures. All supporting figures should be clearly labeled and annotated to highlight the most important points of the figure (i.e. explicitly point out what the reader should focus on or understand about the image).

Note that while all results should be used to help inform design decisions, not all results may be necessary to include in the main body of the report. Extraneous supporting results (e.g. graphs, data, design renderings, drawings, etc.) that are not necessary for presenting the fundamental findings can be placed in one or more appendices. Detailed documentation of each program module can be provided as appendix.

G.1 Modeling Results (example subsection)

G.2 Experimental Results (example subsection)

G.3 Prototyping and Testing Results (example subsection)

G.4. Final Design Details/Specifications (example subsection)

Note that while the design constraints and specifications may have provided minimum or maximum values, or ranges or values, that the design needed to meet, the final design specifications should be listed here showing that the required design values were met. A list of final design details can also be included demonstrate fulfillment of the design objectives.

Note: Preliminary results should be included in the Preliminary Design Report to show the progress made of the selected design concept to-date. This section should be updated for the Final Design Report to include documentation of all of the work that was completed on the project throughout the entirety of the academic year.

Section H. Societal Impacts of Design

In addition to technical design considerations, contemporary engineers must consider the broader impacts that their design choices have on the world around them. These impacts include the consideration of public health, safety, and welfare as well as the potential societal, political/regulatory, economic, environmental, global, and ethical impacts of the design. As appropriate for the project design, discuss how each of these considerations influenced design choices in separate subsections. How will the design change the way people interact with each other? What are the political implications of the design? Does the technology have the potential to impact or shift markets? Does the design have any positive or negative effects on the environment? Don't forget to consider unintended consequences such as process or manufacturing byproducts. What impacts might the design have on global markets and trade? Are there any ethical questions related to the design?

While it is hard to forecast the various impacts of a technology, it is important to consider these potential impacts throughout the engineering design process. When considered during the early stages of the design phase, consideration of these impacts can help determine design objectives, constraints, and specifications and help drive design choices that may mitigate any potential negative impacts or unintended consequences.

Note: A minimum of 4 of these design considerations, including the consideration of public health, safety, and welfare, are required for the Preliminary Design Report while a section for all considerations must be included in the final design report.

H.1 Public Health, Safety, and Welfare

Provide a list of all design safety features and provide a brief description of each. Discuss the potential effects the design may have on public health, safety, and welfare. References to the codes and standards previous provided and the organizations that produced them may be summarized or referenced here.

H.2 Societal Impacts

H.3 Political/Regulatory Impacts

H.4. Economic Impacts

H.5 Environmental Impacts

H.6 Global Impacts

H.7. Ethical Considerations

Section I. Cost Analysis

Provide a simple cost analysis of the project that includes a list of all expenditures related to the project. If an experimental test set-up or prototype was developed, provide a Bill of Materials that includes part numbers, vendor names, unit costs, quantity, total costs, delivery times, dates received, etc. Do not forget to include all manufacturing costs incurred throughout the completion of the project. If the design is expected to become a commercial product, provide a production cost estimate including fixed capital, raw materials, manufacturing (including tooling and/or casting), and labor costs to produce and package the device. Note that this type of detailed cost analysis may be listed as a project deliverable.

Note: The Preliminary Design Report should include all costs incurred to date. It is expected that this section will be expanded and updated between the preliminary and final design reports.

Section J. Conclusions and Recommendations

Use this section to summarize the story of how the design team arrived at the final design. Focus on the evolution of the design through the use of the engineering design process including lessons learned, obstacles overcome, and triumphs of the final design. Revisit the primary project goals and objectives. Provide a brief summary of the final design details and features paramount to the function of the design in meeting these goals and objectives.

A discussion may be included to discuss how the design could be further advanced or improved in the future. If applicable, summarize any questions or curiosities that the final results/design of this effort bring to mind or leave unanswered. If this project might continue on as a future (continuation) senior design project, detail the major milestones that have been completed to date and include any suggested testing plans, relevant machine drawings, electrical schematics, developed computer code, etc. All relevant information should be included in this section such that future researchers could pick up the project and advance the work in as seamless a manner as possible. Documents such as drawings, schematics, and codes could be referenced here and included in one or more appendix. If digital files are critical for future work, they should be saved on a thumb drive, external hard drive, cloud, etc. and left in the hands of the project advisor and/or client.

Appendix 1: Project Timeline

Provide a Gantt chart of similarly composed visual timeline showing the start and end dates of all completed tasks and how they are grouped together, overlapped, and linked together. Include all senior design requirements including design reports and Expo materials (i.e. Abstract, Poster, and Presentation). All major milestones should be included in the timeline.

Appendix 2: Team Contract (i.e. Team Organization)

Step 1: Get to Know One Another. Gather Basic Information.

Task: This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

Team Member Name	***Strengths each member bring to the group***	***Other Info***	***Contact Info***
Arbab Arif	Friendly, 2X SWE Intern Experience (Web Developer)	Interned at CoStar for Full Stack Web Development. Web Stack is React, Express.js/Node.js, .Net(C#), SSMS SQL	arifa6@vcu.edu, arifarbab21@gmail.com, (703)-901-8713
Long Le	Proactive, problem-solving, hard worker, and inclusivity	Worked on a QA team this past summer	lelt8@vcu.edu 804-536-8982
Osman Zafar	Planning, debugging, and industry experience	Interned for the Federal Reserve System for 1.5 years	zafarom@vcu.edu 571-294-6737

Other Stakeholders	***Notes***	***Contact Info***
Bob Dahlberg	*Weekly Zoom meetings on Tuesdays*	**dahlbergra@vcu.edu**

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

Task: Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team's culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

Resources: More information and an example Team Culture can be found in the Biodesign Student Guide "Intentional Teamwork" page (webpage | PDF)

Culture Goals	***Actions***	***Warning Signs***
Ensure each team member shares their thoughts about the project and how to proceed and communicates	- Ask each other questions - Speak up when you have any thoughts or concerns about the project - communicates if they're not going to make a meeting - communicate if they have stuff going on and need help	- A team member does not speak in a meeting - A team member not responding to team messages for a long period of time
Agile Development cycle	- Create overall end goal - create sub goals that lead towards that direction - create demos to see if we need to pivot and change structure	- Not have any plan - Not make any demos - Feel confused on direction of project
		-

Step 3: Time Commitments, Meeting Structure, and Communication

Task: Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will same person take on that role for the duration of the project?

Required: How often you will meet with your faculty advisor advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?
See examples below.

Meeting Participants	***Frequency Dates and Times / Locations***	***Meeting Goals Responsible Party***
Students Only	When needed in Discord and over Discord messages and phone messages	Inform one another about important project details/timeline/obstacles
Students Only	Every Thursday during CMSC 451 time	Work on project and keep one another updated on each individual's progress
Students + Faculty advisor	Every Tuesday from 10-11 am in Zoom	Discuss project goals, give updates to Professor Dahlberg, and ask the professor our questions.

Step 4: Determine Individual Roles and Responsibilities

Task: As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my "natural" role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

Communication Leaders

Suggested: Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

Suggested: Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.
   Required: On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.
2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.
5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.
6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

Team Member	***Role(s)***	***Responsibilities***
Long	Test Engineer	Make sure the work is running according to plans Looks for any defects or bugs
Osman	Logistics Manager	Communicate with Professor Dahlberg Obtain information needed for the project
Arbab	Project Manager	Oversee entire Project Make Timeline and schedule Coordinate meetings Keeps team on track and moving towards goals

Step 5: Agree to the above team contract

Team Member: Long Le *Signature: Long Le*

Team Member:Osman Zafar *Signature: Osman Zafar*

Team Member:Arbab Arif *Signature: Arbab Arif*

Appendix 3: [Insert Appendix Title]

Note that additional appendices may be added as needed. Appendices are used for supplementary material considered or used in the design process but not necessary for understanding the fundamental design or results. Lengthy mathematical derivations, ancillary results (e.g. data sets, plots), and detailed mechanical drawings are examples of items that might be placed in an appendix. Multiple appendices may be used to delineate topics and can be labeled using letters or numbers. Each appendix should start on a new page. Reference each appendix and the information it contains in the main text of the report where appropriate.

Note: Delete this page if no additional appendices are included.

References

Hartman, T. (n.d.). Cobol blues. Reuters. https://fingfx.thomsonreuters.com/gfx/rngs/USA-BANKS-COBOL/010040KH18J/index.html#:~:text=On%20average%2C%20COBOL%20programmers%20are,between%2045%2D55%20years%20old.

Open Mainframe Project. (2021, May 1). Why have colleges and universities stopped teaching mainframe skills? https://openmainframeproject.org/blog/why-have-colleges-and-universities-stopped-teaching-mainframe-skills/

Bajkowski, J. (2019, October 18). Massive CBA outage traced to failed infrastructure upgrade. iTnews. https://www.itnews.com.au/news/massive-cba-outage-traced-to-failed-infrastructure-upgrade-532586