Comment Anywhere

CSC 490 - Senior Project I

Fall 2022

Project Requirements

10/17/2022

Instructor Comments and Evaluation

Table of Contents

Instructor Comments and Evaluation 2

Table of Contents 3

Abstract 5

Introduction 5

- Background 5
- Objective 6
  - Phase 1: Initial Product 6
  - Phase 2: User Engagement 6
  - Phase 3: Metrics and Monetization 6
- Team Details & Dynamics 7

Application Domain 8

- Project Context 8

Initial Business Model 8

- Operating Environment 8
- Description of Data Sources 8
- Use Case Diagrams 9

Initial Requirements 11

- Functional Requirements 11
  - Comment Database and API 11
  - User Accounts 12
    - Registering 12
    - Logging In 12
    - Settings 13
  - Moderation 13
    - Validating Comments and Usernames 13
    - Moderators 13
    - Comment Guideline Violation Reporting 14
    - Account Suspensions 14
  - Metrics 14
  - Static Website 14
  - Telemetry 14
- NonFunctional Requirements 15
  - Optimize Resource Usage to Balance User Experience and Costs 15
  - Documentation 15

Username Policy 15
Comment Policy 15
Back Ups 16
Security 16
User Feedback 16
Appealing User Interface 16
Deployability 16
Legality 17

Documentation 17
Proposal Document 17
Requirements Document 17
Specifications Document 17
Design Document 18
Additional documentation: 18
Testing 18
Versioning 18
Documenting 18
Glossary 19
Build 19
Database 19

Testing and Revisions 19

Appendix 21
Technical Glossary 21
Team Details 28
Workflow Authentication 29
Report from Writing Center 30

References 32

Abstract

We describe the results of Software Requirement Engineering in a Waterfall model as applied to our proposed Software Product, Comment Anywhere, a browser extension for commenting on webpages. We begin by addressing the background, objective, and team details. We then explore the application domain, operating requirements, and describe data sources and use cases. Finally, we explicate the functional, nonfunctional, and documentation requirements of Comment Anywhere.

Introduction

Background

Internet denizens have long found ways to have vibrant communications about a wide variety of content. In the past, more websites supported these conversations through comment sections, but many have closed their comments in recent years (Bode, 2021). Instead, the avenues of discourse have become social media sites such as Facebook or Reddit and bulletin board style forums, decoupling the conversation from the content itself.

Tying comments to social media posts rather than the content has the effect of fragmenting the conversation and diluting information available to viewers. Evidence exists that users spend less time on web pages without comments and report a worse user experience (Djinis, 2021). We infer that there is a market for comments coupled to the content and an application that can bring comments to the content could capture monetizable user engagement.

Objective

The objective of this project is to create a web-based browser extension which allows people to comment on websites based on their URL. This is useful for websites that don't have comment sections, especially when users have additional thoughts or information to offer. We will have typical features of commenting systems, including reporting, automated moderation, manual moderation, and account control and additional unique features. Our aim is to create a profitable business, and our business plan informs our requirements. The goal of this project is to have a Product ready to begin Phase 1 of our business plan which is as follows.

Phase 1: Initial Product

- Post initial Product Launch
- Product published to Platform-specific Marketplaces
- Evaluate initial user feedback and begin adding features

Phase 2: User Engagement

- Promotion of a Community focused on feedback, user engagement, and feature requests.
- Focus on Product Growth

Phase 3: Metrics and Monetization

- Expand upon Metrics gathering, ensuring all legal requirements met.
- Inclusion of an Ad Platform as a revenue stream.

Our objective is to create a product that enables users to create discussions alongside content and facilitates the free flow of information.

Team Details & Dynamics

Our development strategy is one of flexible independence that allows all members to modify all code and documentation. The team strives to adhere to proper engineering practices with attention to documentation, review, and testing.

Just like the philosophy of the user experience of the Product, Comment Anywhere, team members have freedom and the responsibility that comes with it. This means maintaining an objective view in pursuit of the product goals, including — but not limited to — providing and receiving feedback from the team.

To ensure all skillsets and perspectives are utilized, communication amongst the team is the top priority. The team will make an effort to assist each other in the learning of new technologies. This communication will take the form of feedback for others' work or ideas and documentation of contributions and workflows.

Every group member has committed to being the Team Leader for a portion of the project.

Team Leader	**Major**	**Phase**
Robert Krency	Computer Science	Requirements
Frank Bedekovich	Computer Science	Analysis
Karl Miller	Computer Science	Design
Luke Bates	Computer Science	Implementation

Application Domain

Project Context

The application domain of Comment Anywhere is internet communication services. More specifically, the domain is internet users commenting and viewing comments about web pages.

Initial Business Model

Operating Environment

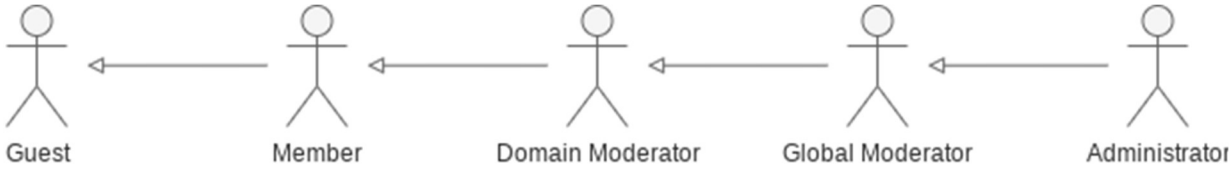
The Front-End user interface and experience will take the form of a browser extension, with initial support targeted at Chromium and Firefox based platforms. The Back End, consisting of the Database and HTTP Server, will be configured to run using containerization to allow for simple deployment on a variety of cloud options.

Description of Data Sources

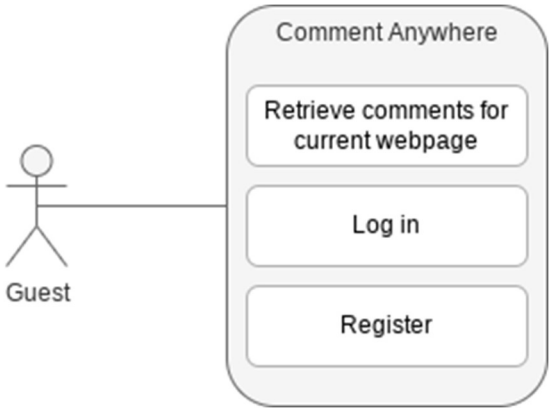
The functionality of Comment Anywhere is enabled through the storing and serving of data primarily originating from users. Users submit data through the extension, such as when they register an account or submit a comment. Comments and Accounts are stored in our database. User login requests and requests to see comments require data sourced from that database. Additional data will be stored, including a record of moderation actions, removed comments, and metrics on usage. To retrieve comments for a specific webpage, the Browser Extension must also provide the Server with a datum of what web page the user is currently viewing and wants comments for. A datum called a Token will also be provided by the server

when a user logs in and at other times, and stored in the Front End, to enable the user to stay logged in by providing that token.

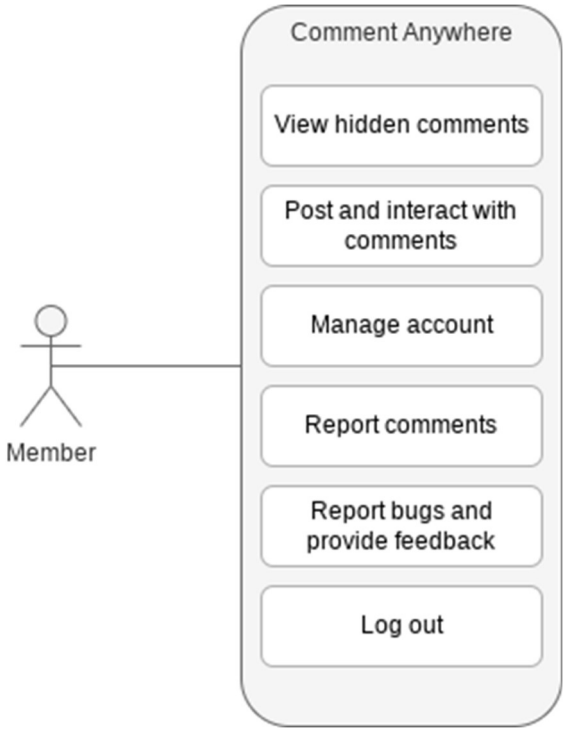
Use Case Diagrams



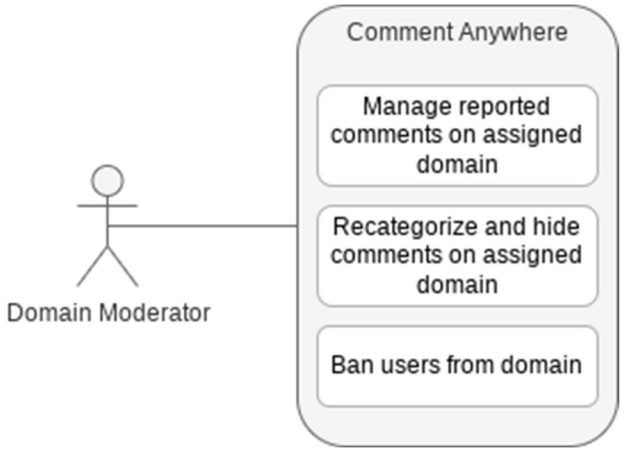
User privileges may be represented by this inheritance diagram. For instance, *members* inherit all *guest* abilities.



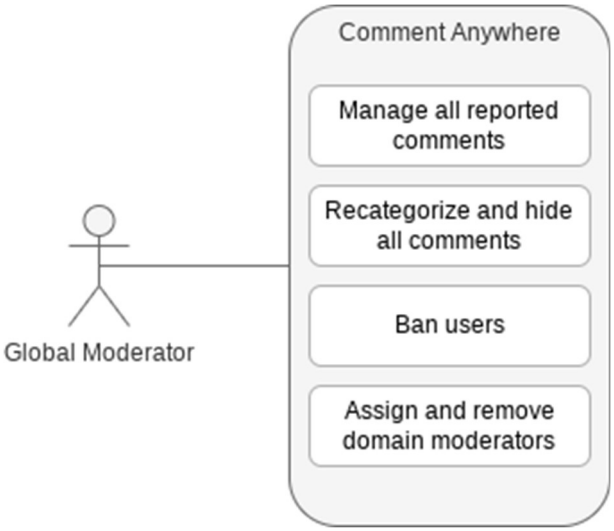
Guests do not need an account to view comments. However, some modes of interaction with our serves are limited without logging into a registered account.



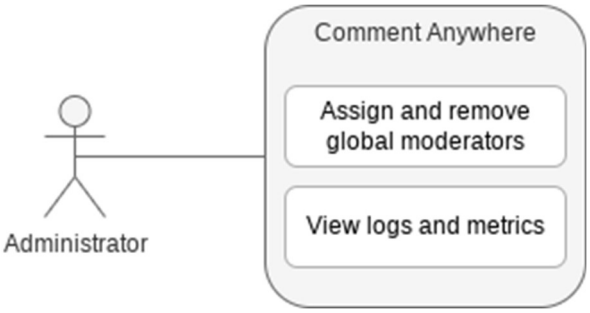
Members may interact with our service by posting and responding to comments. Members may also report comments, view hidden comments, and report bugs.



Domain Moderators manage one or more assigned domains. They may access and remove reported comments, recategorize comments, and ban members. These actions are restricted to the assigned domain.



Global Moderators manage all domains. Additionally, they can remove Domain Moderators.



Administrators manage all domains, remove Global Moderators, and view internal logs, reports, and metrics.

Initial Requirements

Functional Requirements

Comment Anywhere has several functional requirements necessary to meet our objective.

Comment Database and API

A Guest must be able to request comments from the Server by clicking the extension icon in their browser. The Server must be able to serve the User comments related to that URL. The

Browser Extension must be able to display those comments to the user. Members (logged-in users) may also receive comments that are hidden by default to Guests. A Member must be able to post a new comment. The Server must be able to add that Comment to the comment data for the URL the User is commenting on if the user is permitted to comment on that page.

User Accounts

User Accounts are a key component of Comment Anywhere and will support registering, logging in, Moderation activities, and reacting, reporting, and otherwise interacting with comments.

Registering

A Guest must be able to register a new account from the user interface in the drop-down portion of the browser extension. The Server must be able to validate that the Member does not already exist and that their password is of sufficient strength, then either add that Member to the Database or tell the Guest that there was a problem with registration.

Logging In

The Member must be able to log into an account from the user interface in the drop-down portion of the browser extension. The Server must verify whether the Guest has supplied the correct credentials. The Server must be able to track whether an HTTP Request is coming from a logged in Member.

Settings

A Member must be able to change their settings locally to control whether they want to view potentially problematic, hidden, comments. They must be able to reset their password from their settings page.

Moderation

Comment Anywhere must support Moderation actions for privileged users such as viewing flagged comments, suspending users, and taking Moderation Actions on flagged comments. Comment Anywhere will also feature automated comment and username flagging.

Validating Comments and Usernames

The Server must automatically evaluate Comments and Usernames to determine if they may contain words or phrases which violate our policies. They must prevent and remove comments that will violate the policies and flag and hide comments that may do so. It must prohibit the registration of usernames that contain prohibited words.

Moderators

A Domain Moderator or Global Moderator must be able to view comments that have been flagged by Users as rule breaking. They must be able to remove rule breaking comments, take other actions, or clear flags if no action is required.

A Global Moderator must be able to elevate a Member to the status of Domain Moderator. An Admin must be able to assign Global Moderators. Similarly, an Admin must be able to remove Global Moderator permissions and Admins and Global Moderators must be able to remove Domain Moderator permissions. The Server must be able to accurately evaluate permissions and allow or disallow actions based on those permissions.

Comment Guideline Violation Reporting

A Member must be able to report a rule-breaking comment. The Server must be able to track which comments require moderation action. A Member must also be able to report buggy pages.

Account Suspensions

Domain Moderators must be able to suspend privileges for troublesome Members from their domain. The Server must not allow domain-banned Members to post on that domain. Global Moderators must be able to ban Members globally or from a specific domain. Users must be able to appeal Bans and Admins must be able to review banning actions taken by Moderators.

Metrics

Admins must be able to view reports on User Activity, Moderation Actions, and other metrics.

Static Website

There must be a website that provides a description of the project, download links for the Browser Extension, and instructions for installing and using the Browser Extension, which anyone on the internet can view.

Telemetry

When Users submit bug reports, the extensions should collect some basic telemetry about their system and send it to the Server to facilitate diagnosis.

NonFunctional Requirements

Optimize Resource Usage to Balance User Experience and Costs

We anticipate using cloud hosting and having no income throughout Phase 1, so our Product must be designed to minimize cloud costs until we are solvent. However, we must also balance our resource needs with having a performant user experience. We may implement mechanisms such as caching, cache clearing, low cost threads, and offloading certain processing to the Front End to reduce our resource usage while minimizing harm to the user experience.

Documentation

After this initial roll out, we will be in Phase 1 of our Business Plan. During that Phase, we will continue to add features as we build a User Base. Effective code documentation is necessary to enable easily adding new features to Comment Anywhere.

Username Policy

We need to have a username policy that describes guidelines for acceptable Usernames that do not violate general content policies.

Comment Policy

We need a Comment policy which describes which sorts of comments are prohibited or otherwise regulated. Our policies must comply with United States law while promoting freedom of expression.

Back Ups

Our system must be prepared for failures. We will need to create regular backups of our database so we can restore from past points with acceptable quantities of data loss when the system goes down.

Security

We will need to take steps to make the application secure. Generally, most security improvements will be saved until after the initial roll out but some things, such as basic SQL injection prevention and password encryption should be implemented by the initial roll out.

User Feedback

Phase 1 requires that we collect user feedback, bug reports, and feature requests to improve the product throughout that Phase. We must provide ways for users to provide their input to us.

Appealing User Interface

Our user interface will need to be user friendly or else users will simply not use the product. We will attempt to create a pleasing to the eye graphical user interface (GUI).

Deployability

Our build and deployment setup needs to be flexible enough to change hosting providers as our needs evolve. We can set up our build chain to use Containers, such as through Docker, allowing us to deploy our Product on a variety of cloud platforms, or through self-hosting.

Legality

Our Program will have to be compliant with United States law. It must have a Privacy Policy and Terms and Conditions that all Users agree to.

Documentation

Many types of documentation will be created throughout the project's life cycle in order to produce a highly maintainable and extensible product that can adapt and scale after release. The target of the documentation is for current and future developers. Users will require little documentation to interact with the product. Documentation for the developers will consist of documentation generated from code comments and some additional handwritten files.

The standard engineering documents that will be created are:

Proposal Document

The initial document describing the intent of the project.

Requirements Document

This document; used to describe the basic needs of development.

Specifications Document

The document that outlines the actual requirements, standards, testing, and parameters of the final product.

Design Document

The document that describes the specific implementation details.

document that fully fleshes out the project, all minor parts and finishing touches are in here.

Additional documentation:

We will create some extra documentation to enable continued Phase 1 development after the initial rollout. Phase 1 requires that the Comment Anywhere developers continue to add new features. To enable that extensibility, thorough documentation and standard practices are required.

Testing

We will describe a standardized way to write and categorize unit tests and provide instructions for running and writing them.

Versioning

We will describe the use of Git, GitHub and any commit message policies we may implement.

Documenting

We will describe standardized code-commenting practices, the procedure for generating developer project documentation from source code.

Glossary

We will maintain a list of all non-standard words used. It will be divided into general application domain terms and terms specific to this project. It will also have a section describing the contents of each source code folder and a section describing the purpose of each file in root.

Build

We will provide instructions for building and running the project.

Database

We will describe database migration and backup procedures.

Testing and Revisions

This document underwent several phases of testing. The first phase was unit testing during development, when the document was broken into portions for each section. Each unit consisted of a section's content, associated terms, and references. Every team member reviewed each section in isolation many times and improved on the content, references and glossary before the document was combined into an initial rough draft.

The rough draft was continuously integration tested by running a merge script which assembled a markdown file by combining the text, term, and reference components. The document was evaluated as a whole, and improvements were made on component sections to optimize their performance in the rough draft.

After a satisfactory rough draft was produced, the markdown document was copied into Google Docs for quality assurance. Every team member participated in spell checking, formatting, and adding some additional content in the final word document.

Finally, we simulated an acceptance test by taking the document to the Writing Center for review. We made some minor revisions in response to test results.

Appendix

Technical Glossary

Administrator (Admin)

In the context of Comment Anywhere, an Admin is a User with the highest privileges. They can ban users, remove comments, and delegate those authorities by assigning Domain and Global Moderators. They can also access reports.

API

A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

Application Domain

The specific environment in which the product is to operate. (Schach, 2011) Can be an organization, a department within an organization, or a single workspace. (Züllighoven, 2004)

Back End

A Back End is any part of a website or software program the users do not see. It contrasts with the Front End, which refers to a program or website's user interface. (Christensson, 2020)

Browser

A program that accesses and displays information from the Internet.

Browser Extension

A program which has the Operating Environment of a Browser. It can be installed by users of that Browser to add functionality to the Browser.

Bug

A mistake within a computer program that causes unexpected results.

Cloud

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud servers are located in data centers all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines. (Cloudflare, Inc., n.d.)

Comment

A block of text submitted by a User related to some piece of content, which can be viewed by other users.

Comment Section

A group of Comments all related to some specific content. Also, generally, all Comments on a website.

Container

See *Virtual Machine*.

Containerization

See *Virtualization*.

Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are

associated with them, are referred to as a database system, often shortened to just database.Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data. (Oracle Corporation, n.d.)

Docker

An open source platform for building, deploying, and managing containerized applications. (*What Is Docker?*, 2022)

Domain

In the context of Comment Anywhere, a Domain refers to a Domain name, which is a string that identifies a realm of administrative autonomy, authority or control within the Internet. (*Domain Name*, n.d.)

Domain Moderator

A user of Comment Anywhere permitted to take Moderation Actions only for Comments and Users within a specific Domain.

Front End

The Front End of a software program is everything with which the user interacts. From a user standpoint, Front End is synonymous with the User Interface. The Front End of Comment Anywhere is the Browser Extension. (Christensson, 2020)

Git

Git is free and open source software for distributed version control. (*Git*, n.d.)

GitHub

An internet hosting service for software development and version control using Git. (*GitHub*, n.d.)

Global Moderator

An empowered Moderator with the authority to assign and remove Domain Moderators, review Domain Moderator actions, and execute Moderation Actions across all Domains.

Guest

A user of Comment Anywhere who is not logged in. They are capable of viewing some comments for a URL but not posting.

Graphical User Interface (GUI)

A type of user interface which allows a user to interact with an application in ways other than text, for example, with menus and buttons.

HTTP

Hypertext Transfer Protocol; a common protocol for transferring data over the internet.

Inheritance Diagram

A diagram that relates and identifies similar properties among multiple objects.

Integration Testing

The process of testing components together, ensuring that they communicate and work together properly.

Logs

Logs are used by administrators to view technical details of server events.

Member

Any User of Comment Anywhere that is currently logged in.

Moderation Action

Any action taken by a Moderator, Global Moderator, or Admin relating to the banning of users, removal of comments, hiding of comments, or other things necessary for meeting nonfunctional requirements related to what comments and users we allow on our platform.

Moderator

A User who has been granted permission to take Moderation Actions. In the context of Comment Anywhere, Moderators may be Domain Moderators or Global Moderators.

Nonfunctional Requirement

Properties of the product such as platform constraints, response times, or reliability. (Schach, 2011)

Operating Environment

The environment in which a Software Product operates.

Requirements Engineering

The process of defining, documenting, and maintaining requirements in the engineering design process. (*Software Engineering | Requirement Engineering - Javatpoint*, n.d.)

Server

A program which waits for a request then performs some service for the requester and which runs on a computer other than the one on which the requestor/client runs. (Raymond, n.d.) The

Server used by Comment anywhere is an HTTP server because it processes HTTP requests on the Internet.

Static Website

A simple website which serves file data directly and does not perform any advanced functions on the Back End, other than delivering files from the Server to anyone who requests them.

Unit Testing

A testing method whereby the smallest components of a larger item are tested individually.

URL

The address of a resource on the internet.

User

Anyone who uses Comment Anywhere through a Front End. They may be a Guest, Member, Domain Moderator, Global Moderator, or Admin.

User Account

User accounts are used to uniquely identify someone using a particular service.

User Activity

Activity performed by a user of comment anywhere such as viewing comments, posting comments, and taking moderation actions.

User Interface

Also called a "UI" or "interface", a User Interface is the means in which a person controls a software application or hardware device. (Christensson, 2009)

User Privileges

A status that indicates how much a user can interact with a service.

Version control

Systems responsible for managing changes to source code and other collections of information. (*Version Control*, n.d.)

Virtual Machine

Also called a Container, a Virtual Machine is a computer resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual "guest" machines run on a physical "host" machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. (*What Is a Virtual Machine?*, n.d.)

Virtualization

The process of employing a Virtual Machine.

Website

A group of World Wide Web pages usually containing hyperlinks to each other and made available online by an individual, company, educational institution, government, or organization. (*Website Definition & Meaning*, n.d.)

Team Details

This document was created by the team. We met several times in the Student Center to consult with each other as we worked. We utilized Discord to communicate and GitHub to manage project iterations. After the initial draft, all group members participated in simultaneously editing a Google Doc.

Specific contributions are as follows.

Karl Miller created a prototype and test schema to investigate the needs of the project. He created the requirements GitHub repository and wrote a python merge script to merge content and references from subfolders into a single large draft. He wrote the initial Abstract, Objective, Team Details & Dynamics, Operating Environment, Documentation, Testing Revisions, and Functional Requirements sections. He reviewed and edited many other sections.

Luke Bates designed and described the use case diagrams. He helped define telemetry and updated the Comment Policies section. He added necessary terms to the appendix, and improved both grammar and formatting. He participated in group discussions and communicated his ideas and concerns.

Frank Bedekovich Wrote the initial Descriptions of data sources and nonfunctional requirements sections. Also edited and revised the Testing revisions, Glossary definitions, Documentation, Background and objective section. Reformatted the google document after the conversion from the markdown file. Spell checked the entire document for errors.

Robert Krency contributed rewrites, revisions, editing, and formatting of the document as a whole.

Workflow Authentication

I, Karl Miller, attest that I executed the functions listed within the team details section of the document. Also, I agree with all information stated within the requirements document.

Karl Miller	Karl L. Miller	10/12/2022
Printed Name	Signature	Date

I, Frank Bedekovich, attest that I executed the functions listed within the team details section of the document. Also, I agree with all information stated within the requirements document.

Frank J. Bedekovich	Frank J. Bedekovich	10/12/2022
Printed Name	Signature	Date

I, Luke Bates, attest that I executed the functions listed within the team details section of the document. Also, I agree with all information stated within the requirements document.

Luke Bates	Luke N. Bates	10/12/2022
Printed Name	Signature	Date

I, Robert Krency, attest that I executed the functions listed within the team details section of the document. Also, I agree with all information stated within the requirements document.

Robert Krency	Robert Krency	10/13/22
Printed Name	Signature	Date

Report from Writing Center

Client: Comment Anywhere Team

Staff or Resource: Caedon Vogel

Date: 10/10/2022 12:00 pm - 1:10 pm

Did the student request that the instructor receive a visit report? **Yes**

What course was serviced by this visit? **CSC-490**

What goals were established for this tutoring session? **Proofreading and formatting.**

How did the process of this consulting session address the established goals? **We went over the document, then the tutor added his comments to the report page and helped change the minor aspects of the report.**

Please provide any additional comments relevant to this session?

Format Comments

- Add page numbers in accordance to MLA format- top right of page, maybe use 'Comment Anywhere' instead of the "last name". You can also do other things to replicate other documents of this type like 'page # of #' etc.
- Do the paragraphs need to have extra space in between? Most of the document is in MLA, though there isn't a specific format required. I'd probably keep to MLA and get rid of extra spaces in between the paragraphs.
- There are a lot of capitalized nouns that aren't proper nouns. I'm not sure if that is a requirement of a document such as this, but they can probably be uncapitalized because they aren't referring to a specific individual person, place, or thing.
    - Examples under 'Description of Data Sources': "Comments and Accounts are stored in our Database…To retrieve comments for a specific webpage, the

Browser Extension must also provide the Server with a datum of what web page the user is currently viewing and wants comments for. A datum called a Token will also be provided by the server when a user logs in and at other times, and stored in the Front End, to enable the user to stay logged in by providing that token."

- These random capitalizations are strewn throughout the document, and I'm not certain if that is required. I'd go through the paper and uncapitalize them if they aren't required.
    - They may connect to your glossary, and if that's so, it's your call for how to deal with them!

- **Various format things**: a couple headings were mistakes in accordance with the requirement document, talked over some MLA things (the document isn't required to be in MLA but since most of it is, it's better to just stick as close as possible), and other little things.
- Overall, the document looks great! There were zero grammar issues whatsoever, and the document was easily understood, even with my inexperience with Computer Science. The paper was convincing as well, making me look forward to this browser extension's usefulness!

References

Bode, K. (2021, November 8). *Killing Website Comment Sections Wasn't The Brilliant Move Many Newsroom Leaders Assumed*. Techdirt. Retrieved October 7, 2022, from https://www.techdirt.com/2021/11/08/killing-website-comment-sections-wasnt-brilliant-move-many-newsroom-leaders-assumed/

Christensson, P. (2009, March 31). *Home Technical Terms User Interface Definition*. TechTerms. Retrieved October 7, 2022, from https://techterms.com/definition/user_interface

Christensson, P. (2020, April 11). *Backend Definition*. Techterms.com. Retrieved October 07, 2022, from https://techterms.com/definition/backend

Christensson, P. (2020, April 18). *Frontend Definition*. TechTerms.com. Retrieved October 7, 2022, from https://techterms.com/definition/frontend

Cloudflare, Inc. (n.d.). *The Web Performance & Security Company | Cloudflare*. Cloudflare. Retrieved October 7, 2022, from https://www.cloudflare.com/learning/cloud/what-is-the-cloud/

Djinis, E. (2021, November 4). *Don't read the comments? For news sites, it might be worth the effort.* Poynter. Retrieved October 7, 2022, from https://www.poynter.org/ethics-trust/2021/dont-read-the-comments-for-news-sites-it-might-be-worth-the-effort/

Domain name. (n.d.). Wikipedia. Retrieved October 7, 2022, from https://en.wikipedia.org/wiki/Domain_name

Git. (n.d.). Wikipedia. Retrieved October 7, 2022, from https://en.wikipedia.org/wiki/Git

GitHub. (n.d.). Wikipedia. Retrieved October 7, 2022, from https://en.wikipedia.org/wiki/GitHub

Oracle Corporation. (n.d.). *What Is a Database*. Oracle. Retrieved October 7, 2022, from https://www.oracle.com/database/what-is-database/

Raymond, E. S. (n.d.). *The Jargon File - Server*. Catb.org. Retrieved October 7, 2022, from http://www.catb.org/jargon/html/S/server.html

Schach, S. R. (2011). *Object-Oriented and Classical Software Engineering*. McGraw-Hill Education.

Software Engineering | Requirement Engineering - javatpoint. (n.d.). Javatpoint. Retrieved October 7, 2022, from https://www.javatpoint.com/software-engineering-requirement-engineering

Version control. (n.d.). Wikipedia. Retrieved October 7, 2022, from https://en.wikipedia.org/wiki/Version_control

Website Definition & Meaning. (n.d.). Merriam-Webster. Retrieved October 7, 2022, from https://www.merriam-webster.com/dictionary/website

What is a Virtual Machine? (n.d.). VMware. Retrieved October 7, 2022, from https://www.vmware.com/topics/glossary/content/virtual-machine.html

What is Docker? (2022, June 16). IBM. Retrieved October 7, 2022, from https://www.ibm.com/cloud/learn/docker

Züllighoven, H. (2004). *Object-Oriented Construction Handbook: Developing Application-Oriented Software with the Tools & Materials Approach*. Elsevier Science.