CS 463 Capstone Design Document Revisions 1.0

April 17, 2019

Student Letter of Recommendation Management Site

PREPARED FOR

JUSTIN WOLFORD _____ _____

Signature *Date*

PREPARED BY

GROUP 8

THE LETTERMEN

LORENZO AYALA _____ _____

Signature *Date*

MATTHEW KOTRE _____ _____

Signature *Date*

SCOTT WADDINGTON _____ _____

Signature *Date*

JOHNATHAN LEE _____ _____

Signature *Date*

MINGWEI GAO _____ _____

Signature *Date*

Abstract

Oregon State University has had a growing issue with the process of handling letters of recommendation a long with the necessary information to create them. The issue presented has occurred due to the rapidly growing student body at the university which in relation has resulted in a larger amount of request for letters from the professors of the institute. In order to handle the growing issue a solution was proposed to develop a web application to handle all requests as well as information transfer between students and professors to smooth out a complex process. In this document is the discussion of the major technologies/pieces that will be incorporated to create such an application. Each technology will be analyzed in a sense of the uses and design into the application.

CONTENTS

REVISIONS

Section	Original	New
Security/subsection implementation	The packages that will provide the necessary security are the LDAP/SAML packages for login.	Revisions have been made to the original plan for the login of the apllication to better fit Oregon State University's network. We have therefore changed the login package to the django CAS package to allow a login through OSU accounts.
Glossary	Removed LDAP and SAML terms as well as definitions since they are no longer relevant to the project.	Added term CAS to glossary and its definition so that the reader is aware of what the protocol does.

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to expand and elaborate the implementation of the major technological pieces/aspects that will used to create the clients proposed product. An in-depth look will be provided to the necessary components needed to achieve our goals. The in-depth analysis will cover various viewpoints, context, dependencies, and so on while aiming for our target audience which includes the client as well as our users (students and professors of Oregon State University). Through this document the client will be able to assess and evaluate the stated components within the final product to this document to ensure quality implementation.

1.2 Scope

Oregon State University has seen a growing issue to handling documents between students and professor during the process of creating a letter of recommendation. Acting on this issue was the proposition to develop a web application which will handle requests and file transfer between the users. Utilizing the application will extend the services provided by Oregon State University to its students and professors while providing a smoother experience to a process that has grown complex, stressful. Creating such a product will require a variety of pieces to build the desired application. Therefore, in this document technologies are separated into distinct pieces and are analyzed to convey their role in the product. Combining the analyzed technologies will in plan create the desired product for the client and providing a service that will provide a user-friendly experience for handling letters of recommendation.

1.3 Summary

In this document, the opening section will provide a brief introduction to the entire document. The next few sections will provide references, a glossary, and a time line for necessary documentation as well as requirements. In the following sections the introduction and discussion of each technology piece will be presented providing a more in depth look at how the technologies will be implemented within the application.

REFERENCES

[1] Rascia, T. "What is Bootstrap and How Do I Use It?" Tania Rascia, 09 Nov. 2015. Web. 26 Nov. 2018. Available: https://www.taniarascia.com/what-is-bootstrap-and-how-do-i-use-it/

[2] Usage of web servers for websites. Web. Accessed 27 Nov. 2018. https://w3techs.com/technologies/overview/web_server/all

[3] What is a container? — Docker. Web. Accessed 27 Nov. 2018. https://www.docker.com/resources/what-container

[4] Running PHP on Apache httpd. Web. Accessed 27 Nov. 2018. https://wiki.apache.org/httpd/php

[5] How It Works - Let's Encrypt. Web. Accessed 27 Nov. 2018. https://letsencrypt.org/how-it-works/

[6] OWASP Top 10 Application Security Risks - 2017. Web. Accessed 26 Nov. 2018. https://www.owasp.org/index.php/Top 10-2017 Top 10

GLOSSARY

Term/Acronym	**Definition**
User	Those who interact with our project.
CAS	Acronym for Central Authentication Service which is a single sign-on protocol that allows user to access multiple network applications while only submitting their credentials once.
MySQL	An open-source, relational database.
OSU	Acronym for Oregon State University in terms of this document.
UUID	Acronym for Universally Unique Identifier which is a 128-bit value to identify values within a given network/system.

2 USER INTERFACE

The User Interface (UI) of a web application plays a vital role in the usability of the application. The user cannot interact with the functionality of a site without a clear and intuitive interface. The open-source Bootstrap framework will be used to implement the interface for this application. This is a powerful web development framework which has been used in many high-traffic social media sites like Twitter.

2.1 Context

In regards to UI, the users will be defined as those who use the web application. Specifically, students and professors at Oregon State University. The UI is the one aspect of the system that users will directly interact with, as the functionality of the server and database will be handled by the back-end.

2.2 Composition

Bootstrap encompasses four components of web development.

1) Cascading Style Sheets (CSS), incorporated with HTML to control the visual aspects of the page layout
2) HTML, responsible for the elements which makeup the webpage, such as buttons, menus, etc.
3) Javascript, partially responsible for the functionality of the webpage, in conjunction with the database and security protocols.

2.3 Dependencies

Bootstrap requires jQuery, an ”extremely popular and widely used JavaScript library, that both simplifies and adds cross browser compatibility to JavaScript” [1]. It will also need to interact with the MySQL database used to store the letters of recommendation for exchange, as well as the security features used for login and encryption.

2.4 Interfaces

Due to the fact that users will be uploading and downloading documents to and from the server, there needs to be clear requests and responses from the server, Upon upload to the server database, the server should respond to the client with a message containing a notification that the file was successfully uploaded, and possibly the size of the file in bytes for verification. If, for some reason, the file is not uploaded correctly, there should also be a response from the server indicating this.

2.5 Resources

Bootstrap will require the following external resources.

1) **jQuery** javascript to handle the functionality of the application
2) **MySQL** to create a database system for storing and retrieving documents

jQuery can be downloaded and used with Bootstrap, but the javascript will have to be able to work with the MySQL database, which is not hard to accomplish.

3 Web Server

The web server is one of the most important pieces of any web application. The role of the web server is to serve the web pages and code to the user. This application will be implemented using the Apache HTTP server. Apache is the most used web server on the internet [2]. Due to that fact, Apache is both very widely supported and documented. These factors make Apache an ideal choice for the use in the student letter of recommendation application.

3.1 Requirements

A web server is not useful if it cannot be accessed by the user. Apache needs to run somewhere reliable. In the case of the student letter of recommendation manager, Apache will be running inside of a Docker container to make deployment and maintenance as painless as possible. A container is similar to a virtual machine and allows an isolated virtual environment to be configured which allows the program to run with the same configuration where ever the container is deployed [3].

3.2 Modules

Natively, the Apache HTTP server only serves static content. That is: HTML, CSS, and JavaScript. Modules are required in order to support dynamic back-end content and interact with the MySQL database server. Since the student letter of recommendation manager will primarily use PHP, the PHP module for Apache is required [4]. A separate module to access the MySQL database server is not required because that functionality is included in the PHP module.

3.3 Security

Any application that handles user data should strive for security. That is also the case for the student letter of recommendation manager even though the application will not handle much in terms of sensitive user data. One of the most important means, and also easiest, of securing a web server is through the use of SSL encryption. SSL encryption allows the user to encrypt the data in a way that only the intended destination server can decrypt [5]. SSL also makes Man-In-The-Middle (MITM) attacks much more difficult. SSL encryption requires a certificate from a Certificate Authority. The student letter of recommendation manger will use a free certificate from Let's Encrypt. Apache will be configured to use the certificate to encrypt all traffic between the application and the user including the user's password and documents.

4 DATABASE MANAGEMENT

Management of the database utilized for the web application will be essential to providing an efficient and fast service which will handle requests a long with file transfer. Given the provided scenario access and optimization of the database will be key.

4.1 Context

Since the project is focused on user management and communication, using the simple and effective language of MySQL to build the database is the main part of back-end design. The MySQL language is the best choice because of the clear structure, strong transplantation with good expansibility and function. The developers can process large sets of data more efficiently by using MySQL (Structured Query language) as the main tool building our back-end. Meanwhile, the MySQL language can be supported by the ODBC (Open Database connectivity) applications. We can use the ODBC API to interact with the database server. In the end, the team will need to check the final outcome and modify the configuration parameters to optimize the performance of MySQL database and test the database work through C++ language.

4.2 Testing for Optimization

In the provided subsection is an example of testing the database for access and optimization. For the examples provided syntax is not entirely accurate as LaTex does not allow the use of certain characters.

Database access:
ODBC API
1. Add dependency:
include ⟨sql.h⟩
include ⟨sqlext.h⟩
include ⟨odbcinst.h⟩

pragma comment (lib, odbc32.lib)
pragma comment (lib, odbcccp32.lib)

odbc32.lib corresponding with sqlh andsqlest.h
odbcccp32.lib corresponding with odbcinst.h

2. establishing the database connection

Database environment handle: SQLHENV m hEnviroment;

Database connection handle: SQLHDBC m hDatabaseConnection;

Execute statement handle: SQLHSTMT m hStatement;

Overall, using ODBC API establish the database connection can be divided by three steps:

1. Apply the environment handle.

2. using environment handle apply connection handle.

3. using connection handle connect database.

Code sample:

```
SQLRETYRN l uiReturn = SQLALLocHandle(SQL HANDLE ENV, NULL, m hEnvironment);
```

//apply any handle will use this function, parameter1 is the type of handle will be applied, parameter2 is the corresponding handle (Null because no corresponding handle), the application result will be saved in parameter 3

MYSQL using

```
/*application statement handle*/
SQLRETURN l uiReturn = SQLALLcoHandle(SQL HANDLE STMT, m hDatabaseConnection, m hStatement);
```

```
/*compose the SQL statement*/
CString l cstrSql
L cstrSql.Format(T(SELECT * FROM database ));
```

```
/* SQL statement execution*/
L uiReturn = SQLExecDirect(m hStatement, L cstrSql.GetBuffer(), SQL NTS);
```

```
/*read out the results*/
L uiReturn = SQLFetch(m hStatement);
/*getting data*/
SQLGetData(m hStatement, 1, SQL C ULONG,1 siID,0,l siIDLength);
```

```
/*free statement*/
SQLFreeHandle(SQL HANDLE STMT, m hStatement);
```

The major parameter optimization:

1. Slow queries can check the status of parameter. By open the slow query log in my.cnf, track the visit situation.

2. linking number:

Sometimes if the service access raising, we need consider increase the scatter reading pressure. The ideal setting is MAX used connections/ max connections *100

3. Key buffer size

The key buffer size is the most importent parameter for MyISAM. If the key cache miss rate is under the 0.01, it's mean key buffer size allocate too much memory, we can reduce it appropriately. The ideal setting: Key block used/ (Key block unused + Key blocks used)*100

4. temporary table

Temporary table can be used in advanced query like Group by. Every time create temporary, Create tmp tables increase, if we create the temporary table on disk, create tmp disk tables also increase. The ideal setting: Created tmp disk table / create tmp tables*100

5,.open table situation

The ideal setting: Open tables / Opened tables *100

Open tables/table cache *100

6. Query cache

checking cache utilization: (query cache size Qcache free memory)/ query cache size *100if the cache utilization is under the 25Checking cache hit ratio: (Qcache hits Qcache inserts) / Qcache hits *100In other words, if Flush Query cache takes a long time, the cache is too large.

5 VERSION CONTROL SYSTEMS

The version control system is an essential component to consider for while the project is being built from the bottom up. Without a version control system there would be a mangled mess of varying projects with differing progress. It is a way to keep the work tightly organized within a repository.

5.1 Context

The use of a version control system is a way for the project to be stored. Its purpose is for developers to stay on track of the project and keep their files organized. It is not a system that the users will be seeing on the website. The project's storage will not be made public for all to see. It will only be made viewable privately to shared individuals.

5.2 GitHub

The version control system that has been selected for this project is Microsofts GitHub. This is a distributed version control system where collaborators can pool in their edits to files. They can merge together new versions and have them separate from a main tree of code. Since they also record a history of edits, it is possible to revert to a previous version in case something fails.

5.3 Usage

The team will be using GitHub in order to keep a record and history of what has been accomplished in the project. The team will not only include code that is relevant to the project but also documentation files. It will be the central base of storing and sharing project code. Since GitHub is the designated storage place, it will be pertinent to organize some of the files and segregate them based on documentation or code.

6 SECURITY

Security is an essential piece of a web application especially when the application is handling sensitive data from users within a educational institute such as Oregon State University. With the implementation of Django as the framework there are a number of security advantages and disadvantages that should be noted for implementation.

6.1 Context

In terms of security, users will be performing various actions that will need to be secured such as file transfer, request, and log-in. On top of securing user actions, the application must also be protected from the Open Web Application Security Project (OWASP) top ten risks. The OWASP top ten is a collection of the most common security risks that modern web applications face which consist of risks such as SQL injection, broken authentication, cross-site scripting, and so on [6]. Thankfully, with the implementation of the Django framework the application will be protected from a number of these threats but even Django is not perfect. During development, securing the weak spots that even Django cant handle will be a key aspect to handle. Even after deployment security will be a concern as updates will introduce new bugs and flaws which will need to be handled.

6.2 Viewpoint: Administrators

Provided the service of the web application security is an implementation that most users will assume is there. Therefore, it is the responsibility of the administrators of the application to set and maintain the security standards for

the web application. Database security is most likely the largest concern to administrators for the project as the various functionalities introduce a range of routes that could lead to malicious users access. Access with malicious intent would be detrimental to the university, so securing our application through provided Django modules, implemented security ideals, and updates is essential. Data will also be sent between hosts so another task the administrators must face is the encryption of data across the network. While Django can provide the tools necessary to encrypt data it will be up to the administrators to implement those tools and verify that they are utilizing secure as well as efficient protocols.

6.3 Design Concerns

The major concern of security implementation if verifying that all cases and flaws have been considered. With the desired functionality and the OWASP top ten there are a good number of security risks that must be covered/handled. Covering most security risks should be straightforward as Django provides packages that can handle the most well-known security flaws and encrypt data traveling across the network between end hosts and the server. Django also provides various secure log-in packages such as SAML and LDAP which implement tools for fast, easy connection to the Oregon State University servers. While Django provides a good amount of security implementation it isnt exactly perfect. During development the addition of a blacklist, data sanitation, and UUID's for requests a long with file transfer will be necessary to prevent directory traversal within the application. Overall there are some concerns that need to be noted to secure the application.

6.4 Implementation

Establishing security for the application will be straightforward. As previously stated, Django provides numerous packages that handle a majority of security flaws that web applications face. The packages that will provide the necessary security are the Django CAS login package for login, HTTPS/SSL packages for secure protocol implementation, and the UUID package to set file names to a unique I.D. generated by the application to prevent traversal through file names. After implementation, the security of the application will still be under the works since Django is an open source framework therefore constant updates will be necessary for security updates. Package management must also happen so that packages are updated and packages which are out of date are substituted with a proper replacement. With the application secure after implementation the product will be closer to meeting the clients requirements while providing a safe environment for the web applications users.