Comment Anywhere

Pennwest California

CSC 490: Senior Project 1

Design Document

Dr. Chen

<DATE>

Group Members

| Luke Bates | Computer Science | Implementation |
| --- | --- | --- |
| Frank Bedekovich | Computer Science | Specifications & Analysis |
| Robert Krency | Computer Science | Requirements |
| Karl Miller | Computer Science | Design |

# Instructor Comments and Evaluation

TO DO LIST

* For todo items, see the Trello board: <https://trello.com/b/cULajlBV/design-document>
  + Feel free to add more cards to the board for other work you do
  + You can double click a card to add yourself to the card
  + Move something from “to-do” to “doing” when you are working on it
  + Move something to “review” when you are done working on it

# Table of Contents

[**Instructor Comments and Evaluation**](#_xbkix2sy7ya7) **2**

[**Table of Contents**](#_c4zhs0i6c92k) **3**

[**Abstract**](#_kbk9tzzii0om) **4**

[**Description of Document**](#_kwyvaym0rwfe) **5**

[Purpose and Use](#_powhccbml2c) 5

[Ties to Specifications Document](#_382ibyoavn87) 5

[Intended Audience](#_i6d93d4ih6qx) 5

[**Project Diagrams**](#_clf8f9tz7wt) **6**

[**Design Details**](#_at6bmxgghdcg) **7**

[System Modules and Responsibilities](#_8kjoj65zxz34) 7

[Design Analysis](#_jwdpsmhb1rh) 8

[Design Organization](#_1k0r6hlnk926) 9

[Implementation Timeline](#_cb0fhcylp0q) 10

[Testing](#_uxfgx4p9hjou) 11

[**Appendix**](#_ed6zo1hifewf) **12**

[Appendix A: Team Details](#_x0ywrxecj0sv) 12

[Appendix B: Workflow Authentication](#_bovmkuttulgb) 13

[Appendix C: Writing Center Report](#_3ebbvepb2umf) 14

# Abstract

This document is a description of the design specifications for the Comment Anywhere software product, hereafter referred to generally as Comment Anywhere. The product allows Users to bring discussions to any web content through a Browser Extension. This document is for the developers of the software product. This document is provided to lay out information about the development process and manage guidelines and requirements for development. Architectural details provide a basis for understanding the product modules, the inter-module communication, and the tools of development.

# Description of Document

## Purpose and Use

This document serves to describe the Architecture of the software product, Comment Anywhere.

## Ties to Specifications Document

…

## Intended Audience

…

# Project Block Diagram

## Figure X

# Design Details

## System Modules and Responsibilities

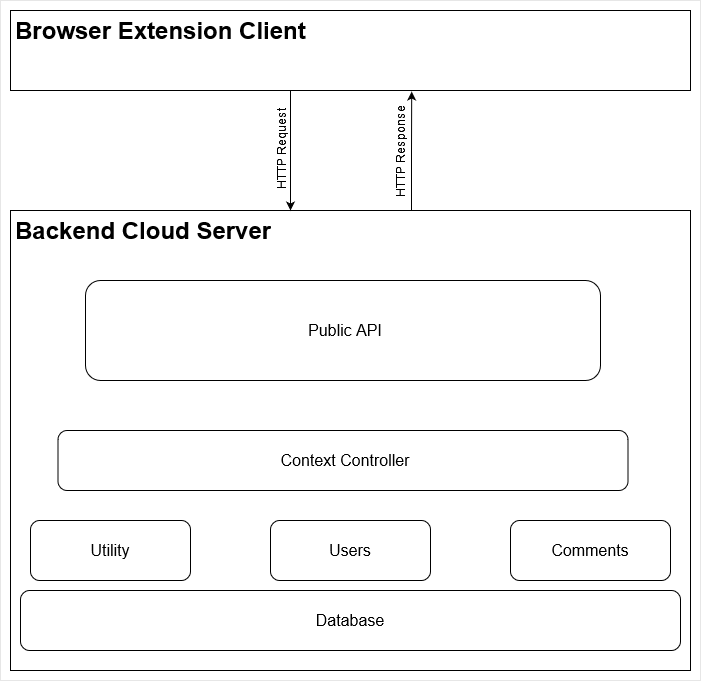


Figure 2: Architectural Diagram

Comment Anywhere is bifurcated into a Client-Server architecture, as depicted in the Architectural Diagram (Figure 2 above). The client is the Browser Extension, a platform specific graphical user interface that runs within a web browser. The client and server communicate through an Application Programmer Interface (API) using HTTP requests and responses to submit and retrieve comments, to register, login, and change user data, and access moderation functionality.

The server consists of several modules: the API, Context Controllers, Users, Comments, Utilities.

The API Module is responsible for accepting HTTP Requests at each exposed endpoint. It ensures valid incoming requests, and delegates to a Context Controller.

The Context Controllers interface with the User Manager to determine User Roles and Permissions and perform User Account related actions. Controllers also interface with the Comment Manager to retrieve, submit, or change content for the specific Context.

The Users module consists of User data retrieval and User account management, and directly interfaces with the Database for storage. Account management includes actions such as creating accounts and updating account information.

The Comments module handles the lifecycle of content, primarily the Comments that make up discussions on Comment Anywhere.

The Utilities module consists of utility functionality such as logging and data conversion or composition.

### Module Cohesion

Modular Cohesion is achieved through restricting module functionality to a category of Data that each Module is responsible for. For example, the Users module is the sole manager of all User related data and management, such as saving and loading User data from the Database. This is mirrored in the design of the Comments module as well.

### Module Coupling

Modules are designed to be as loosely coupled as possible. Context Controllers are the main source of inter-module interaction. This helps enforce separate modules have no direct interaction. Each API endpoint is defined as a specific route, and generates a Context Controller. The Controller then interfaces with the other modules depending on the action generated, such as retrieving User information and permissions from the User module or passing a designated filter to the Comments module to retrieve specific comments.

## 

## Design Analysis

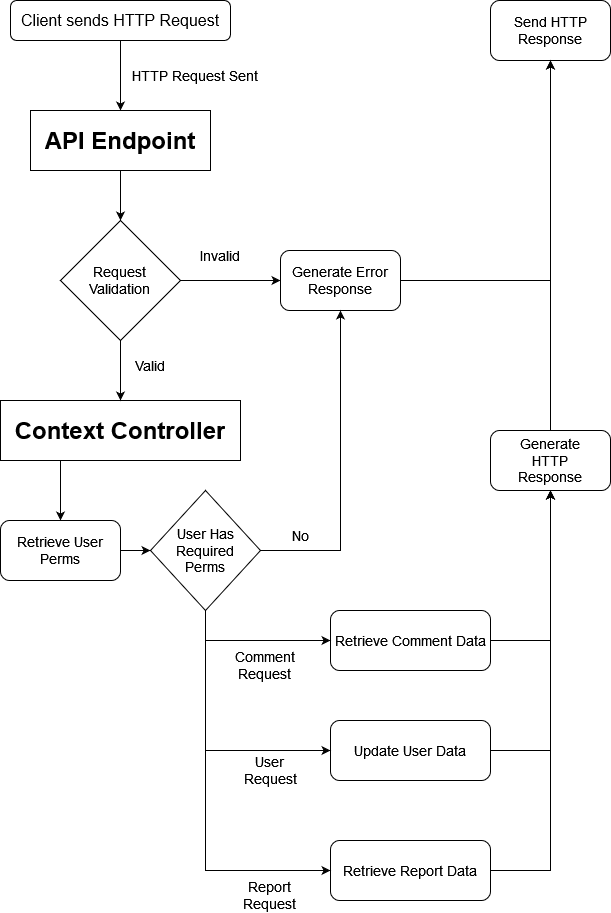


Figure 3: Data Flow Diagram

The Data Flow Diagram (Figure 3 above) shows a high-level view of the lifecycle of an HTTP Request sent from a Client to the Server. An incoming request is validated to be a properly formed request before the appropriate API Endpoint generates a Context Controller for the request. The Context Controller then retrieves User Permissions to verify if the User is able to make the request. Upon success, the Controller performs the request action based on the request type, and sends a successful payload response back to the client in the form of an HTTP Response. Failed request responses will be delivered with appropriate error messaging.

## 

## 

## Design Organization

Detailed Tabular Description of Classes / Objects

Description

Data Members, Types, Constraints

Member Function Listing, Description

Functional Description

Input, Output, Return Params, Types

Modules used

Files Accessed

Real-time Requirements

Messages

Narrative, PDL

1. Start screen
   1. Log in
   2. Create a account
2. Log in
   1. Enter username
   2. Enter password
   3. Reset password
3. Create a account
   1. Enter a email.
   2. Enter a password
   3. Submit. Is there a email and a password? Is the email valid?
      1. Yes
      2. No
      3. Yes
      4. No

### Implementation Tools

* Backend in Go
* Frontend in Typescript
  + Usage of Extension Frameworks

## Implementation Timeline

Blah

## 

## Testing

The team utilized Trello, a task tracking tool. The construction of the document was divided up into smaller, individual tasks. Each member of the team was assigned specific tasks to work on independently, moving each item into the appropriate category on the board. When an item was moved into the “Review” category, each other team member would review the work and submit feedback. After implementing any feedback, final approval is required by the team. This process was used to ensure all tasks were completed in a timely manner and reviewed by the entire team.

# Appendix

## Appendix A: Team Details

## 

## Appendix B: Workflow Authentication

## 

## Appendix C: Writing Center Report