|  |
| --- |
| CptS 484: Software Requirements |
| WRS Evolution |
| Requirements Elicitation |

|  |
| --- |
| 10/15/2022 |

|  |
| --- |
| Arlo Jones, Kristofer Koehn, Vlad Onyshchuk, and Jadin Sadler |

Contents

[Revision History 2](#_Toc117027972)

[[1] Introduction 2](#_Toc117027973)

[1.1. Purpose 2](#_Toc117027974)

[1.2. Scope 2](#_Toc117027975)

[1.3. Objectives and Success Criteria 3](#_Toc117027976)

[1.4. Definitions, Acronyms, and Abbreviations 3](#_Toc117027977)

[1.5. Overview 3](#_Toc117027978)

[[2] Preliminary Definition 3](#_Toc117027979)

[2.1. Preliminary Subsystem 1 3](#_Toc117027980)

[2.2. Preliminary Subsystem 2 4](#_Toc117027981)

[2.3. Preliminary Subsystem 3 4](#_Toc117027982)

[2.4. Preliminary Functional Requirements 4](#_Toc117027983)

[2.4. Preliminary Non-Functional Requirements 4](#_Toc117027984)

[[3]Issues with the Preliminary Definition Given 5](#_Toc117027985)

[3.1 Subsystem 1 Issues 5](#_Toc117027986)

[3.2. Subsystem 2 Issues 6](#_Toc117027987)

[3.3. Subsystem 3 Issues 7](#_Toc117027988)

[3.4. Functional Requirements Issues 7](#_Toc117027989)

[3.5. Non-Functional Requirements Issues 9](#_Toc117027990)

[[4] WRS 10](#_Toc117027991)

[4.1.W 10](#_Toc117027992)

[4.1.1. Problem 10](#_Toc117027993)

[4.1.2. Goals 11](#_Toc117027994)

[4.1.3. Improved Understanding Subsystems 11](#_Toc117027995)

[4.1.3.1. Improved Subsystems 1 11](#_Toc117027996)

[4.1.3.2. Improved Subsystems 2 11](#_Toc117027997)

[4.1.3.3. Improved Subsystems 3 12](#_Toc117027998)

[4.1.3.4 Improved Functional Objectives 12](#_Toc117027999)

[4.1.3.5 Improved Non-Functional Objectives 13](#_Toc117028000)

[4.1.4 Stakeholders 13](#_Toc117028001)

[4.2. RS 14](#_Toc117028002)

[4.2.1. Functional Requirements 14](#_Toc117028003)

[4.2.2. Non-Functional Requirements 14](#_Toc117028004)

[4.2.3. Specifications 15](#_Toc117028005)

[[5] Prototype Interface Mock-ups 17](#_Toc117028006)

[[6] User Manual 20](#_Toc117028007)

[[7] References 21](#_Toc117028008)

[Appendix I: Process Details 21](#_Toc117028009)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changes | Editor |
| 10/15/2022 | 1.0 | Creating the WRS Document | Vlad Onyshchuk |
| 10/16/2022 | 1.1 | Small edits and grammar changes | Arlo Jones, Kris Koehn |
| 10/17/2022 | 1.2 | Preliminary def. & Issues with Pre. Def. | Vlad, Kris |
| 10/18/2022 | 1.3 | Finishing Preliminary def. & Issues with Pre. Def. | Kris |
| 10/18/2022 | 1.4 | Finished Prototype Interface Mock-ups & User Manual | Vlad |
| 10/18/2022 | 1.5 | Added Improved Subsystems | Jadin |
| 10/18/2022 | 1.6 | Adding Finishing Touches | Arlo, Kris, Jadin, Vlad |

# [1] Introduction

## 1.1. Purpose

The purpose of this project is to create a blog posting system to visualize the impact of increasing the minimum wage in Washington had on the restaurant industry. To clarify, our team was tasked with creating a tool that would show charts and graphs of different data from the Washington Hospitality Association that could then be used in creating blog posts and outsourced to different webpages.

## 1.2. Scope

The project has been broken down into three main deliverables which are creating a data uploading system, creating a data visualization tool, and the blog posting system. This is an ongoing project and the previous teams that have worked on it have made the data uploading portal in AWS and a form of data visualization. The purpose of our group is to create the blog posting system, which means we would take the data from the server and run it through the data visualization the have it malleable to how the user of the blog posting system wants.

## 1.3. Objectives and Success Criteria

The objectives and success criteria of this project are:

1. Create a data ingestion system
2. Make the data work with Power Bi and tableau
3. Create a date visualization tool
4. Make a webpage that has access to the data visualizer
5. Create a blog posting system
6. Create an Api so others can integrate the tool into their website

## 1.4. Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Power Bi | Power BI is an interactive data visualization software product made by Microsoft |
| Tableau | Tableau is an interactive data visualization software |
| API | Short for Application Programming Interface |
| AWS | Short for Amazon Web Services |
|  |  |

## 1.5. Overview

In section two we are discussing the preliminary definition of the project. This is broken down into three smaller parts which are the preliminary domain, preliminary functional requirements, and the preliminary non-functional requirements. In section three we look at the issues that were given in the preliminary definition. Section four we talk about the problems with the project, the goals to fix these problems, the improved understanding of domain, stakeholders, functional, and non-functional objectives, and the specifications. In section five we discuss the preliminary prototype of this project. Section six is the protype interface mock-up. Section seven, eight, nine are the user manual, traceability, references.

# [2] Preliminary Definition

## 2.1. Preliminary Subsystem 1

|  |  |
| --- | --- |
| **PD\_ID** | **Preliminary S1 Description** |
| PS1\_1 | user upload from various sources. |
| PS1\_2 | Data anonymization and obfuscation |

## 2.2. Preliminary Subsystem 2

|  |  |
| --- | --- |
| **PS2\_ ID** | **Preliminary S2 Description** |
| PS2\_1 | A web-based tool incorporating Microsoft Power BI on a website. |
| PS2\_2 | allow users to generate graphs/charts of the data. |

## 2.3. Preliminary Subsystem 3

|  |  |
| --- | --- |
| **PS3\_ ID** | **Preliminary S3 Description** |
| PS3\_1 | Blog posting system |

## 2.4. Preliminary Functional Requirements

|  |  |
| --- | --- |
| **PFR\_ ID** | **Preliminary Functional Requirements Description** |
| PFR\_1 | Needing to have the data sanitized and aggregated for legal reasons. |
| PFR\_2 | Making the process as automated as possible to make there be no need for too much overhead. |
| PFR\_3 | Make the whole system need very little maintenance so that you do not need a very large team to keep it running. |

## 2.4. Preliminary Non-Functional Requirements

|  |  |
| --- | --- |
| **PNFR\_ ID** | **Preliminary Non-Functional Requirements Description** |
| PNFR\_1 | Database can handle repeated queries without giving out. |
| PNFR\_2 | Graphs and tables are suitably useful while also looking great. |
| PNFR\_3 | The website does not go down when new data is being ingested into the database. |

# [3] Issues with the Preliminary Definition Given

## 3.1 Subsystem 1 Issues

|  |  |  |
| --- | --- | --- |
| **Subsystem 1 Issue ID** | **Subsystem 1 Issue Description** | |
| S1I\_1 | PD\_ID | PS1\_1. Data ingestion and user upload from various sources. |
| 1. Ambiguous. The “users” here were not properly specified and the scope was ambiguous. It has been determined that the “users” that are uploading are the DOR and other government Unsound: where exactly is the data coming from? We will need to structure our solution to make this easy. | |
| Option 1 | Manual data upload. Require someone manually upload the CSV/whatever files to the data processing AWS service. |
| Option 2 | We use the system that is already in place. |
| Choice | Option 3 |
| Rationale | No changes necessary, changing this is out of the scope of the project. |

|  |  |  |
| --- | --- | --- |
| **Subsystem 1 Issue ID** | **Subsystem 1 Issue Description** | |
| S1I\_2 | PD\_ID | PS1\_1. Data anonymization and obfuscation |
| 1. Ambiguous. It does not tell us what parts of the data will be anonymized. It also does not tell us what parts of the data will be obfuscated. | |
| Option 1 | We talk with the project stakeholders and nail down the parts of the data that will need anonymization and obfuscation. |
| Option 2 | We use our best judgment on what parts of the data need to be anonymized and obfuscated and do it ourselves. |
| Option 3 | We use the system that is already in place. |
| Choice | Option 3 |
| Rationale | Option 3 is the best candidate because the system was already implemented by a previous capstone team for this project. |

## 3.2. Subsystem 2 Issues

|  |  |  |
| --- | --- | --- |
| **Subsystem 2 Issue ID** | **Subsystem 2 Issue Description** | |
| S2I\_1 | PFR\_ID | PS2\_1. A web-based tool incorporating Microsoft Power BI on a website. |
| 1. What is a web-based tool? | |
| Option 1 | Use the models that were already made for this project. |
| Option 2 | A desktop top app that uses chromium. |
| Option 3 | An API so that anyone can access it. |
| Choice | Option 1 |
| Rationale | Making changes to the models is outside the scope of the project |
| Satisfied by |  | |

|  |  |  |
| --- | --- | --- |
| **Subsystem 2 Issue ID** | **Subsystem 2 Issue Description** | |
| S2I\_2 | PFR\_ID | PS2\_2. allow users to generate graphs/charts of the data. |
| 1. after talks with the client, the users will be WSU (Washington State University) researchers and the public will not necessarily have direct access to the tools we are making. | |
| Option 1 | Hook up the database that already houses the info to a power BI web interface. |
| Option 2 | Create a simplified web interface that hooks into Power BI and allows less complicated graphs and charts to be made easily |
| Choice | Option 1 |
| Rationale | Making changes to the models is again outside the scope of the project |
| Satisfied by |  | |

## 3.3. Subsystem 3 Issues

|  |  |  |
| --- | --- | --- |
| **Subsystem 3 Issue ID** | **Subsystem 3 Issue Description** | |
| PS3\_1 | **PS3\_ ID** | PS3\_1. Blog posting system |
| 1. Where will the blog posting take place? Where will the user post? How will the user get the privileges to post? How will data interact between the blog posting system and Power BI? | |
| Option1 | We identify the problem with meetings. |
| Option2 | We don’t make a blog posting system. |
| Choice | Option 2 |
| Rationale | After talks with our main stakeholder we have concluded that the project does not need any form of blog posting system linked with the Power BI dashboard and so we do not have to implement one. |
| Satisfied by |  | |

## 3.4. Functional Requirements Issues

|  |  |  |
| --- | --- | --- |
| **Functional Requirements Issues ID** | **Functional Requirements Issues Description** | |
| FRI\_1 | **PFR\_ID** | PFR\_1. |
| Needing to have the data sanitized and aggregated for legal reasons | |
| Option1 | Use the triggers in the database to sanitize the data upon ingestion. |
| Option2 | Use the system that is already in place in the AWS server. |
| Choice | Option 2 |
| Rational | We already have a database in the AWS server that handles all the data sanitization. |
| Satisfied by |  | |

|  |  |  |
| --- | --- | --- |
| **Functional Requirements Issues ID** | **Functional Requirements Issues Description** | |
| FRI\_2 | **PFR\_ID** | PFR\_2. |
| Making the process as automated as possible to make there be no need for too much overhead. | |
| Option1 | Create sufficient documentation and processes to automate most tasks. |
| Option2 | Create an AI that can do automation tasks for us. |
| Choice | Option 1 |
| Rational | It would be important for us to make the documentation and automate everything so that the system functions in a way so that the only manual step that you must do is drop data in. |
| Satisfied by |  | |

|  |  |  |
| --- | --- | --- |
| **Functional Requirements Issues ID** | **Functional Requirements Issues Description** | |
| FRI\_3 | **PFR\_ID** | PFR\_3. |
| Make the whole system need very little maintenance so that you do not need a very large team to keep it running. | |
| Option1 | Write detailed documentation for future engineers. |
| Option2 | Write nothing so that the internal components are considered a black box to everyone. |
| Choice | Option 1 |
| Rational | Having someone that knows how the internals of the system works would make it better for future engineers to fix or improve the system if need be. |
| Satisfied by |  | |

## 3.5. Non-Functional Requirements Issues

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirements Issues ID** | **Non-Functional Requirements Issues Description** | |
| NFRI\_1 | **PNFR\_ID** | PNFR\_1. |
| Database can handle repeated queries without giving out. | |
| Option1 | Build the database in a way where we restrict the number of queries that can be sent at a given interval of time. |
| Option2 | We write our quires intelligently so that we don’t have to make many of them to get the data we need. |
| Choice | Option 2 |
| Rationale | If we write the queries properly we should not have any problem with the amount of traffic the site will generate. |
| Satisfied by |  | |

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirements Issues ID** | **Non-Functional Requirements Issues Description** | |
| NFRI\_2 | **PNFR\_ID** | PNFR\_2. |
| Graphs and tables are suitably useful while also looking great. | |
| Option1 | Make our own visualization tools. |
| Option2 | Use the built in tools of Power BI. |
| Choice | Option 2 |
| Rationale | We don’t think that we could out do Microsoft's team of designers at their own game. |
| Satisfied by |  | |

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirements Issues ID** | **Non-Functional Requirements Issues Description** | |
| NFRI\_3 | **PNFR\_ID** | PNFR\_3. |
| The website does not go down when new data is being ingested into the database. | |
| Option1 | Disconnect the database ingestion from the data that we pull. |
| Option2 | Use the solution that already exists |
| Choice | Option 2 |
| Rationale | The solution that already exists is that the data that is being ingested is in basically a entirely different database while the one that will be hooked up to BI is separate. |
| Satisfied by |  | |

# [4] WRS

## 4.1.W

### 4.1.1. Problem

|  |  |  |
| --- | --- | --- |
| **Problem ID** | **Problem Description** | **Corresponding Goals** |
| P1 | No access to the data ingestion system | G1 |
| P2 | No access to the sanitized database | G2 |
| P3 | No information on the structure of the sanitized data | G3 |
| P4 | No information on where the Power BI models are hosted | G4 |
| P5 | No clear information on the nature of the blog posting system | G5 |
| P6 | No access to the AWS systems | G6 |

### 4.1.2. Goals

|  |  |  |  |
| --- | --- | --- | --- |
| **Goal ID** | **Goal Description** | **Backward Traceability** | **Forward Traceability** |
| G1 | Gain access to the data ingestion system | P1 | FO1 |
| G2 | Gain access to the database | P2 | FO1 |
| G3 | Get information about the structure of the data | P3 | FO1 |
| G4 | Gain access to the Power BI models | P4 | FO1 |
| G5 | Get clear requirements for the blog posting system | P5 | FO1 |
| G6 | Gain access to the AWS system | P6 | FO1 |

### 4.1.3. Improved Understanding Subsystems

### 4.1.3.1. Improved Subsystems 1

|  |  |
| --- | --- |
| **Improved Subsystem ID** | **Improved Subsystems Description** |
| IS1\_2 | The data anonymization and obfuscation systems data comes from multiple government sources that we don’t have to worry about. |
| IS1\_2 | Anonymization and obfuscation of the data will happen to the data in the database upon ingestion. |

### 4.1.3.2. Improved Subsystems 2

|  |  |
| --- | --- |
| **Improved Subsystem ID** | **Improved Subsystems Description** |
| IS2\_1 | A webpage that will lead to a Microsoft Power BI dashboard that displays the data in from the database after it has been anonymized and obfuscated. |
| IS2\_2 | This will allow users to generate their own graphs and charts of the data using the tools of Power BI that are integrated into the dashboard. |

### 4.1.3.3. Improved Subsystems 3

|  |  |
| --- | --- |
| **Improved Subsystem ID** | **Improved Subsystems Description** |
| IS3 | After talking with the client, there will be no integrated blog posting using this data. |

### 4.1.3.4 Improved Functional Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| Improved FR  Objective ID | Objective Description | Alleviates Problem | Achieves  Goal |
| IFRP\_1 | The data needs to have some of its columns removed / hashed to keep the identities of the workers and businesses from the public for privacy / legal reasons | P2, P1, P3 | G3, G2 |
| IFRP\_2 | Reduce the interaction with the system down to just having to give it data and use the power BI dashboard to ask for what data you want from it. | P4 | G4, G1 |
| IFRP\_3 | Create the proper documentation so that maintaining the system is straightforward and is as easy to follow as a step by step guide. | P5 | G5 |

### 4.1.3.5 Improved Non-Functional Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| Improved NFR  Objective ID | Objective Description | Alleviates Problem | Achieves  Goal |
| INFRP\_1 | The database will have intelligent queries written so that there is not a lot of them being passed to the database and the database will have a maximum request limit so that things never bog down. | P3 | G3 |
| INFRP\_2 | The most relevant statewide statistics will be displayed first and by default, as percentages changed over time. With user options to filter the data down per their inquiries | P6, P4 | G4, G5 |
| INFRP\_3 | The website will have power BI linked to the second database that houses the obfuscated data and while ingesting data is happening it does not affect the second database until complete. | P3 | G6 |

### 4.1.4 Stakeholders

|  |  |  |
| --- | --- | --- |
| **Stakeholders Name** | **Role/Association** | **Needs/Wants** |
| Anthony Anton | President and CEO of the Washington Hospitality Association | A tool to evaluate the state and trends of the hospitality industry |
| William Bonner | WSU IS Director of Network & Cloud Engineering Services | Stable, easy to maintain system |
| Greg Neunherz | Director Carson College of Business Office of Technology | Ease of use |
| Nathan Roberts | Carson College of Business Data Analyst | Project is useful for research |
| Sarah Druffel | Carson College of Business Public Relations and Communications Manager | Project has successful public release |
| Brad Gaolach | Director Metropolitan Center for Applied Research and Extension. | Complete the project |
| Bolong Zeng | Program Coordinator and Scholarly Assistant Professor Software Engineering | Creation of a robust, generic data visualization pipeline |
| Corrie Wilder | Exec Director for Marketing and Communications WSU Everett | Project has a successful public release |
| Jacob Murray | Program Coordinator and Scholarly Assoc. Professor Electrical Engineering | That the project satisfies the client, and students met all capstone requirements. |
| Mark Beattie | Assoc. Vice Chancellor for Academic Affairs. | Creative, highly useable solution. |

## 4.2. RS

### 4.2.1. Functional Requirements

|  |  |
| --- | --- |
| **FR ID** | **Description** |
| FR\_1 | Having an AWS server that hosts two databases and the triggers for data ingestion |
| Satisfies Functional Requirement Issue | FRI\_1 |
| Satisfies Objectives | IFRP\_1 |

|  |  |
| --- | --- |
| **FR ID** | **Description** |
| FR\_2 | Having a Power BI dashboard setup that is connected to the data. |
| Satisfies Functional Requirement Issue | FRI\_2 |
| Satisfies Objectives | IFRP\_2 |

|  |  |
| --- | --- |
| **FR ID** | **Description** |
| FR\_3 | Documentation for the system as a whole that will be used for maintaining and developing the platform. |
| Satisfies Functional Requirement Issue | FRI\_3 |
| Satisfies Objectives | IFRP\_3 |

### 4.2.2. Non-Functional Requirements

|  |  |
| --- | --- |
| **NFR ID** | **Nonfunctional Requirement 1** |
| **NFR\_1** | A query limiter on the database so that we don’t overload it. |
| Operationalized Functional Requirements | FR\_1 |
| Satisfies Nonfunctional Requirement Issue | NFRI\_1 |
| Satisfies Non-functional Objective | INFRP\_1 |
| Constrains | P1 |

### 

|  |  |
| --- | --- |
| **NFR ID** | **Nonfunctional Requirement 2** |
| **NFR\_2** | Users need to be able to find the most relevant information without needing to navigate the website or toggle options. Information need to be easily parsable. |
| Operationalized Functional Requirements | FR\_1, FR\_2 |
| Satisfies Nonfunctional Requirement Issue | NFRI\_2 |
| Satisfies Non-functional Objective | INFRP\_2 |
| Constrains | P6 |

|  |  |
| --- | --- |
| **NFR ID** | **Nonfunctional Requirement 3** |
| **NFR\_3** | A proper connection between BI and the database hosted in the AWS server. |
| Operationalized Functional Requirements | FR\_1, FR\_2, FR\_3 |
| Satisfies Nonfunctional Requirement Issue | NFRI\_3 |
| Satisfies Non-functional Objective | INFRP\_3 |
| Constrains |  |

### 4.2.3. Specifications

|  |  |
| --- | --- |
| **Functional Specification ID** | **Functional Requirement** |
| FS4 | The webpage will have a small timer in-between filter changes so users cant generate 100’s of queries by toggling filters on and off quickly. | |
| Satisfies Non-Functional Requirement | NFR\_1 |
| Satisfies Objectives |  |

|  |  |
| --- | --- |
| **Functional Specification ID** | **Functional Requirement** |
| FS5 | The website will host layers of content. Appearing first and foremost will be a graph that represents the health of the industry at a glance. Other metrics will be accessible through further inquiry / exploration of the website. |
| Satisfies Non-Functional Requirement | NFR\_2 |
| Satisfies Objectives |  |

|  |  |
| --- | --- |
| **Functional Specification ID** | **Functional Requirement** |
| FS6 | The website will be taken down for maintenance once a quarter, for a small window of time at night while the aggregate database is updated. |
| Satisfies Non-Functional Requirement | NFR\_3 |
| Satisfies Objectives |  |

# [5] Prototype Interface Mock-ups

Mock-up entry Page:

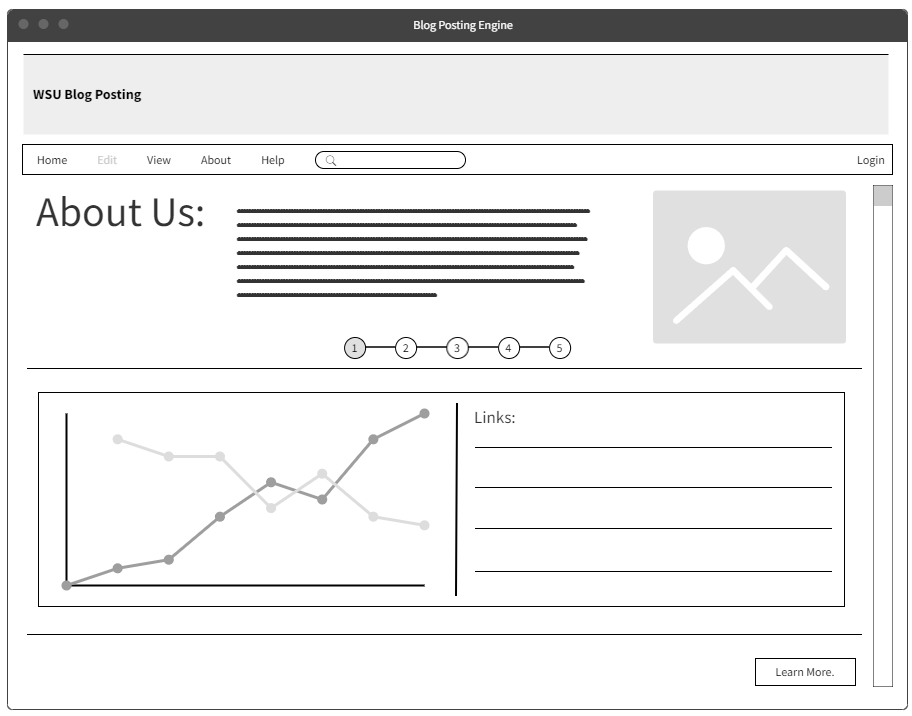


Figure 1: This is the landing page that the user would see if they opened the blog posting engine, it has a small description of what the tool does, how it was made, and some links to blogs already created

Much-up User Manual:

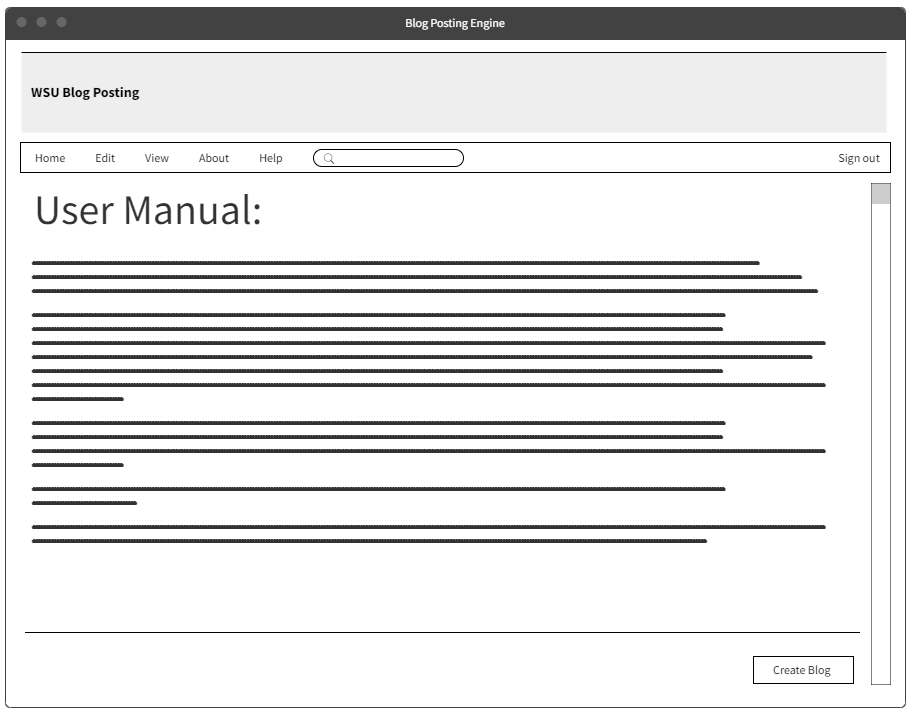


Figure 2: This is the user manual page that would teach a new user how to go about the website and make their own blogs.

Mock-up Chart Selection:



Figure 3: This is the chart/ graph selection page, this page has the ability to show different charts and graphs based on the selected of data the user would like to see, also there’s multiple formats so the user can pick the ones they deem fit and create a blog post with them.

Mock-up Blog Posting:

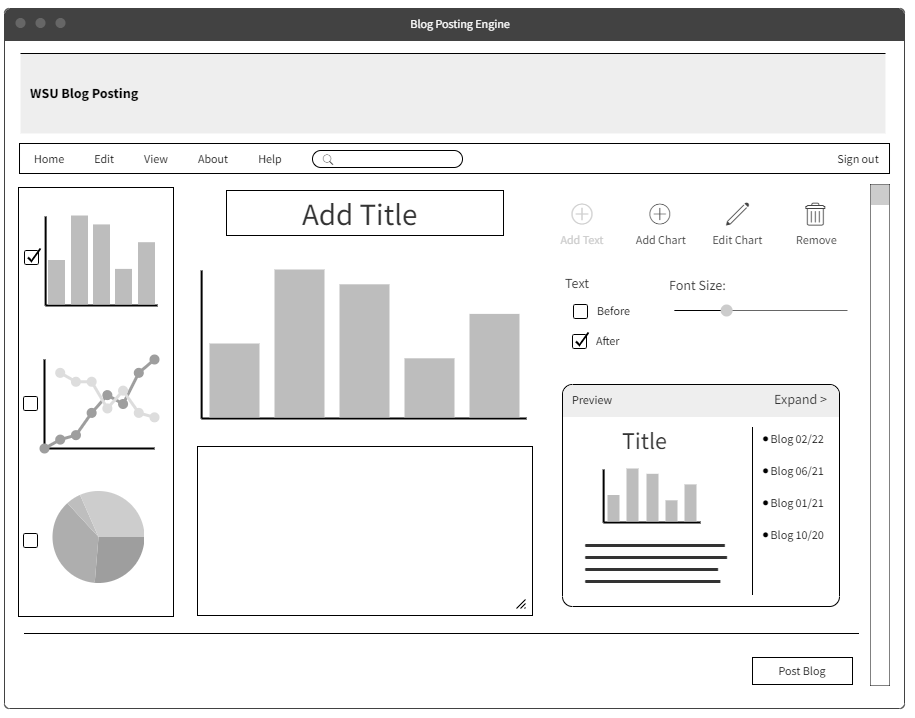


Figure 4: This is the blog posting page where a logged in user can add text, add charts/graphs, edit, and preview their blog before posting it. Once a new blog is created it will appear at the top of the homepage blogs list as the newest blog created.

# [6] User Manual

Blog Posting User Manual V.02:

Homepage:

The landing page has the history of the website alongside a complicated graph, which has links to previously created blog posts. To read a blog click on one of the links, they are sorted by the newest published. If you would like to learn more click the ‘Learn more’ button which will take you to the user manual.

Navigation Tab:

The navigation tab has the option Home, Edit, View, About, Help, Search, and Login. Some features of the engine are disabled, such as edit, until the user logs in. Home takes the user to the home page. Edit lets the user create and edit their previous blog posts. View lets you see all the blogs created by users. About taking the user to a brief history of the website and our mission statement. Help takes the user to the user manual and the contact page. The search is used for finding links to anything on the website. And the login button takes the user to a login page where they can sign in to create blogs.

How to Create a Blog:

In order to create a blog, the user must first have an account with WSU. Make your way to the login button and sign in. Once you are logged in you can now click the edit button to look at charts/graphs. Once you are in the edit page you can select different metrics to modify the graphs and change the data they display. Once you have made some changes and are satisfied with a few charts/graphs check the select box to use them in the blog. Click the ‘Add Blog’ button to go to the blog engine.

Once you are on the blog engine page you can now select the charts and graphs you made previously and add text. Add a title and you will see it update real time in the preview tab. Click the ‘expand’ button to see what the blog post will look like. Now add a blog, if you would like the blog to be above or below the graph select the ‘before’ or ‘after’ button. Add more graphs and text if you’d like. Click the ‘Post Blog’ button once you are finished editing the blog. Now you can preview your blog on the home page or in the view tab. Click it to see links to your new blog and old blogs you made.

# [7] References

[1] Produle, “Wireframe tools, prototyping tools, UI mockups, UX suite, remote designing,” MockFlow. [Online]. Available: https://wireframepro.mockflow.com/. [Accessed: 16-Oct-2022].

## Appendix I: Process Details