Software Requirements Specification

for

Program Review Information System Management

Version 1.0 approved

Prepared by
David, Leanne
McLees, Andrew
Sarenas, Justin
Solis, Ben Jair

September 4, 2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial Draft	2017-12-08		1

1. Introduction

1.1 Purpose

The document specifies functional and nonfunctional requirements for the CSULA Program Review Subcommittee document storage and management system. This document specifies all requirements for version 1 of the system.

1.2 Intended Audience and Reading Suggestions

This document is intended for developers, end users, and testers to get an overview of the overall process of a program review. They shall also see what is integrated within the system to reflect the program review process.

1.2.1 Developers and other technically knowledgeable people should refer to this document to gather the technical specification of the software requirements.

1.2.2 For end users and testers, this document shall provide the intended purpose of the system along with its features and functions.

1.3 Product Scope

The software product will be called Program Review Information System Management. PRISM will be a workflow management tool including an optimized document storage and management system where users will be able to store and download all files related to program reviews. The system will also have other features such as e-mail notifications and a calendar so the document review process runs more effectively. Once the software is released, PRISM will replace the current document system that the committee currently uses. The software will store all relevant documents pertaining to program reviews such as the review document, templates, agendas, and minutes. This software will be an improvement to their previous system where they sent e-mails back and forth with documents attached.

1.4 Definitions, Acronyms, and Abbreviations

Acronyms:

- CSULA - California State University of Los Angeles
- E2E - End-to-End (Testing)
- HTTP - HyperText Transfer Protocol
- HTTPS - HyperText Transfer Protocol over TLS
- MOU - Memorandum of Understanding
- PRISM - Program Review Information System Management
- PRS - Program Review Subcommittee

- SAML - Security Assertion Markup Language
- SMTP - Simple Mail Transfer Protocol as specified in RFC 5321
- TLS - Transport Layer Security

Definitions:

- Document - A single document being produced that contains multiple revisions of its file. (i.e. Self Study, Summary Report, etc.)
- File - Consists of arbitrary data that holds content for a document (i.e. Self Study v1.docx, Self Study v2.docx, etc.)
- Program - A degree program under specific college departments (i.e. Computer Science, Electrical Engineering, etc.)
- Review - A single review process for a department's degree program during a specific year (i.e. CS MS 2018, CS BS 2024, etc.)
- Shibboleth - An implementation of SAML authentication that provides an Identity Provider among other products

1.5 References

Angular Documentation - https://angular.io/docs

The Angular documentation is a collection of a tutorial, specific API documentation, and techniques for achieving common functionality in Angular. It is open source on GitHub, meaning that anyone can contribute to the documentation via pull requests, making it an up-to-date and valuable resource when working with Angular.

Mongoose Documentation - http://mongoosejs.com/docs/guide.html

The Mongoose documentation describes the Mongoose API along with examples of common use cases. As none of the team members had prior experience working with Mongoose, it has already proven itself a valuable resource.

NodeJS Documentation - https://nodejs.org/dist/latest-v8.x/docs/api

The NodeJS documentation describes the NodeJS API. It will be useful when using features offered by NodeJS such as file operations.

Express Documentation - http://expressjs.com/en/api.html

Though the system will be a single-page application, Express will still be used for its middleware functionality and for REST API endpoints. Thus, its API documentation will likely be of use.

PrimeNG Documentation - https://www.primefaces.org/primeng

PrimeNG will be the system's user interface library. Its documentation will be useful as it contains live examples of each component it provide on its website complete with source code.

Node Express Mongoose Demo - https://github.com/madhums/node-express-mongoose-demo

This demo application is a useful resource for finding common idioms in Mongoose development. It is a blog application containing many features common across full-stack applications in general including user authentication and CRUD operations.

The Twelve-Factor App - https://12factor.net

The twelve-factor app is a methodology for creating applications in general to avoid common pitfalls of development. It presents some best practices which will be used in the development of our system.

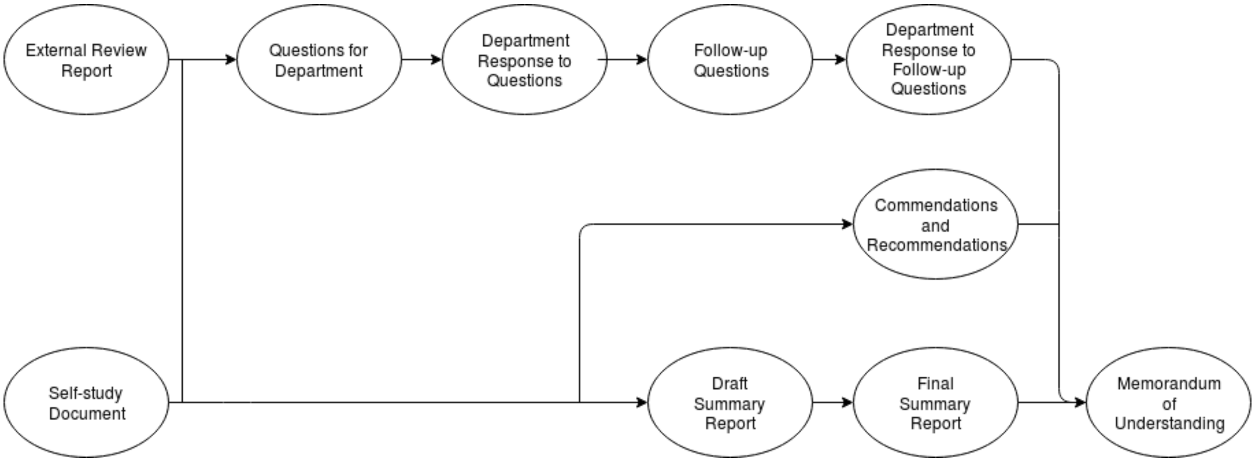
Murray, Nathan, et al. *The Complete Guide to Angular 4*. 4th ed., CreateSpace Independent Publishing Platform, 2017.

This book presents a guide to Angular 4 that presents the important features of Angular 4 in more depth than the online documentation. It will be useful as many of the group members have experience with older versions of AngularJS, but not with Angular 4.

2. Overall Description

In order to understand the features that automate it, it is first necessary to understand the program review process. The program review process includes selecting a review team of experts, preparing a self-study document, conducting a visit by the review team, preparing a response to the review team recommendations, submitting the response and action plans to the Dean, and revisiting progress on action plans.

Below is an overview of the documents involved in the review process.


External Review Report
Questions for Department
Department Response to Questions
Follow-up Questions
Department Response to Follow-up Questions
Commendations and Recommendations
Self-study Document
Draft Summary Report
Final Summary Report
Memorandum of Understanding

Scheduling of Reviews:

Each program is reviewed on a regular cycle. Reviews may be scheduled to coincide or coordinate with other reviews (e.g. accreditation) or to address a special concern about a particular program. Reviews that coordinate with other reviews should still address the core criteria for academic program review described in the self-study section. Usually, the length of time between reviews is no more than five years.

The Office of the Provost and Vice President for Academic Affairs will determine the specific review cycles for each department/division/school and interdisciplinary program by consulting with the administration of the colleges. For new programs, a five year development period is required to establish a valid measure of their productivity. The President, Provost, and Vice President for Academic Affairs, or the Educational Policy Committee may request additional reviews of programs in any given academic year for purposes of planning or to satisfy a request from the Chancellor’s Office or the Board of Trustees. Reviews other than the university reviews may of course be conducted by colleges or departments/divisions/schools at times other than those scheduled by the Office of the Provost and Vice President for Academic Affairs, and they may request a university review on their own initiative. The schedule for program review and all subsequent modifications will be published by the Office of the Provost and Vice President for Academic Affairs and distributed to the faculty.

Selection of Visiting Team and Visit Schedule:
When a program is reviewed, the faculty members select a team of recognized peers from distinguished programs in the CSU system or outside the CSU system to visit the campus and to consult with faculty, staff, students, and administrators on the future plans of the program. The college Dean approves of the review team members.

The college is responsible for coordinating the schedule for the visit with the unit undergoing review. The length of time the team is on campus is at least two days. Then, the team prepares a written evaluation; which will become part of the permanent Program Review file. Sufficient funds to cover the expense of the team will be included in the budget of the University.

External Review Protocol:
All programs to be reviewed by the Program Review Subcommittee (PRS) are subject to an independent evaluation by external reviewers. In the case that an external review or national accreditation has been conducted within the previous three years, the department/division/school will be exempt from this requirement. The external reviewers will report their findings to the PRS and to the appropriate department/division/school or college within two weeks of their visit.

Internal Review Protocol:
The initial review of the self study consists of a discussion of draft questions gathered by the PRS through a meeting with the presence of the Dean or designee. In addition, the Dean or designee may add input at these meetings by means of attending or through e-mail/written correspondence. The approved questions are then sent to the college and program the following week. The college and program under review have two weeks to respond to the questions addressed by the PRS.

Following the initial review is a meeting with the department; consisting of the Chair, Department representatives, and Dean. A discussion of the responses from the college and program is held at the meeting. A list of follow-up questions are then sent to the college and program the following week. The college and program will then respond to these questions two weeks after they were sent.

Next, the PRS lead reviewer(s) are responsible for completing a draft of the commendations and recommendations, due the week before a recap meeting is held. In this recap meeting, the presence of the Dean or designee is required. A discussion of the commendations and recommendations is held at the meeting with input from the Dean or designee. What follows is a draft summary report due to the PRS and program the week after the recap meeting is held.

The draft summary is then discussed at another meeting with the Dean or designee, Chair, and department representatives. Here, the department provides feedback on the draft summary. The final summary report is due a week after the draft summary report is discussed.

Ultimately, the final summary is approved by the Chair and Dean or designee in the week of finals.

Accredited Programs:
Programs that have undergone accreditation must also undergo program review, albeit modified, so as to utilize the work done for the accreditation. Thus, one year prior to the scheduled program review the program will develop a document comparing the accreditation standards and criteria with those of the program review. This document will be submitted to the college dean. An ad hoc committee then reviews this document along with the accreditation documents. This ad hoc committee is composed of the executive secretary of the PRS, the chair of the PRS, the college dean, and the chair of the program. At the conclusion of the review of documents, the ad hoc committee will report to the Provost and Vice President for Academic Affairs the extent to which the program's accreditation documents meet the requirements for the program review self-study, and identify any areas that should be addressed in a modified program review self-study.

The PRS develops a report after reviewing the program and its report is transmitted to the department/division/school and college for their responses. The report of the PRS shall be prepared by the voting members of the subcommittee. The PRS report and response will be submitted to the Educational Policy Committee.

2.1 Product Perspective

The product is a full-stack web application and will be required to interface with an external authentication server administered by CSULA.

This software has many similarities to other workflow automation, document storage, and review systems. The primary motivation for creating a new system is that a new system can be tailored specifically towards the needs of the Program Review Subcommittee and can be kept within CSULA's network.

2.2 Product Functions

The following list summarizes the major functions of the system.

- Track and provide an interface to view the progress of each review as it proceeds through the review process
- Store, track the progress of, and allow collaboration on review documents
- Store and automatically source new documents from review document templates
- Store meeting agendas and minutes
- Maintain a calendar of PRS meetings and send e-mail notifications upon changes
- Track which programs are due for review
- Send e-mail notifications upon events relevant to the user

Storage of a document is used to mean providing CRUD operations on the document. Progress tracking means providing the user with the ability to determine the current progress of each document and each review according to the review process. Collaboration features include a commenting system and versioning system. All features are detailed in the functional requirements in section 4.1 of this document.

2.3 User Classes and Characteristics

User classes are broken down to the various members involved in the program review process.

2.3.1 Administrators

- Administrators will have full control and access to the system so that they can ensure the program review process proceeds smoothly. They will be able to view the same screens and do the same tasks as the other users.

2.3.2 Non-PRS Faculty (Department Chair, Dean, Associate Dean)

- Non-PRS faculty shall have the power to upload only to documents they produce as a part of the review process
- Non-PRS faculty shall be able to download questions for their programs only after the questions document is finalized

2.3.3 External Reviewers

- External reviewers shall be given a one-time access to the system. They shall only have the ability to upload and submit their final external report.

2.3.4 Lead Reviewers

- Lead reviewers shall be able to upload revisions of documents pertaining to the program for which they are the lead reviewers

2.3.5 Committee Members

- Committee members shall have limited access and may only view active program reviews. Their features include mainly reading and writing comments to any version of a document.

2.4 Operating Environment

The software is planned run on a server using Mongoose, Express, Angular 4, and NodeJS to perform various functions.

The software will be required to interface with existing authentication systems of California State University, Los Angeles. These systems will be necessary for user authentication and thus will affect the software.

2.5 Design and Implementation Constraints

Hardware limitations will be largely determined by budget. Based on current market prices for server hardware, these limitations should not be an obstacle in obtaining a server that will allow the software to meet the performance requirements.

Interfacing with the current authentication systems of the university can be done using SAML. Implementation of the software (prior to deployment) will be largely unaffected by this requirement as it is specific to authentication and not critical to the function of the software until it is deployed.

The software will play a critical role in the program review process; therefore, it must be highly reliable. This may increase the amount of time necessary for implementation as time is focused on testing to ensure reliability.

As part of its operation, the system will have access to many documents that are not accessible to the public. Thus, ensuring security of these documents and strength of the authentication and access control systems will be a part of implementation.

The documents created as a part of the program review process must be stored and will require disk space. The amount of disk space on a modern hard drive is enough to host many years of documents, as the documents contain no multimedia. Constraints on permanent storage should not prove an issue.

2.6 User Documentation

A PDF guide will be delivered separately from the software. It will explain how to use all features the software provides.

2.7 Assumptions and Dependencies

As mentioned in section 2.5, the product will be required to interface with Shibboleth at CSULA for authentication. This requires access to CSULA's Shibboleth system to be granted by the CSULA Information Technology Services. Without access to this system, the requirements specified in this document would need to be changed so that authentication through the existing mechanisms of the university is not required.

Planned dependencies include NodeJS version 8, Angular 4, the Passport authentication library, bcrypt password hashing, and Mongoose for database schema.

2.8 Apportioning of Requirements

No requirements are planned to be delayed until future versions of the system.

3. External Interface Requirements

3.1 User Interfaces

The only user interface is hosted in the user's web browser with internet access. The browser is used to navigate through the system and communicate with the database. The client-side portion of the application, running in the user's browser, is essential to this functionality. Thus, all interfacing with the system is to be done using a JavaScript-enabled web browser.

Common features found in websites are implemented to help users interact with the system. To facilitate the navigation of a user, a standard navigation bar is implemented on most or all pages. For help with finding specific stored information, a search bar is implemented where deemed necessary. In addition, standard buttons are implemented for actions to be taken by users. These buttons are colored and spaced nicely to allow users to distinguish button functionality. In the event of an error, the system displays a static error page with the error code displayed.

For details of the user interface design pertaining to each page, please see the design document for the system.

3.2 Hardware Interfaces

Because the software is ultimately a web-based system that handles file archiving and other digital document functionalities, external hardware devices are not required.

3.3 Software Interfaces

The main software interface that the software must interact with is the existing authentication system of the university, an implementation of SAML, Shibboleth. The application must interface with the Shibboleth Identity Provider to authenticate users through the university system.

The software will also be required to send e-mail notifications. This will require access to an e-mail server, which will be interfaced with using SMTP. The e-mail server will be run on the same server as the system and thus will not be external.

3.4 Communications Interfaces

The software will be run on a server which will server HTTP requests by a user’s web browser. Messages will be valid HTML which will be parsed by a user’s web browser. Data transfer rates must be sufficient to upload and download files in review, which are typically under 10 kilobytes.

Communication with the Shibboleth Identity Provider of the university will require use of SAML. This communication is required to authenticate users using their university-issued credentials.

4. Requirements Specification

4.1 Functional Requirements

Functional requirements define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs.

Requirements Related to the Review Process (1.1, 1.2, ...)	
Requirement No.	Requirement Description
1.1	The administrator shall be able to create a new review.
1.2	The administrator shall be able to access any review.
1.3	The administrator shall be able to update any review.
1.4	The administrator shall be able to delete any review.
1.5	The system shall track the status of each document pertaining to a review.
1.6	The system shall display the status of a review.
1.7	The system shall display the documents of a review.
1.8	The system shall track the estimated completion date of each document.
1.9	The system shall allow lead reviewers and administrators to change the estimated completion date of each document.
1.10	When a review is initiated, the system shall import the templated documents to be part of the review's documents.
1.11	The system shall allow users to upload non-standard documents.
1.12	The system shall allow users to access non-standard documents.
1.13	The system shall allow users to update non-standard documents.
1.14	The system shall allow users to delete non-standard documents.
1.15	The system shall allow administrators to upload admin-controlled templates.
1.16	The system shall allow administrators to access admin-controlled templates.
1.17	The system shall allow administrators to update admin-controlled templates.
1.18	The system shall allow administrators to delete admin-controlled templates.
1.19	The system shall allow administrators to finalize a document.
1.20	When a review is initiated, the system shall be able to send an e-mail to the chair of the department of the program under review.
1.21	When the administrator creates new reviews, the system shall provide a list of program which should be up for review during the current year.

Requirements Related to Document Collaboration (2.1, 2.2, ...)	
Requirement No.	Requirement Description
2.1	The system shall allow users to upload new revisions of a document.
2.2	The system shall store all revisions of a document.
2.3	The system shall be able to convert a Microsoft Word 2013 document to a PDF file.
2.4	The system shall allow users to download the current revision of a document.
2.5	The system shall be able to send e-mail notifications to selected PRS members when a document is uploaded.
2.6	The system shall allows users to specify e-mail notification recipients by name.

Requirements Related to Miscellaneous Storage (3.1, 3.2, ...)	
Requirement No.	Requirement Description
3.1	The system shall be able to store agendas and meeting minutes uploaded by a user.
3.2	The system shall be able to store resources to be made available to all PRS members.
3.3	The system shall be able to send e-mail notifications to selected PRS members when an agenda is uploaded or changed.
3.4	When sending e-mails, the system shall be able to attach a PDF document to the e-mail.

Requirements Related to Calendar (4.1, 4.2, ...)	
Requirement No.	Requirement Description
4.1	The system shall store the dates and times of meetings of the PRS.
4.2	Users shall be able to create a new meeting for the calendar to track.
4.3	The system shall display all the dates and times of meetings of the PRS to PRS members.
4.4	Users shall be able to update the date and time of a meeting.
4.5	Users shall be able to delete a meeting.
4.6	The system shall be able to associate a list of meeting agendas or minutes with each meeting.
4.7	Users shall be able to modify the list of associated meeting agendas and minutes for a meeting.
4.8	The system shall send e-mail reminders to PRS members 24 hours before a meeting with all associated meeting agendas as PDF attachments to the e-mail.
4.9	The system shall send e-mail notification to all PRS members when a meeting is deleted.

Requirements Related to Comment System (5.1, 5.2, ...)	
Requirement No.	Requirement Description
5.1	The system shall allow users to create comments on a document.
5.2	The system shall display the comments of a specified document to users.
5.3	The system shall allow users to comment on an existing comment.
5.4	The system shall support only single nested comments.
5.5	The system shall display the author of each comment.
5.6	The system shall display the date of each comment.
5.7	The system shall allow users to edit their comments.
5.8	The name of the current revision at the time of the creation of the comment shall be displayed with each top-level comment.

Requirements Related to the User Authentication (6.1, 6.2, ...)	
Requirement No.	Requirement Description
6.1	The system shall direct users to a login page if that user is not currently logged in.
6.2	The user shall be blocked from accessing the system if the user is not logged in.
6.3	The system shall allow users to log in using their CSULA-provided credentials if they are CSULA faculty.

4.2 External Interface Requirements

4.2.1 HTTP(S)

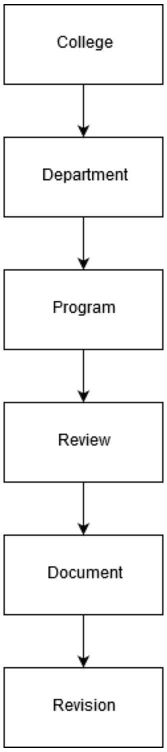
- Will be used to send the client-side portion of the application to the user's browsers
- Will be used for two-way communication between the REST API and the client-side portion of the application
- Runs on TCP which provides reliable transfer

4.2.2 SAML

- Will be used to allow authentication of users via their university credentials
- Will be used to communicate with the university's Shibboleth implementation in order to authenticate users via their university-provided credentials

4.3 Logical Database Requirements

See diagram on right for the overall data hierarchy. Exact database schema details will be left to the design document.



All persistent data will be stored in MongoDB, although files may be stored outside MongoDB. MongoDB will provide sufficient performance for reading and writing the data. Mongoose is planned to be used for all interfacing with MongoDB.

Data Entities

- User
  - Username
  - Full name
  - Authentication data
- Group
  - Name
  - Members
- College
  - Name
  - Dean(s)
- Department
  - Name
  - Chair(s)
  - College

- Program
    - Name
    - Department
- Review
    - Program
    - Start date
    - Stage
    - Review finalized
- Stage
    - Documents
    - Document completion dates
    - Document prerequisites
    - Document statuses
- Document
    - Name
    - Current Revision
    - Revisions
    - Comments
- Revision
    - Name
    - File

4.4 Design Constraints

Interfacing with the current authentication system of the university can be done through SAML. Passport, a NodeJS library for authentication, is an example of a library that will allow this. Implementation will not be heavily affected by authentication through SAML because there are existing libraries to handle this functionality.

Increased reliability of the software should be achieved through a combination of E2E and unit testing.

The memory required to meet the performance requirements is significantly lower than the amount in modern consumer-grade personal computers. The low number of concurrent active users means memory will not be a significant constraint.

The disk space required to store the documents should not present a problem. The lack of multimedia in the documents created limits their size to approximately 100 kilobytes. This means millions of documents may be stored on a modern hard drive. The software itself and its dependencies will not present a storage problem.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

Dependability

The system should maintain an uptime of greater than 99.5%.

Usability

The system should be usable by all members of the user base after a user consults the PDF manual.

Interoperability

The system shall support access via the following web browsers:

- Google Chrome (Version 55.0.2883+ and subsequent versions)
- Mozilla Firefox (Version 50.0+ and subsequent versions)

The system may support access via the following web browsers:

- Microsoft Edge (Version 15.14986 and subsequent versions)
- Internet Explorer (Version 11.0+)

Verifiability

Proper operation of the system should be verified by conducting E2E testing with the browsers it is required to support.

5.2 Safety Requirements

The software will be a web application and as such will not have any safety requirements.

5.3 Security Requirements

The system must authenticate users based on their university-provided credentials. Access to any part of the system other than the login page(s) will require prior authentication.

There are no external policies regarding security which must be satisfied.

5.4 Software Quality Attributes

The customers value the software for the long-term role it will play in the document review process. Due to this, they value the robustness and reliability of this system directly. Details regarding the robustness and reliability requirements of the system may be found in section 5.1 of this document.

In the long term, the software will need to be maintained; this maintenance can be made simpler by organized program code and tests.

The system is required to be operable by all users. This should be verified by testing the software with members of the future user base.

5.5 Business Rules

The system will only be accessible to faculty members associated with program review and the external reviewer. Administrators will have full access to the system while also in charge of granting and removing permissions from other faculty members. The college department users which consists of the Chair, Dean, and Associate Dean will be able to upload documents that they are relevant to as well as download the questions pertaining to the review. The External Review will only have permission to access the site once to upload their final external review document. The Lead Reviewer will be able to upload and update the revisioned document. The Committee Members will be able to view the program they are assigned to and comment on the different revisions of the review documents as well as view previous approved documents.

6. Other Requirements

The system does not have any other requirements.