
Student Letter of Recommendation Management Site

CS 461, CLIENT REQUIREMENTS REVISIONS 1.0

GROUP 8: THE LETTERMEN

OREGON STATE UNIVERSITY, FALL 2018

Authors: Lorenzo Ayala, Johnathan
Lee, Scott Waddington, Matthew
Kottre, and Mingwei Gao

Prepared for Justin Wolford

Signature

Date

Abstract

Oregon State University has a vast number of students who create relationships with their professors. As students continue their college careers they will be in need of letters of recommendation and one of the most popular sources are professors. However due to the massive increase in the student body, professors have faced a massive number of requests which can be difficult to track. On the other end students can have a difficult time providing the documents professors need to write an effective letter of recommendation. Currently the process for letters of recommendation is a bit of a hassle. Our solution to this process is a letter management site for OSU students and professors. Our site will be a hub for students to store files and make requests to professors for letters while professors can manage requests and submit completed letters. With this implementation we be able to create a smooth process for acquiring letters of recommendation.

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, Acronyms, and Abbreviations	2
1.4	References	3
1.5	Overview	3
2	Overall Description	3
2.1	Product Perspective	3
2.2	Product Functions	3
2.3	User Characteristics	3
2.4	Constraints	4
2.5	Assumptions and Dependencies	4
3	Specific Requirements	4
3.1	External Interfaces	4
3.2	Specific Functionality	4
3.3	Performance Requirements	6
3.4	Logical Database Requirements	6
3.5	Design Constraints	6
4	Gantt Chart	7

REVISIONS

Section	Original	New
---------	----------	-----

1 INTRODUCTION

1.1 Purpose

The purpose of this requirements document is to set as well as address the specifications of our web application. Within this document we will distinguish functionality, usability, performance, attributes, and design constraints without becoming too technical with project details. The given audience for our letter of recommendation will be the students and professors at Oregon State University which is inclusive of our client who is also a professor at the university. Therefore our project must be a mutual agreement between our group and client while providing a satisfactory product to our intended users. Leading to this document which will allow the client to assess our work while having a reference to compare our overall product to our provided documentation.

1.2 Scope

The letter of recommendation management site will be a hub for both students and professors of Oregon State University to cooperate together in handling necessary exchanges for a letter of recommendation. In order to meet our vision the application will implement two interfaces, one for students and one for professors. The two interfaces will allow us to populate the user data to fill in personal account data from a database. With the database we will be able to upload, download, and store required files for the students and professors for future need and management. Professors will also have a unique profile option to customize their requirements so that students will know what documents the professor requires and how much of a load (requests) they have along with access to those students files upon acceptance.

Professors receive a large amount of requests for letters of recommendation and with the implementation for this site we can smooth the process while providing benefits to both users. The benefits for students is the instant knowledge of necessary documentation for professors while professors themselves benefit from the custom requirements and instant access to student files to begin their letter. With this site we will be able to extend the services provided to current staff and students at Oregon State University.

1.3 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
User	Those who interact with our project.
LDAP	Acronym for Lightweight Directory Access Protocol which is a log-in protocol for authentication to applications accessing directories within a network/database
SAML	Acronym for Security Assertion Markup Language which is a common log-in protocol to authentication and authorization between parties.
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.

1.4 References

Currently no references are in use for this document however this is subject to change as our details and metrics are defined as the project continues.

1.5 Overview

Following this brief introduction we will describe our management site as a whole. Our description includes Product perspective, functionality, user characteristics, developer constraints, and group assumptions/dependencies. With our description we will provide an overall look into our vision of the management site for our client/readers. We will conclude the document with a breakdown of specific requirements.

2 OVERALL DESCRIPTION

2.1 Product Perspective

In terms of perspective, our management site is an extension to the various services provided by Oregon State University to students and staff. With the stated perspective in mind we will need to match our product to Oregon State University design style while implementing the necessary functionality. The management site is also dependent on implementation with the Oregon State University network. With both design/functionality and network necessities, we will need to implement a new service for letters of recommendation while utilizing existing network accounts and analyzing network efficiency of our application so it can handle massive amount of requests as well as document management that will occur.

2.2 Product Functions

Overall the product will provide two separate interfaces and experiences dependent on whether the user is a student or a professor. All data will be stored in a database so that users pages will be uploaded with existing files and so on. One a user is logged in and presented the correct interface there will be two different experiences dependent on if a student or professor is logged in. Students will be able to update or upload files for professors to view and/or download. Students can also make requests to professors for letters of recommendation while viewing the load a professor is currently under so that the student can determine if the professor is a good candidate. Professors will be able to personalize their profile in a sense of setting a preference list for documents they need to write a letter of recommendation. In addition professors can manage requests from students as well as gain access to students files for whom they are writing a letter of recommendation for. Combined these will make up a product that will make a more efficient process for letters of recommendation.

2.3 User Characteristics

Our product will have two separate users, students and professors/staff. Provided two specific users we can determine useful characteristics from both for our applications implementation. The characteristics we are concerned with are technical skills provided. Granted there is a wide range of skill provided from both users since users may have used applications for file requests and management while others may have no idea how to use such an application leading to a variety of interactions with our management site. Therefore most of our design will be dependent on the range of technical characteristics which will be determined through various surveys/interviews to gain a stronger grasp on our audience.

2.4 Constraints

Our management site will be very dependent on our users so through their characteristics which will be able to define constraints to our product. The general fields of constraints we will have to manage are users technical abilities and familiarity with this type of software. These two fields will be key in the design of our interface and functionality to adapt to these two characteristics. In terms of network constraints we will need to make sure our application can handle the various users and user requests on the application while not being too resource hungry on the Oregon State University engineering servers. We will also have to make sure that file management is efficient and our memory size is enough to store a large batch of files.

2.5 Assumptions and Dependencies

Since our project is to be a fully functioning application upon completion we assume our solution will be an effective extension of OSU's services to students and staff. With this large assumption there are some underlying dependencies to track in terms of maintenance and security.

Since our project will rely on open source libraries and tools there is room for insecurities arise. Open source is great because it is free and can be updated by a supporting community. However, with this structure secure flaws can arrive via updates and new releases to existing frameworks. These previously stated issues can go hand in hand with maintenance of the site as updates will need to be done frequently to patch any flaws or errors while checking if the update itself affects performance. These issues give us our two dependencies of those who are maintaining the site will keep it up to date while checking for any current flaws in the system and personal information could be stolen. Other dependencies can be determined during user surveys and prototyping.

3 SPECIFIC REQUIREMENTS

3.1 External Interfaces

For the letter of recommendation management site we will be implementing two user interfaces based on account type which are student and professor/staff. Each interface will have its own unique selection of options for each user based on functional requirements. Overall we have a vague description of our interfaces as they will develop with our design, development, and testing of our application. During development we will have a more concrete description of our interfaces however we know in essence they should be simple and user friendly for quick, easy-to-learn use.

3.2 Specific Functionality

We briefly discussed our overall objectives in our websites functionality but here we will breakdown specific functionality and how we intend on implementing them.

- Secure Log-in

Description: Users will be provided a log-in page where they must pass in credentials that match their OSU account credentials.

Implementation: A web page will be given to the user requiring a log-in which will be authenticated using LDAP or SAML with the Oregon State University accounts database to check and approve credentials.

- Upon log-in users data will be loaded into the provided interface

Description: When a user(student/professor) from OSU logs into our application they will be given one of our two interfaces with existing information loaded onto the page based on their credentials. If no data exists they will be provided a basic empty interface to begin.

Implementation: We will implement this aspect by utilizing a MySQL database which we will run user credentials against to retrieve necessary information for each specific user.
- Students can upload and update files

Description: Students will be able to upload and update files on their main page which professors can see if they accept student requests.

Implementation: To implement this functionality we will use the MySQL database to store uploaded files using a UUID binded to the files linked to the user to ensure that when credentials are ran the correct files are stored/displayed for the user.
- Students search and request letters from professors (also view professor requirements)

Description: Students will be able to search for professors and request letters from them via a search bar while viewing their load and preferred documents.

Implementation: Via a search bar students will be able to search for professors at the university based on professor accounts from the university network and make requests through code functionality which will also display the current accepted requests the professor has made based on a counter linked to the professor account and display the professors preferred documents based on a preference variable stored in our database.
- Professors can personalize their preferred documents

Description: After log-in professors will be able to personalize their profile in a sense of what documents they prefer from students to develop an effective letter of recommendation.

Implementation: For personalizing professor professors we will have the database contain an existing column necessary to store the preferences and upon viewing a professor profile a query is ran to grab this information for display.
- Professors can handle requests

Description: Professors can handle various requests from students by either accepting or declining (can provide reason for decline).

Implementation: A request list will be saved for each professor so that upon viewing we will grab the list of requests from the database which the professor can make a decision on.
- Professors view student documents

Description: Upon accepting a student request the professor will be given access to the student documents which they can use/download for developing a letter.

Implementation: After a request acceptance the professor will be given permission to view the students file page

and view their uploaded documents.(operation for providing documents will be similar to student perspective)

- Professors can submit complete letters into student files.

Description: Professors can submit a complete letter to the student which will then appear in their files with an indicator so that students know that that file is a complete letter of recommendation with a reference to the professor who completed a specific letter.

Implementation: We will implement this functionality taking the submitted file and storing it to the students files while implementing case to implement CSS to highlight the letter and link the professor the letter by running the professor credentials with the submitted file and linking to their profile.

3.3 Performance Requirements

In terms of performance we essentially want all functionality to be smooth and error free to provide a nice experience to the user while keeping congestion low on the network for fast use. From a network perspective we will have to make sure that the amount of users and streams of data passed from files can be handled properly to prevent lag and loss of files. Performance metrics will grow overtime as we defined more details of our project but overall we want the performance to be fast as well as efficient.

3.4 Logical Database Requirements

Our database will be split for students and professors with each having their own unique setup of data. Our database should be able to handle the data types of docx, pdf, and strings from the users. Upon storing of these inputs we will need to utilize UUID's to link files to accounts within the database and to secure the database so data can not be iterated through.

3.5 Design Constraints

Design of our project will be constrained to user input through various surveys and prototype testing of functionality as well as user interfaces. For now we are constrained to a design that is simple as we want our users to come to an application that should feel familiar and easy to learn if they are new. While our constraints are limited for now, however we will have a growing set of constraints through testing and user feedback in the future.

4 GANTT CHART

