**Your Project Title**

**Design Document**

mm/dd//yyyy

Version <major.minor>



**Your Team’s Name**

(Provide a list of team members)

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

**Note**:

Length = 3-4 pages text+ appendixes as needed. Cover page, table of contents, pictures, images, use-case UML diagrams do not count for the 3 pages text.

Posted as a PDF file.

Typed single‐spaced.

Typed with black text.

Typed with #11 font size.

Typed using Calibri font.

Typed with one inch margins on sides, top and bottom.

**(Please erase this page in your final document.**

**TABLE OF CONTENTS**

[I. Introduction 3](#_Toc495418913)

[II. Architecture Design 3](#_Toc495418914)

[II.1. Overview 3](#_Toc495418915)

[III. Design Details 3](#_Toc495418916)

[III.1. Subsystem Design 3](#_Toc495418917)

[III.1.1. [Subsystem Name] 4](#_Toc495418918)

[III.1.2. [Subsystem Name] 4](#_Toc495418919)

[III.2. Data design 4](#_Toc495418920)

[III.3. User Interface Design 4](#_Toc495418921)

[IV. Testing Plan 4](#_Toc495418922)

[V. References 5](#_Toc495418923)

[VI. Appendix: Grading Rubric (for iteration-1) 5](#_Toc495418924)

# Introduction – Needs work - Jared

Explain the purpose for providing this design document If this is a revision of an earlier document, please make sure to summarize what changes have been made during the revision (keep this discussion brief).

The purpose of providing this design document is to illustrate and outline the main component and sub-system diagrams that will be used as a guide to serve as a visual aid for our web development project. The goal of this project is to build a website for the WSU Cougs In Space Club. We want to build this website so that the site administrator will be able to post updates about events and happenings from within the club.

Then provide a brief description of your project and state your project goal.

At the end of the introduction, provide an overview of the document outline.

*Section II includes …*

*Section III includes …*

**Document Revision History**

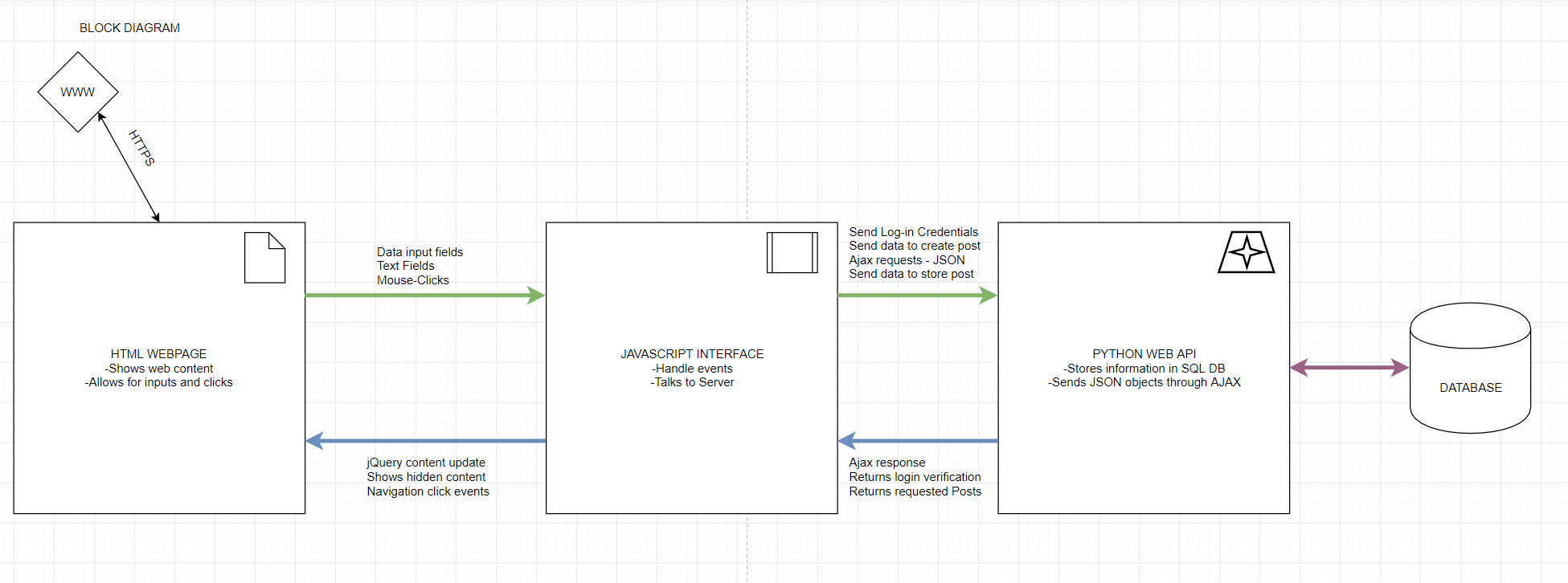
Rev x <date> <comment> (Ex: Rev 1.0 2017-­09-09 ­ Initial version)

Rev 1.0 10-16-2017 ­ Initial version

# Architecture Design – DONE! -Jared was here

## Overview

Our software system will be a website designed to handle basic navigation and content post updates per team page that is implemented. We have decided that the best way to serve up a solid website to our end user is to build three main components, or sub-systems that will work cohesively within the entire website system structure. The first of these sub-systems is going to be our user interface. This user interface is going to written in HTML with the help of the Bootstrap libraries and grid layout CSS styling. Our user interface will be designed in such a way that will allow it to have many interactive buttons and action events that users may trigger with mouse clicks and keyboard input. Our second sub-system will be the JavaScript database interface and event handling software. This JavaScript sub-system will allow the user to modify certain input data fields and handle many mouse button clicks on certain pages. In this sub-system we will also make extensive use of the jQuery libraries to dynamically update our HTML front end content. Our third and final sub-system interface is the python database server interface. This API will be designed to work with the JavaScript and jQuery commands from the second interface to store and withdraw data from an SQL database. For our software system we have decided to implement a Model View Controller architectural pattern. The rationale behind choosing this pattern was the fact that our website will need to run interactivity commands on the client side and be able to make requests to a database structure on the server side. With the Model View Controller architectural pattern, we are able to process and handle many user interactions and interfaces on the client side and minimize the amount of server-side computation that must be handled.



# Design Details

## Subsystem Design - Fill in

This section provides more detail about each subsystem in your architecture. For each subsystem, include a sub-section and explain:

* The important features internal to the subsystem such as descriptions of critical algorithms used, protocols, and key invariants. Also, wherever possible items should be linked back to your requirements specification.

If there are messages sent between clients and servers, **you should identify what messages and what data they contain, and in what format, and in what order they should be sent.**

(Note: *We expect to see a more refined design for the features to be included in the current iteration, and perhaps a more rough design for the features to be implemented in future iterations.* )

* (in iteration -2) Provide your class level design. You should include a UML class diagram visualizing your class level design. In addition, explain each class in detail, specify and explain their methods.

If you have considered alternative designs, please describe briefly your reasons for choosing the final design.

(suggested template)

### [HTML User Interface] – Finish

* The important features internal to the subsystem such as descriptions of critical algorithms used, protocols, and key invariants. Also, wherever possible items should be linked back to your requirements specification.

If there are messages sent between clients and servers, **you should identify what messages and what data they contain, and in what format, and in what order they should be sent.**

(Note: *We expect to see a more refined design for the features to be included in the current iteration, and perhaps a more rough design for the features to be implemented in future iterations.* )

* (in iteration -2) Provide your class level design. You should include a UML class diagram visualizing your class level design. In addition, explain each class in detail, specify and explain their methods.

If you have considered alternative designs, please describe briefly your reasons for choosing the final design.

### [JavaScript Interface] – Finish

* The important features internal to the subsystem such as descriptions of critical algorithms used, protocols, and key invariants. Also, wherever possible items should be linked back to your requirements specification.

If there are messages sent between clients and servers, **you should identify what messages and what data they contain, and in what format, and in what order they should be sent.**

(Note: *We expect to see a more refined design for the features to be included in the current iteration, and perhaps a more rough design for the features to be implemented in future iterations.* )

* (in iteration -2) Provide your class level design. You should include a UML class diagram visualizing your class level design. In addition, explain each class in detail, specify and explain their methods.

If you have considered alternative designs, please describe briefly your reasons for choosing the final design.

### III.1.3.[Python Database Server] – Finish

* - The important features internal to the subsystem such as descriptions of critical algorithms used, protocols, and key invariants. Also, wherever possible items should be linked back to your requirements specification.

If there are messages sent between clients and servers, **you should identify what messages and what data they contain, and in what format, and in what order they should be sent.**

(Note: *We expect to see a more refined design for the features to be included in the current iteration, and perhaps a more rough design for the features to be implemented in future iterations.* )

* (in iteration -2) Provide your class level design. You should include a UML class diagram visualizing your class level design. In addition, explain each class in detail, specify and explain their methods.

If you have considered alternative designs, please describe briefly your reasons for choosing the final design.

## Data design - Finish

Describe all data structures (including the internal and temporary data structures), and the database tables created as part of the application. Provide the schemas (attributes) of the tables.

## User Interface Design - Finish

Provide a detailed description of user interface you have built so far. The information in this section should be accompanied with proper images of your screenshots. Make sure to mention which use-cases in your “Requirements Specification” document will utilize these interfaces for user interaction.

# Testing Plan

(starting iteration 2)

In this section goes a brief description of how you plan to test the system. Thought should be given to how mostly automatic testing can be carried out, so as to maximize the limited number of human hours you will have for testing your system. The effort you put early on automated testing will pay off when you have to ensure that you are not breaking existing functionality in future iterations.

Consider the following kinds of testing:

* **Unit Testing:** Explain for what modules you plan to write unit tests, and what framework you plan to use.
* **Functional Testing:** What routes and APIs does your server support that you plan to test? How will you test them? What tools you will use? Will you write mocks?
* **UI Testing**: How do you plan to test the user interface?

# References

Cite your references here.

For the papers you cite give the authors, the title of the article, the journal name, journal volume number, date of publication and inclusive page numbers. Giving only the URL for the journal is not appropriate.

For the websites, give the title, author (if applicable) and the website URL.

# Appendix: Grading Rubric (for iteration-1)

(Please remove this page in your final submission)

These is the grading rubric that we will use to evaluate your document.

|  |  |
| --- | --- |
| Max Points | **Design** |
| 10 | Are all parts of the document in agreement with the product requirements? |
| 15 | Is the architecture of the system described, with the major components and their interfaces? |
| 5 | Is the rationale for subsystem decomposition and the choice for the architectural pattern explained well? |
|  | Are all the external interfaces to the system (if any) specified in detail? |
| 15 | Are the major internal interfaces (e.g., client-server) specified in detail? |
| 15 | Are the subsystems that the team has started to implement are described in the document? Are the algorithms and protocols for those subsystems explained in sufficient detail? |
| 10 | Is there sufficient detail in the design to start Iteration 2? |
|  | **Clarity** |
| 5 | Is the solution at a fairly consistent and appropriate level of detail? |
| 3 | Is the solution clear enough to be turned over to an independent group for implementation and still be understood? |
| 12 | Is the document making good use of semi-formal notation (UML, diagrams, etc) |
| 5 | Is the document identifying common architectural or design patterns, where appropriate? |
| 5 | Is the document carefully written, without typos and grammatical errors? |

Grading rubric for iteration-2 and 3 will be posted later.