**FINAL REPORT:**

Plan, Analysis, Design, Development, Test, and Implementation

|  |  |
| --- | --- |
| **Team Number and Name:** | Team 1, HAX Microservice |
| **Document Version:** | 2.0 |
| **Document Date:** | 04-May-2020 |

**AUTHORS**

This document was prepared by:

|  |  |
| --- | --- |
| **Report Section** | **Author Name And Title** |
| 1.0 Report Purpose | Nick Johnson, Robert Sonnelitter, Joe Wanat, Bryce Coppadge, Shraddha Venkatraman |
| 2.1 Plan – Stakeholder Support | Nick Johnson, Robert Sonnelitter, Joe Wanat, Bryce Coppadge, Shraddha Venkatraman |
| 2.2 Plan – Project Risk | Nick Johnson, Robert Sonnelitter, Joe Wanat, Bryce Coppadge, Shraddha Venkatraman |
| 2.2 Plan – Project Management | Bryce Coppadge |
| 3.1 Analysis – User Research | Joe Wanat, Robert Sonnelitter |
| 3.2 Analysis – User Requirements Specification | Nick Johnson, Joe Wanat |
| 3.3 Analysis – Business Process Models | Joe Wanat, Bryce Coppadge |
| 4.1 Design – Technical Specifications | Robert Sonnelitter |
| 4.2 Design – Design Tools | Bryce Coppadge, Shraddha Venkatraman |
| 5.1 Development – Development Plan | Robert Sonnelitter |
| 5.2 Development – Final System | Robert Sonnelitter |
| 6.1 Test – Requirements Traceability Matrix | Joe Wanat |
| 6.2 Test – Test Scripts | Nick Johnson, Joe Wanat |
| 6.3 Test – Test Results | Nick Johnson |
| 7.1 Implementation – Transition Plan | Joe Wanat, Robert Sonnelitter |
| 7.2 Implementation – Training Guides | Joe Wanat, Bryce Coppadge, Robert Sonnelitter, Nick Johnson |
| 8.0 References | Nick Johnson, Robert Sonnelitter, Joe Wanat, Bryce Coppadge, Shraddha Venkatraman |
| 9.0 Appendix | Nick Johnson, Robert Sonnelitter, Joe Wanat, Bryce Coppadge, Shraddha Venkatraman |

**VERSION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Document Version** | **Document Revision History** | **Document Author or Reviser** |
| 04-May-2020 | 2.0 | Second Version (Final Report) – Added Design, Development, Testing, and Implementation sections | Project Team |
| 01-Mar-2020 | 1.0 | Initial Version (Mid Semester Report) | Project Team |

**APPROVALS**

The following teammates are responsible for the completeness and accuracy of this report:

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Document Version** | **Approver Name and Title** | **Approver Signature** |
| 04-May-2020 | 2.0 | Bryce Coppadge, Project Manager | BryceA.Coppadge |
| 04-May-2020 | 2.0 | Robert Sonnelitter, Development and  Implementation Lead | Robert Sonnelitter |
| 04-May-2020 | 2.0 | Nick Johnson, Testing Lead | Nick Johnson |

**1** **REPORT PURPOSE**

This final report provides a summary for all of the System Development Life Cycle (SDLC) phases for the **HAX CMS Microservice** project. This includes the following:

· **Plan:** Updates on stakeholder support and project risk, and an overview of the team’s project management tools

· **Analysis:** A summary of user research, user requirements specification (URS), and business process models (BPMs)

· **Design:** A summary of the preliminary brainstorming and evaluation of possible design solutions to prepare for the design phase

· **Development:** A summary of the development plan and a description of how to access the final system design

· **Test:** The system test scripts, requirements traceability matrix (RTM), and a summary of the testing results

· **Implementation:** A summary of the implementation and/or transition plan and the system training guides

**2** **PLAN PHASE**

**2.1** **Stakeholder Support**

Table 2.1 describes each stakeholder that is involved with our project. We have a total of five stakeholders with a specified level of importance. The importance level is based on their impact on the project and to what degree we will rely on them for our project’s success.

**TABLE 2.1.1 Analysis of HAX CMS Stakeholders**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Importance level (1-5)** |
| Bryan Ollendyke | Lead developer of HAX CMS | 5 |
| Michael Potter | Member of Elmsln that is experienced with docker | 4 |
| Elmsln | Developer group that owns HAX CMS | 3 |
| Users of HAX CMS | Those who use HAX CMS | 4 |
| Alison Murphy | General guidance on the project | 4 |

**Bryan Ollendyke:** Bryan Ollendyke is the lead developer of HAX CMS and our main client for this project. He is a developer activist so he is highly technical which will give us a big advantage while creating the microservice. He isn’t the most knowledgeable on Docker from his group, but he specializes in web development which will make him the go to person when we have questions regarding HAX itself. Mr. Ollendyke has indicated that he would have meetings over Zoom to assist with our project after the it went remote.

**Michael Potter:** Michael Potter is a member of Mr. Ollendyke’s group that works directly with Mr. Ollendyke on the development of HAX CMS. Mr. Potter’s main area of expertise is with Docker and microservices in general making him a great resource for any Docker related questions that Mr. Ollendyke can’t answer. He was a great resource for the development phase of this project and helped troubleshoot an issue we ran into.

**Elmsln:** Elmsln is the name of the group that Mr. Ollendyke works with on HAX CMS. They communicate with each other through their personal Slack workspace that we have access to. Should Bryan Ollendyke and Michael Potter be unavailable for us to contact, we can turn to the rest of Mr. Ollendyke’s group for assistance on this project. The group is known for their development knowledge and will certainly be useful for us should we need to contact any of them. We did not need to interact with Elmsln developers as Mr. Olesyke and Mr. Potter have provided us with all the information we needed.

**Users of HAX CMS:** Users of HAX CMS will be who we mainly turn to for when we want feedback about this kind of feature. The students and faculty of Penn State are the people who would most benefit from this microservice. IST 440 student, Andrew Wertz, was one of the students selected to test the microservice. The input of students, such as Mr. Wertz, helps us greatly understand both the importance of the project to the user as well as give us much-needed feedback during the testing phase. Other possible people to utilize in the future include respondents of our user requirements survey.

**Alison Murphy:** Professor Murphy, while being the instructor of the class, will also serve as a great resource for this project. Her knowledge on projects and project management as a whole will help us ensure we stay on the right track throughout this project. Her feedback helps to ensure that the quality of the project is high and all documentation is up-to-par. She has made herself available via Zoom to ask for any help with the project while working on this project remotely.

To maximize the success of our project, we need to ensure that our targeted stakeholders are at the very least supportive of the project in some way. The following table shows where we believe the current level of support to be and where we deem it necessary to be by the end of the project.

**TABLE 2.1.2 Stakeholder Support Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stakeholder** | **Actively Opposed** | **Passively Opposed** | **Neutral** | **Passively Supportive** | **Actively Supportive** |
| **Bryan Ollendyke** |  |  |  |  | XO |
| **Michael Potter** |  |  |  | X | O |
| **Elmsln** |  |  |  | X | O |
| **Users of HAX CMS** |  |  | X | O |  |
| **Allison Murphy** |  |  | X | O |  |

**Key**

X = Current level of stakeholder support

O = Where stakeholder support level needs to be for project success

**2.2** **Project Risk**

The following risk register (Table 2.2) is used to assess the probability and impact of the top four risks we identified for our project. Each risk has a score summarizing its severity as well as a potential response for if that risk were to occur. Below the table, each risk described in additional detail.

**TABLE 2.2 HAX CMS Microservice Risk Register (Version 4)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk**  **ID** | **Risk** | **Cause** | **Probability (1-5)** | **Impact (1-5)** | **Risk**  **Score (1-25)** | **Response** | **Owner** |
| R1 | Time Constraints | Running out of time for overall project to meet deliverables | 4 | 5 | 20 | Mitigate by continuing to meet at least twice a week with our group to stay on schedule. | Project Manager,  Bryce Coppadge |
| R2 | Lack of Development Support | Developers from Elmln are unable to provide support | 2 | 3 | 6 | Accept by attempting to complete all tasks without assistance, but still attempting to reach out if needed | Development Lead,  Robert Sonnelitter |
| R3 | GitHub Change Compatibility | Pushed change destroys original successful code | 2 | 2 | 4 | Mitigate by referencing previous documented code to return altered code to function | Development Lead,  Robert Sonnelitter |
| R4 | Stray from Original Scope | Scope creep causes team to take on more or less deliverables than intended initially | 3 | 1 | 3 | Accept by submitting completed work to stakeholders and allow them to take lead if not completed | Project Manager, Bryce Coppadge |
| R5 | Remote Collaboration | No in-person meetings may decrease productivity and effectiveness of work | 2 | 3 | 6 | Mitigate by increasing number of virtual Zoom meetings to foster heightened communication levels | Project Manager,  Bryce Coppadge |

**Time Constraints (R1):** Since this is a semester long project, we will be working against a running clock. In the scenario in which we are unable to finish within the semester, we will hand-off the code we had to the primary client, Mr. Bryan Ollendyke. This will mean that we will have to remove multiple user requirements. To combat this risk, we will constantly meet outside of class to maintain a good pace throughout the semester. While we were able to complete the original goal of creating a microservice for HAX CMS, we were unable to complete our stretch goal of fully implementing the microservice. Instead, we will pass off all the work we have completed to Bryan Ollendyke for HAX CMS to implement into the site.

**Lack of Development Support (R2):** The development team along with Bryan Ollendyke and Michael Potter have offered to assist our team if we run into any issues on the development side. If we have problems and they are unavailable to offer help we could potentially be stuck on a problem and unable to move forward. A lack of developmental support will tack on an extra hour of troubleshooting for each problem that we run into. If we do not have support from the team, we will adjust the user requirements to eliminate what we could not complete. Should this happen, we will collaborate on the issue and brainstorm on any fixes. Another medium we could possibly look to for help would be online forums such as Stack Overflow. There have been no issues relating to this risk because we received assistance as needed.

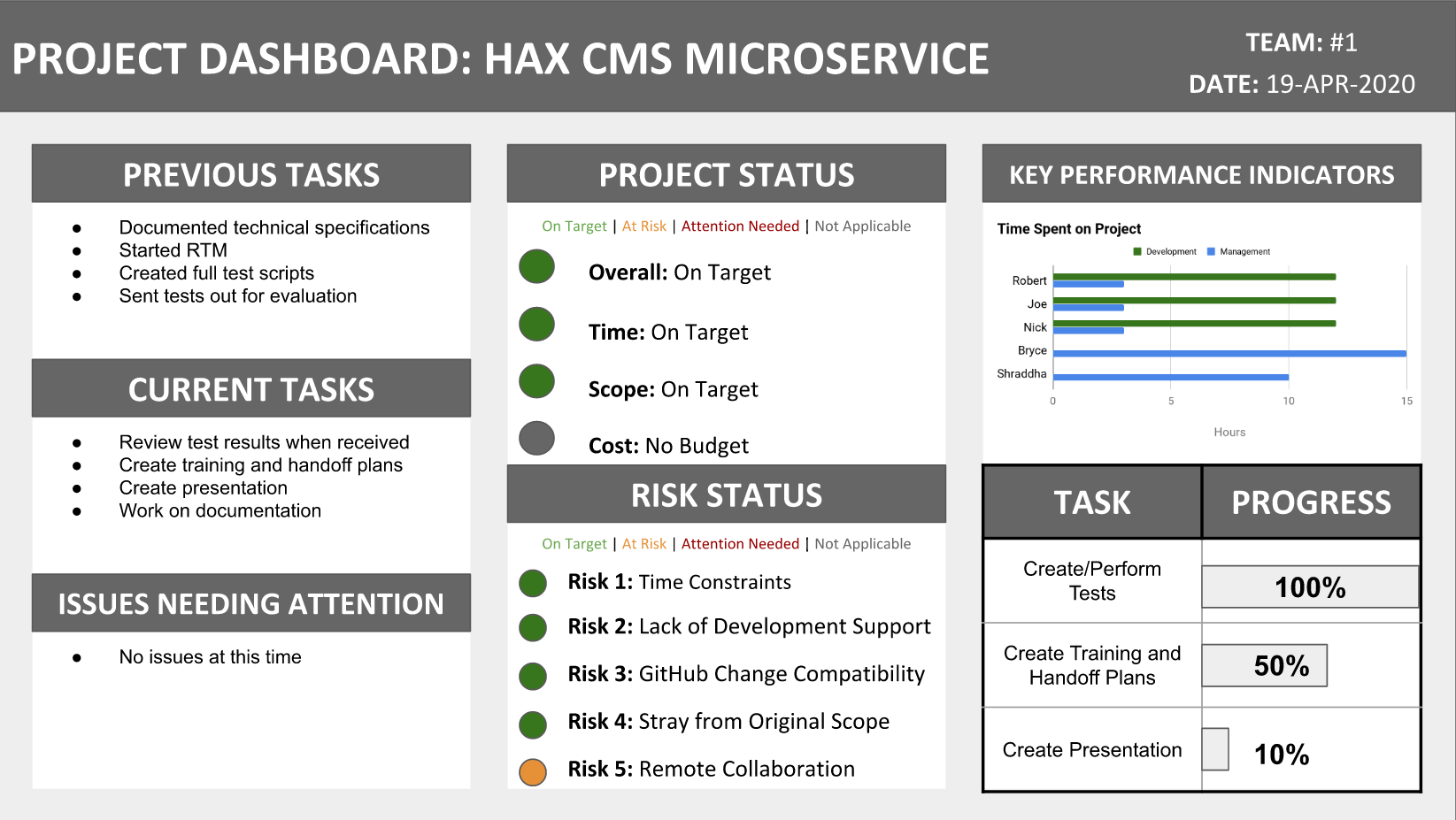
**GitHub Change Compatibility (R3):** Throughout the semester we will be documenting and storing our code on GitHub. In this scenario, a pushed change could potentially alter or damage the code we previously had saved. If the change we made breaks the code we had saved, we will reference the code we previously documented to change it back to normal. A change in the code would take roughly one hour to look through and fix. Fortunately a risk like this would not affect any of the user requirements as we would simply identify the change and revert it back to the original code. Although initially anticipated to be a larger problem, we actually had zero issues associated with this risk. We had no issues because many, if not all, problems within code were discussed during meetings and fixed in real time locally. Thus, changes never had to be pushed over GitHub and avoided potential conflicts.

**Stray from Original Scope (R4):** Given that we will be in constant communication with our stakeholders, scope creep is a risk that we are aware of. An increase of goals or requirements from the stakeholders can alter the direction that this project will go in. A change midway through the project could stall and prevent us from completing in time. An increase in tasks needed to complete the project will add on hours to the project and we may be unable to complete it in time. Scope creep will force the team to eliminate many of the user requirements listed beforehand. The scope has stayed the same throughout the project and did not add on deliverables this semester.

**Remote Collaboration (R5):** Due to the ongoing pandemic, the group has had to resort to remote collaboration for the safety of everyone. This may lead to a decrease in productivity among members of the team. This decrease in work may lead to upwards of seven extra hours of work to catch up. It may also cause the team to be unable to fulfill all of the user requirements stated in the beginning of the project. To prevent any roadblocks or miscommunications from remote collaboration, we will have an increase of meetings each week to ensure we are all on the same page. Luckily, the increase in weekly meetings was a success as the entire team continued to complete their share of the work delegated.

**2.3** **Project Management**

The project team has been creating weekly dashboards similar to the one depicted in Figure 2.3 to give a visual understanding of where the group stands in relation to the project as a whole. Mr. Ollendyke is being contacted weekly with updates to what the group is currently working on as well as any potential issues. The status report shows our recently completed tasks, current tasks and issues we have encountered. We have been keeping track of overall time spent on the project by each team member. Although there may be some overlap in the future, the technical leads spend their time primarily on development while the management leads work on project management and documentation tasks. Additionally, our status reports keep track of our three main tasks currently and approximately how far we have gotten.



***Figure 2.3*** *Example Dashboard/Status Report*

In order to have an understanding of the path going forward, we created a gantt chart using Google Sheets. The Gantt chart displays a timeline of the events and tasks needed to complete this project. Each key part has tasks within them that also contain task owners. Google Sheets allowed the entire project team to collaborate on the gantt chart simultaneously. The Gantt chart can be found here: <https://docs.google.com/spreadsheets/d/1AaY094vEGH5rYylfwvglIaLXrSpUoWVFZwRSQB4ud4U/edit?usp=sharing>

The project team utilizes multiple applications and tools in order to conduct our work. Slack was chosen as our main communication due to its ability to act as both a messenger tool as well as its ability to easily share code with other group members. Zoom will also be used for video calls to work on the project remotely with other team members. Google Drive was chosen as our main tool for documentation, whether that comes in the form of a Google Doc, Slide, Sheet, or Form. Drive was selected because it allows the team to easily collaborate in real-time on any given document. Finally, the project team chose to use GitHub in order to co-develop with one another and manage version control along the way.

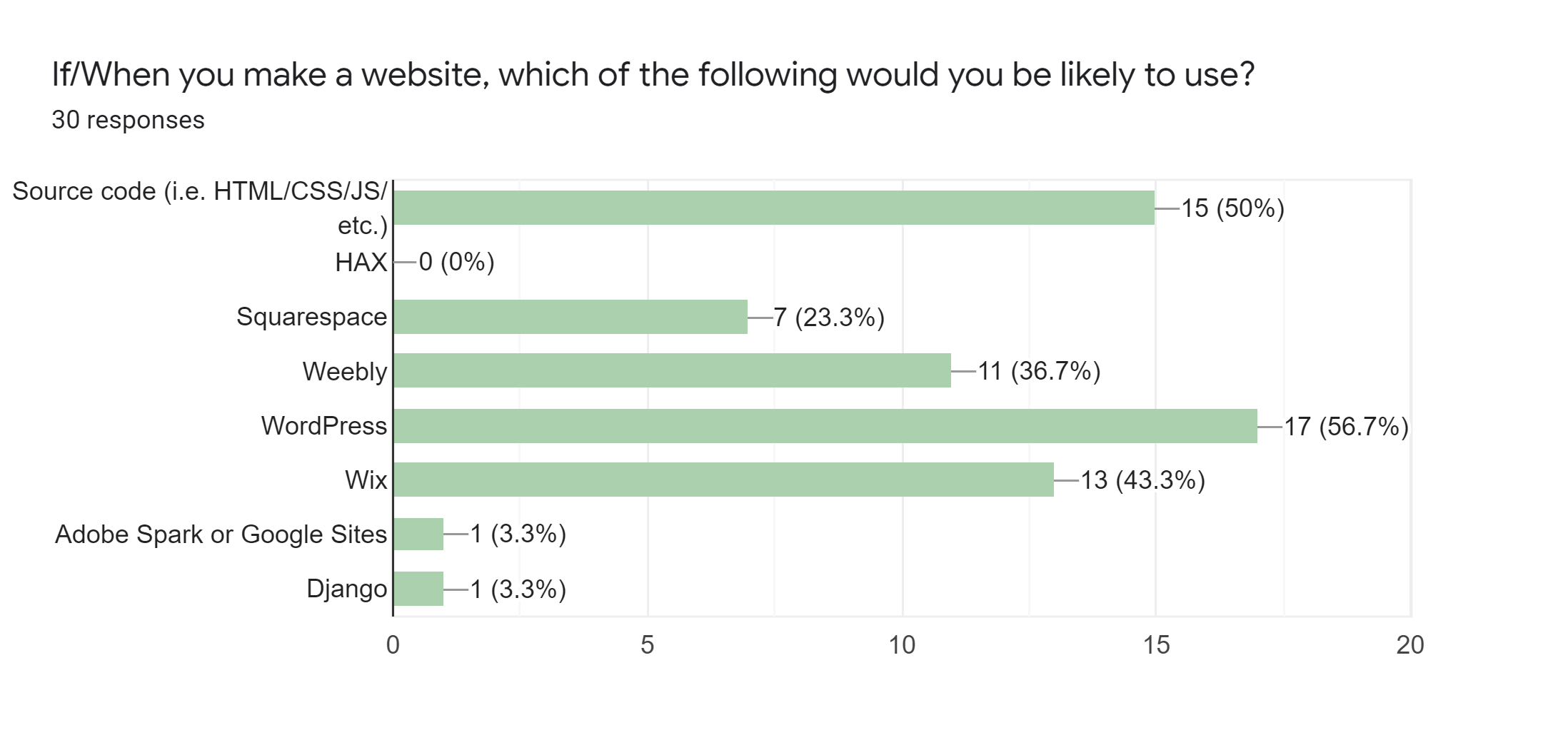
**3** **ANALYSIS PHASE**

**3.1** **User Research**

In order to have a better feel for the product we are developing, we decided to conduct various types of research to ensure the best possible solution. Initially, we developed a survey to send out to potential users. Next, we looked at relevant legal information to the project. Finally, we took a look at our client’s vision for this microservice in his product (HAX CMS) as a whole and how it may lead to future projects as well.

A [survey](https://forms.gle/2WjCYs3atdjRMUBw5) was sent out to a large body of Penn State students representing all classes and colleges. We received 30 responses throughout the polling period. The [responses](https://docs.google.com/spreadsheets/d/1suDtXxk2S1Au3Uo51_d2PShAq5NavWmVBT28tjTFxJY/edit?usp=sharing) showed that only one of the 30 participants have heard of HAX CMS previously. It is also worthy noting that this respondent did not answer that they would use HAX in the development of a website. As a result of the lack of experience with HAX, we have more freedom with our proposed solution due to not having to cater to prior expectations of the system.

Meanwhile, the most intriguing result, as depicted in Figure 3.1, shows that half of the respondents said that they are either familiar with and/or frequently use source code (HTML, CSS, JS) for web development. Unsurprisingly, WordPress, Wix, or Weebly were other popular tools for this task. We were able to achieve several insights from the results as well. Customizability and ease of use are two common themes amongst why certain options were selected. These themes remained consistent as many respondents stated that they want to see them in a file conversion tool such as the one we are developing for this project.



**Figure 3.1** Survey Result Depicting Most Popular Web Development

Tools Among Students.

In legal terms, our team cannot guarantee what material is being passed through the software. Our goal is that only material that is non-copyrighted or given permission for use and reuse be converted using the microservice. We will include a disclaimer explaining the legal consequences of using unauthorized materials in our product. In addition, rather than paying for the licensing of certain software or starting development completely from scratch, our project team searched GitHub and found an open source project containing a Docker container with Pandoc already installed. This container will allow the team to get a jumpstart on our project and not have to worry about the tedious details of initialization and configuration.

HAX CMS is Mr. Ollendyke’s attempt at competing with major Content Management Systems such as Drupal, WordPress, and Grav CMS. At the moment, HAX CMS is still in its very early stages and as such, relies on innovation to gain notoriety. Mr. Ollendyke aims for HAX to first replace sites.psu.edu which is Penn State’s WordPress-based CMS that students are allowed to use free of charge. Should HAX CMS reach this point, Mr. Ollendyke would then look to use HAX CMS as a possible replacement for Canvas. How he plans to do this is one of the major goals for both himself and his group, Elmsln. Their main goal is to discover new online learning tools, such as Open Educational Resources, and add features to HAX that leverages them. If HAX manages to reach a point where it could beat out Canvas as an educational management tool, Mr. Ollendyke would then proceed to offer it as a Software as a Service product. The microservice our team is tasked to create is intended to be the first hypothetical domino in a line of features HAX could encompass to eventually reach Mr. Ollendyke and Elmsln’s goals.

**3.2** **User Requirements Specification**

The following table (Table 3.2) contains both the functional and non-functional requirements for the HAX CMS Microservice. Each requirement is classified as either a want or need and is assigned to a task owner. FR-04 has been updated to fix the deliverable that had been asked for. Version one of this requirement requested that personas to be created. However, this deliverable was misinterpreted for its currently updated version of identifying use-cases for the system. Additionally, requirements FR-05 and NFR-02 have been determined to be out-of-scope for this project. FR-05 became out-of-scope because it involves creating a GUI that is no longer determined to be necessary. To incorporate the microservice into HAX, an adapter must be put into place between the current interface and the Docker container. NFR-02 was a stretch goal for the project and became out-of-scope because we did not have enough time after going remote to work with the Elmsln team to modify HAX to appropriately connect our microservice to it.

**TABLE 3.2 Functional and Non-functional Requirements for HAX CMS Microservice**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req-ID** | **Requirement Statement** | **Need or Want** | **Responsibility** |
| FR-01 | The microservice shall be capable of converting files between a defined set of inputs and outputs. | Need | Robert Sonnelitter,  Development Lead |
| FR-02 | The microservice shall be capable of presenting the file to the user after the conversion process. | Need | Robert Sonnelitter,  Development Lead |
| FR-03 | The microservice team shall create a handoff plan explaining how to integrate the microservice to the CMS. | Need | Bryce Coppadge, Plan Lead and Robert Sonnelitter, Development Lead |
| FR-04.1 | The project team shall identify and create use-cases for potential users of the microservice within the CMS. | Want | Joe Wanat, Analysis Lead and Shraddha Venkatraman, Design Lead |
| ~~FR-05~~ | ~~The project team shall create a graphic user interface for the user to utilize the microservice within the CMS.~~ | ~~Want~~ Out-of- Scope | ~~Shraddha Venkatraman, Design Lead~~ |
| NFR-01 | The microservice shall be ready for implementation by the end of April 2020. | Need | Robert Sonnelitter,  Implementation Lead |
| ~~NFR-02~~ | ~~The project team shall fully integrate the microservice into HAX CMS.~~ | ~~Want~~  Out-of- Scope | ~~Robert Sonnelitter,~~  ~~Implementation Lead~~ |
| NFR-03 | The microservice shall include a disclaimer warranting use to non-copyrighted material or copyrighted material with granted permission. | Want | Bryce Coppadge, Design Lead |
| NFR-04 | The microservice shall be free of any bugs and/or errors by release time. | Need | Nick Johnson,  Test Lead |
| NFR-05 | File conversions shall be successful a minimum of 97% of the time. | Need | Nick Johnson,  Test Lead |

**Key**

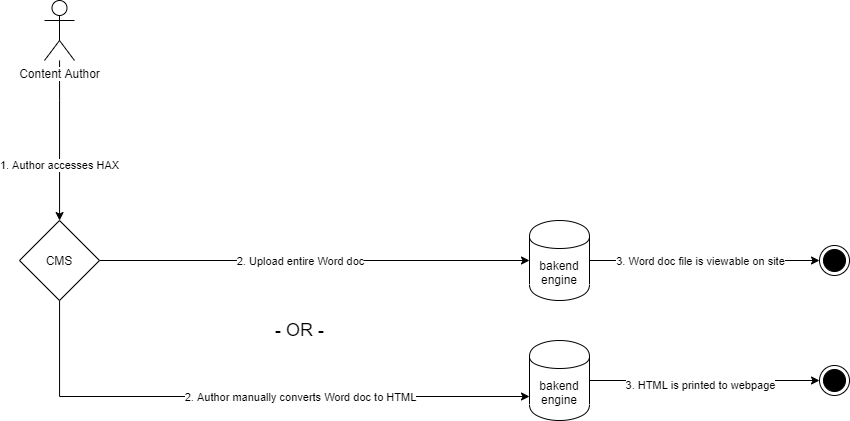
FR = Functional Requirement

NFR = Non-functional Requirement

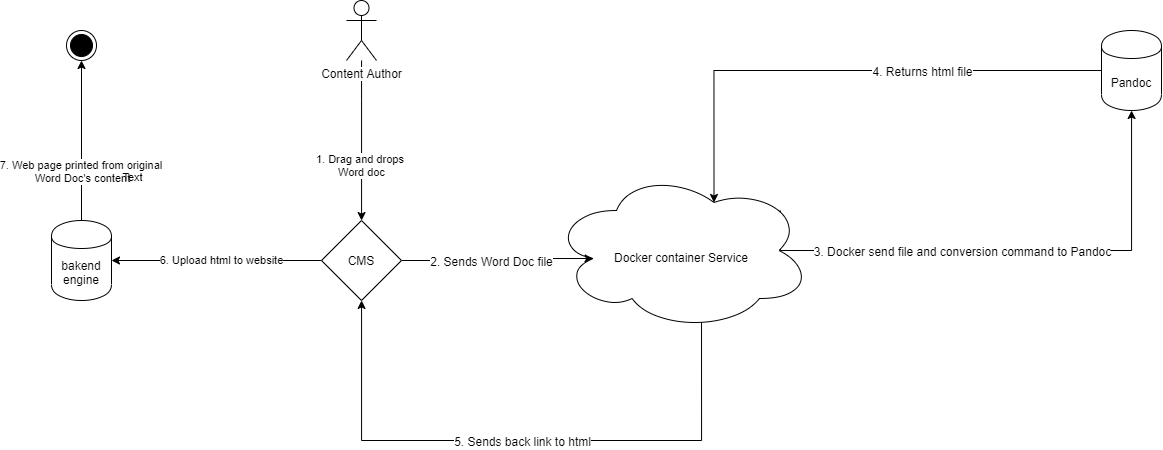
**3.3** **Business Process Models**

Business Process Models (BPM) allow us to create visual representations of important processes within our project. Specific use cases allow us to view scenarios that can assist in controlling our scope, identifying risks,and increasing organization. The two diagrams below allow a use case of converting a file and the file types that are available to be inputted and outputted.

Figure 3.3.1 displays the workflow for someone trying to display the contents of a Word document on HAX currently. A user must either upload the entire document or or convert the document into HTML themselves. Meanwhile, Figure 3.3.2 is a diagram that depicts the workflow of converting file types using the Pandoc-based microservice. The diagram provides an example of a user inputting a word document into HAX CMS. The file is then sent to the Docker container that runs the Pandoc framework. Pandoc then converts the file to a .html and returns it back to the user. This diagram is very important because it is the main feature of our project.



***Figure 3.3.1*** *Workflow for “Conversion” of File Types Without Pandoc-based Microservice*



***Figure 3.3.2*** *Workflow for Conversion of File Types Using Pandoc-based Microservice*

**4** **DESIGN PHASE**

**4.1** **Technical Specifications**

In order to fulfill our obligations for this project, the system needs to contain certain requirements specific to the microservice. These requirements include information primarily regarding system dependencies and software. This material can be found in Table 4.1.

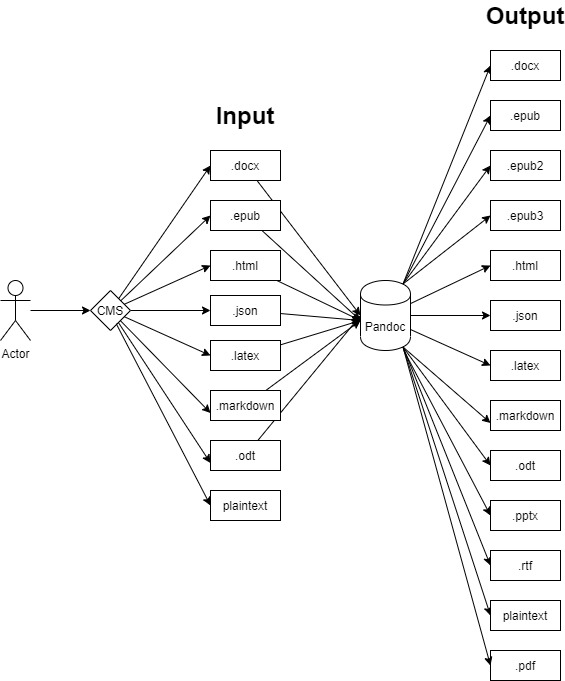
**TABLE 4.1 Technical Specifications**

The full table can be found here:

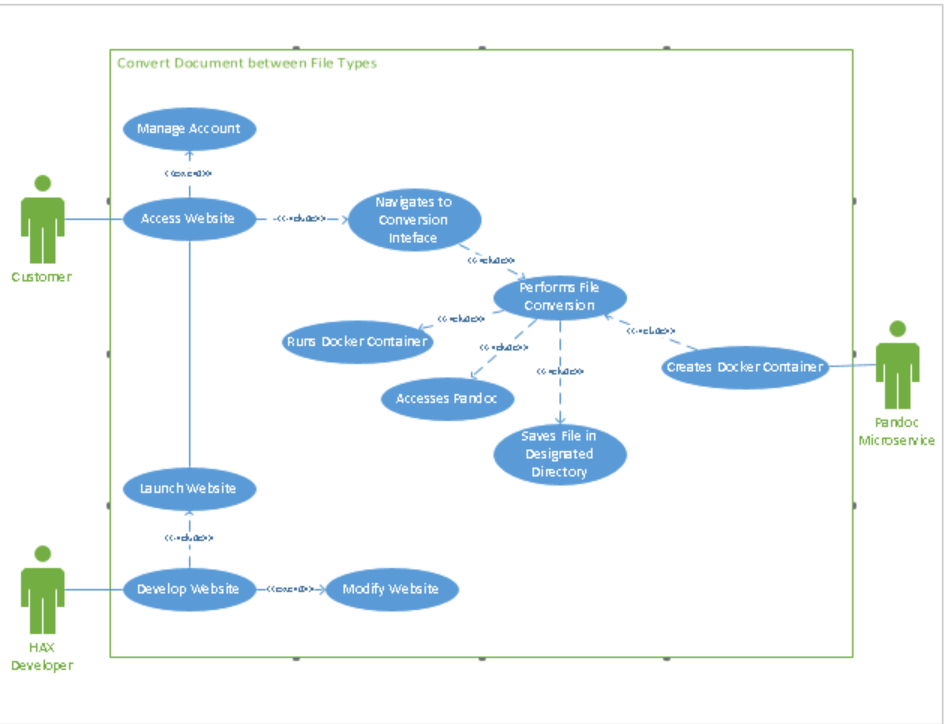
<https://docs.google.com/spreadsheets/d/15Ajvn9pETtdHrow7LM5Q2HdHMoGZjAsDpw3VvX76-v4/edit?usp=sharing>

**4.2** **Design Tools**

Figure 4.2.1 portrays all of the input to output file conversions that should be possible because of our microservice. All of the inputs have the possibility to be converted and exported as a new file type (the outputs). This is important because it represents all of the possibilities for our service. We received this information by understanding the capabilities of Pandoc and what file types it can produce.

***Figure 4.2.1*** *Input and Output File Types for Microservice*

This UML use case diagram in Figure 4.2.2 shows the interaction between the HAX developer and the customer, to capture and showcase the user requirements. Use cases are the description of the systems functionalities written in coherent order so that they can be related to respective actors.Actors can be considered internal, external applications or human users. The HAX developer and customer can be considered internal and human actors and the external application is the Pandoc microservice. The system model and development would again be the Pandoc microservice.



***Figure 4.2.2*** *Use Cases for File Conversion*

**5** **DEVELOPMENT PHASE**

**5.1** **Development Plan**

For our system, we went with a Coded System approach. We built our system off of the containerization service known as Docker. We used a command line based text editor like the nano tool to code our Dockerfile, and then compiled the file once it was written. Once we were able to get the container to build, figuring out what command to pass through was the next step. This obstacle was overcome with the help of Mr. Potter, and we were successfully able to get our Coded System to operate as expected.

Our plan entailed using a containerized instance of Pandoc that activates when a user of HAX CMS clicks on the convert prompt. The container, written on a Dockerfile and ran on the Docker Engine, takes in the file that the user wants converted. The file is then put through a command inside of the container where Pandoc is used to convert the file to whatever format the user desires. At the end of the process, the containerized process will be ended so the Docker container can cease to run and discontinue using resources on the back end.

The reason that this approach is believed to be the best is because it utilizes the Function as a Service (FaaS) model. Creating this FaaS allows back end resources to be used more efficiently, and significantly simplifies the conversion process. Previously, if one desired to convert a file, they had to go to a completely different site or use a special tool. This created additional work time for website creators as most other Content Management Systems like Wordpress don’t include this functionality. On the back end, a FaaS like this one guarantees that Pandoc’s conversion engine will always be available and is only run when requested. Since this microservice is built using Docker, we can specify what OS the container will run which will ensure that the server OS won’t ever run into compatibility issues. Additionally, the container is lightweight and never remains active following a conversion, allowing for a more efficient use of computational resources.

**5.2** **Final System**

Using the Technical Specifications described in Section 4.1, our team was able to put together a fully-functional file conversion microservice based on a Pandoc-enabled Docker container. Our final product can be found on GitHub through the following link: <https://github.com/elmsln/Pandoc-Service>. The GitHub has a brief ReadMe file to describe the functionality of the microservice at a very high level. Meanwhile, Section 7.2 highlights a Training Guide document to further explain the system. In addition, we have a [short video demonstration](https://drive.google.com/file/d/1vINFKtiucLmtLS1InWshmWnV6EBHQ_XE/view?usp=sharing) that provides some background on the system followed by a run-through of one example file conversion.

**6** **TEST PHASE**

**6.1** **Requirements Traceability Matrix**

The Requirements Traceability Matrix (RTM) is a table used to help determine the completion of various requirements throughout the project. The RTM includes the testing status, applicable incident reports, and associated documentation for each requirement. Section 3.2 describes the changes to scope of the requirements listed in this document.

**TABLE 6.1 Requirements Traceability Matrix**

The full table can be found here: <https://docs.google.com/spreadsheets/d/1eUMQe69ODpfh4xAMuT6MwqlPf_F3b-wILLxl8JlOwuY/edit?usp=sharing>

**6.2** **Test Scripts**

In order to ensure the reliability of the microservice, we tested five potential users using Test-01. We gave each of them a test script which included instructions on how to set up the lab. We used google sheets so we could receive the testing results in real-time. In addition Test-02 and Test-03 were used to test our non programmable functional and non-functional requirements. The blank test scripts can be found here:

<https://docs.google.com/spreadsheets/d/13Hz4ua2qsGodHiPuNGc9N_tnMZW1v7y7detV99Xsgeg/edit?usp=sharing>

**6.3** **Test Results**

In the test scripts, we color coded each tab to indicate the result of the test. Of the five tests ran, two were completed successfully (green). Three of the test subjects encountered user errors (yellow). These errors can all be found within the Incident Report. Red would have indicated a system failure with our team at fault.

In regards to the problems encountered, one user’s Play with Docker session had timed out so he could not complete the lab. The other two had trouble with the instructions to complete the test fully. All of the testers were able to reach step 6c in the script, proving that the container would at the very least build successfully. The completed test scripts can be found here:

<https://docs.google.com/spreadsheets/d/13grBHEQtrUU-L0mLN8HoQaPlaUqUHoYDvRgWpThsi0g/edit?usp=sharing>

Meanwhile, the Incident Report can be found here:

<https://docs.google.com/spreadsheets/d/1hmCHGtgnOs9obvSFXRKaZJrBMtVunqk1eXGZcMckUNM/edit?usp=sharing>

For future use, we will alter the instructions so that people who may not be as experienced using Linux-oriented commands. We will also highlight the commands needed for the command line so there is no longer any confusion.

In addition to these aforementioned tests, we performed testing on the inputs and outputs in Figure 4.2.1 that were intended to be available for conversion. For example, files were not capable of being converted from .pdf. The full matrix of possible conversion can be found here:

<https://docs.google.com/spreadsheets/d/1vYWvgQ7T0PwcnToshBOIv8cSMOaXGcJfMpO0ptaiZrY/edit?usp=sharing>

**7** **IMPLEMENTATION PHASE**

**7.1** **Implementation or Transition Plan**

Due to our time constraints and limited capabilities following the transition to remote collaboration, our once stretch goal of fully implementing the microservice into HAX CMS will be handed off to Mr. Ollendyke, Mr. Potter and the Elmsln team. The completed service will be available via GitHub and information can be found in our included Training Guide in Section 7.2. Additional information regarding this transition can be found in the Support Policy which is also in Section 7.2.

In theory, the implemented microservice would be run as part of the Content Management System (CMS). This means that whatever piece of hardware is actually running HAX CMS would also be responsible for running the microservice when the Function as a Service is called. This hardware would either be servers like the ones Mr. Ollendyke is currently using to host HAX CMS, a HAX user's server, or a user’s local computer if they were to load the CMS on there.

In terms of access, the functional interface would be found via the site page editor that currently exists on HAX CMS. The user, a website developer, would be able to find the tool while viewing the page editor, and would be prompted to upload their input file for conversion. The user can then choose whether they want to return the file as a download or upload the file to their web page immediately as HTML.

Elmsln will be responsible for ensuring the two previously mentioned items. First, this involves installing the microservice on HAX servers. After that, Elmsln would need to place a button on HAX’s page editor to allow access to the microservice. This portion can be done through some simple modification of HTML. Finally, they would need to create an adapter capable of reading the input file, determining the desired output type from the user, and inserting this information into the conversion command. As mentioned in Section 3.1, there are very few users of HAX CMS let alone know what it is. With this being the case, the implementation of our microservice should occur through a plunge conversion with little to no issues. The microservice can be delivered as a feature in a routine update to HAX.

Once this implementation is complete, a file can be converted using the microservice directly in HAX. The conversion would occur, and the file will either be sent back to the user for download or pasted into the web page they are currently creating. An example of the first case is if the user wishes to upload their .doc file to the website and have it converted to .pdf similar to many other available tools. Where the system would be unique is allowing the user to upload their files and immediately make their content viewable as part of the web page rather than using a viewer of some sort. From a user standpoint, our microservice, once embedded, automates the process of creating a web page from external documents while making it as simple as dragging and dropping the file.

**7.2** **Training Guides**

To allow for an easy transition when acquiring the system, the following documents are to be included in the handoff handoff project. The Training Guide explains step-by-step how to convert a file using the microservice. It has been formatted in a similar manner to the Docker labs that Mr. Potter created for Mr. Ollendyke’s IST 402 classes. The document can be found here:

<https://docs.google.com/document/d/1aOVpa0hsvkaIhAWQj5FekfvBu9KFn80k11PlC8F_sL0/edit?usp=sharing>

The Support Policy gives information directly relevant to the transition process itself. It includes information about the transition between teams that are responsible for the product, accessibility, and recommendations for the future. The Support Policy can be found here:

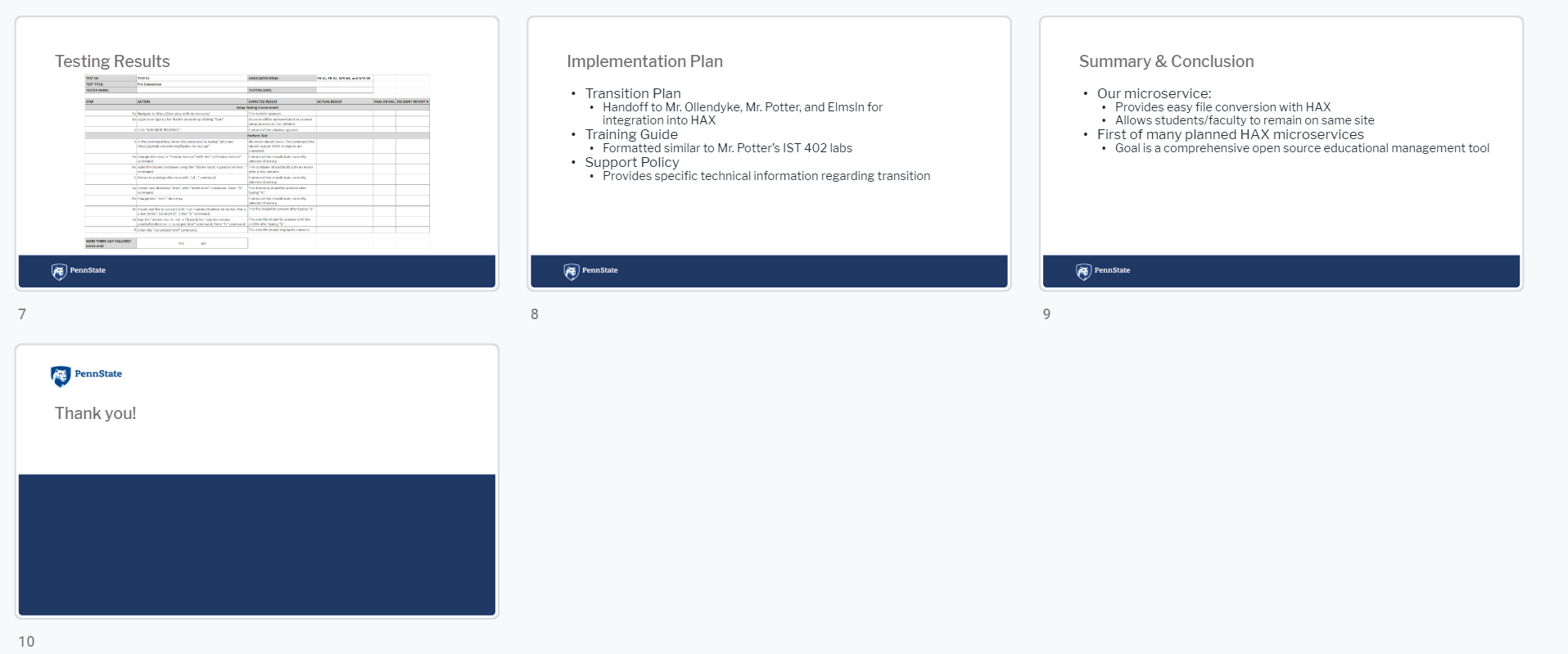
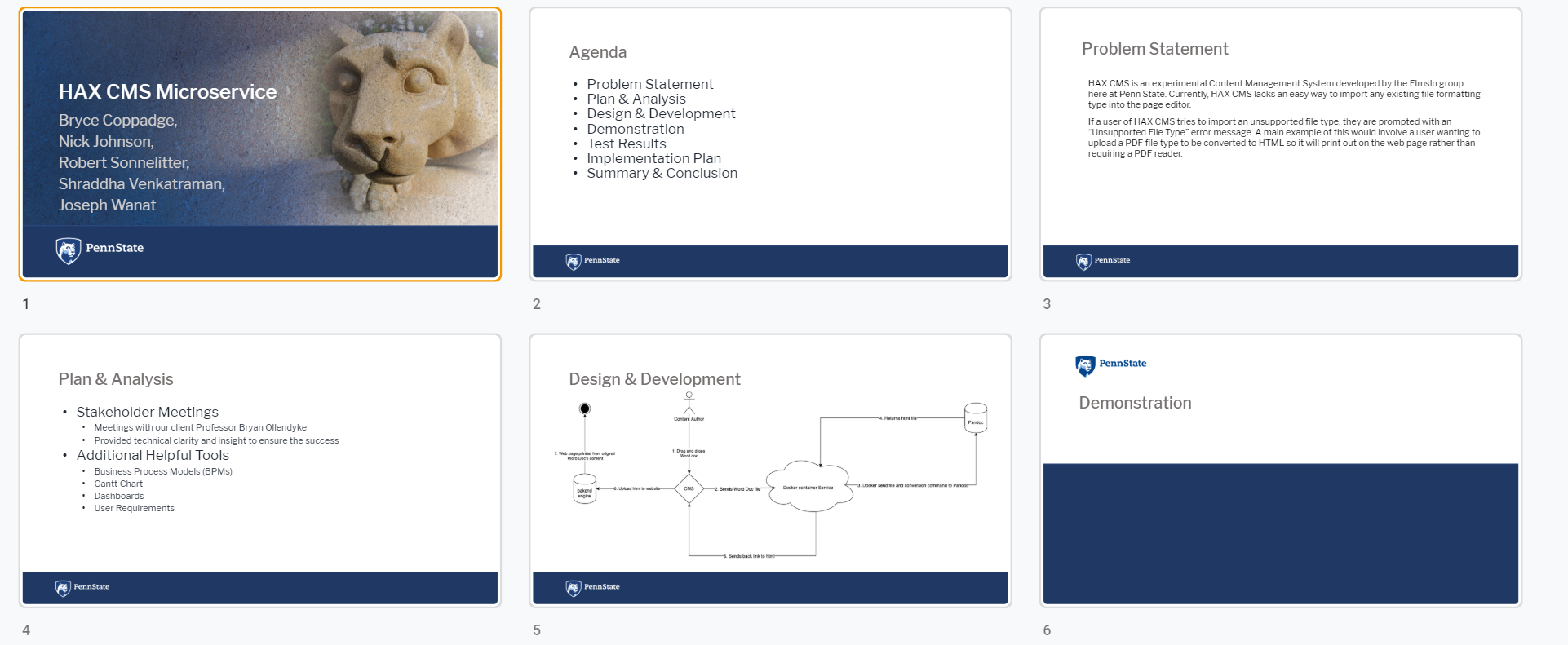
<https://docs.google.com/document/d/1MmlbFEbOLi2DL183FM3NumDP2vOTrB0hQwcJmek2Cy8/edit?usp=sharing>

**8** **REFERENCES**

About pandoc. (n.d.). Retrieved February 13, 2020, from <https://pandoc.org/>

**9** **APPENDIX**

**9.1** **Appendix A**



***Figure 9.1*** *Project Presentation*