CSE Annual Performance Review Automation Design Documentation

Quinn Wass, Richard Pallangyo, Amanda Zheng, Malav Dipankar

Version 1.0.1 - 11.8.2020



# Contact

## Developers:

Quinn Wass [wassq@seattleu.edu](mailto:wassq@seattleu.edu)

Richard Pallangyo [rpallangyo@seattleu.edu](mailto:rpallangyo@seattleu.edu)

Amanda Zheng [zhengchujun@seattleu.edu](mailto:zhengchujun@seattleu.edu)

Malav Dipankar [mdipankar@seattleu.edu](mailto:mdipankar@seattleu.edu)

## Faculty Advisor:

Eric Larson [elarson@seattleu.edu](mailto:elarson@seattleu.edu)

## Sponsors:

Dr. Michael Quinn [quinnm@seattleu.edu](mailto:quinnm@seattleu.edu)

Dr. Pejman Khadivi [khadivip@seattleu.edu](mailto:khadivip@seattleu.edu)

Dr. Donna Sylvester [dsylvest@seattleu.edu](mailto:dsylvest@seattleu.edu)

Dr. Cheryl Wotus [wotusc@seattleu.edu](mailto:wotusc@seattleu.edu)

## Table of contents

[Contact 2](#_Toc56017484)

[Developers: 2](#_Toc56017485)

[Faculty Advisor: 2](#_Toc56017486)

[Sponsors: 2](#_Toc56017487)

[Table of contents 3](#_Toc56017488)

[Introduction 4](#_Toc56017489)

[Requirements 4](#_Toc56017490)

[Architecture 7](#_Toc56017491)

[Architecture Overview 7](#_Toc56017492)

[4 + 1 Diagrams 7](#_Toc56017493)

[System Overview 8](#_Toc56017494)

[Versioning 9](#_Toc56017495)

[Models 10](#_Toc56017496)

[Use Cases 10](#_Toc56017497)

[System Diagrams 12](#_Toc56017498)

[Layer Architecture 15](#_Toc56017499)

[Deployment Diagram 16](#_Toc56017500)

[Physical Diagram 16](#_Toc56017501)

[State Diagrams 17](#_Toc56017502)

[Wireframes 22](#_Toc56017503)

[Version Control Table 32](#_Toc56017504)

# Introduction

### Executive Summary

This design document is created by CS 21.16, a 2020-2021 senior capstone team, for the Seattle University (SU) College of Science and Engineering (CSE) Faculty Annual Review Process (APR) Automation project. The purpose of this document is to outline the APR Automation design and specification workflow, system architecture, and system requirements—as agreed upon by both the team and sponsors.

The audience of this design document is the student project team and project sponsors. The project development will be stretched over three quarters of the 2020-2021 SU academic year, with most of the work done when school is in session. The Fall quarter will involve creating a scope document, backlog, design document, project plan document, and research of the tools we will need to start the developmental stage. The development of the software will take place in the winter quarter. While testing and debugging will occur throughout our development process, the spring quarter will be dedicated to testing and debugging the APR system. The Spring quarter will also be used to prepare for senior project presentations. As a team, we plan to use winter break to do more research, learn technologies needed, and set up a server to deploy the project.

The final product will be an electronic system that will allow annual performance reviews for faculty in the CSE to be performed online. The electronic system aims to save faculty time by automating tasks such as pre-filling data in the APR process.

### Requirement and Design Summaries

In summary, using the APR web app, a faculty shall sign in using their Seattle University credentials, upload their CV, complete their APR forms digitally as well as store completed forms for future reference. Also, the dean/chair shall review faculty’s form, provide faculty with feedback, and approve form completion. Therefore, the system will let a faculty complete a form; the department chair reviews the form and adds comments, and the dean reviews the completed form.

The design diagrams model the decisions we have made to meet our sponsors’ requirements and provide a suitable foundation for project implementation. This design document provides high-level diagrams and descriptions of how the APR system will function, how the user will interact with the system and how the system will be developed to fulfill the requirements of our project sponsors. Additionally, use case diagrams, system models, and architectural views will illustrate the system features for a clear picture of what is to be developed in the project.

# 

# Requirements

### Functional Requirements

|  |  |
| --- | --- |
| **Description** | **Targeted User** |
| Clients shall be able to save their partially completed form and retrieve their work from where it’s left off, see Work in Progress(Faculty/Chair). | Faculty/Chair |
| Optional Fields functionalities shall be available when filling out a form. Which will alert the client that it is not mandate to complete, see Work in Progress(Faculty/Chair). | Faculty/Chair |
| CV Upload: Clients shall be able to upload their CV as part of the APR process, see Faculty Homepage, Chair Homepage. | Faculty/Chair |
| Admin Control: Admin shall have control of view and add or remove all the forms and editing positions of the user, or a to re-assign a new admin, see Admin Homepage, Individual Form Entry. | Admin |
| Goals Section: The client shall use this feature to set and write down their goals for the next year. These goals can be reviewed and evaluated by the professor and carried forward to the following year. | Faculty/Chair |
| Prefill forms: Clients shall be able to prefill certain section/filed if it’s the same as the previous years. Such as name/position (if hasn’t been changed). | Faculty/Chair |
| Chairs and Dean shall be able to see which of their colleague has turned in the form, either for review or submit for approval, see View Colleague (Chair/Dean) | Dean/Chair |
| Faculties shall be able to sign off on forms before pass to the Dean | Faculty |
| Clients shall be able to export their current forms or old ones, see Current Form (Faculty/Chair) | Faculty/Chair |
| Clients shall be able to preview their current forms or old ones, see Current Form (Faculty/Chair) | Faculty/Chair |
| The APR service shall only allow users to access sections in which their permission is able to. | All |

### Non Functional Requirements

|  |
| --- |
| **Description** |
| The APR service shall be written in a way that allows for reusability and extensibility of components so that other developers can extend the project. |
| The server shall be stateless in order to scale out into multiple nodes if needed. |
| The server, libraries, and client shall be written using portable, OS agnostic languages, packages, and frameworks. |
| The server shall be able to handle the several hundred users at peak usage times. |
| The server shall not need to always be running. It can be restarted when not in the APR period. |

To visualize our tool, you can view the following mockups:

* [Login Page](#_Login_Page)
* [Faculty Homepage](#_Faculty_Homepage_1)
* [Chair Homepage](#_Chair_Homepage)
* [Dean Homepage](#_Dean_Homepage)
* [View Colleague (Chair/Dean)](#_View_Colleague(Chair/Dean))
* [Current Form (Faculty/Chair)](#_Current_Form_(Faculty/Chair))
* [Work in Progress(Faculty/Chair)](#_Work_in_Progress(Faculty/Chair))
* [Admin Homepage](#_Admin_Homepage)
* [Individual Form Entry](#_Individual_Form_Entries)

# Architecture

## Architecture Overview

This section outlines the key architectural designs for the CSE Annual Performance Review Automation project. The architecture section is composed of several subsections including the System Overview and induvial component architectures.

## 4 + 1 Diagrams

### Logical View

You can view the various logical workflows via the linked model diagrams:

* [Faculty Workflow](#_Faculty_Workflow)
* [Dean/Chair Workflow](#_Dean_/_Faculty)
* [Admin Workflow](#_Admin_Workflow)

### Process View

Here is the overall [System Processing Diagram](#_System_Diagram) that we will use to guide the flow of between client, server and database. Here is a [Load-Balancing variant](#_System_Diagram_with).

Here are the State Diagrams for the following processes:

* [Login](#_Login)
* [Faculty Homepage](#_Faculty_Homepage)
* [APR Form](#_APR_Form)
* [Dean/Chair Review](#_The_Dean_or)
* [Admin Login/Update Database](#_Admin_logging_in)

### Development View

For a deep dive into the development view, please see [System Overview](#_System_Overview).

If you would simply like to view the models, please visit [Models/System Diagrams](#_System_Diagrams_1).

### Physical View

Our project has a limited Physical View, with the core physical interaction is between the browser, server and databases. A [layer diagram](#_Layer_Architecture) is linked for a more visual understanding.

We also have a [deployment diagram](#_Deployment_Diagram) to describe how we plan to deploy our product and a [physical diagram](#_Physical_Diagram) that outlies the hardware usage of our system (note that all connections in the physical diagrams are internet connections).

### Scenarios and Use Cases

#### Faculty Scenario

The workflow for the faculty scenario can be found in the [Faculty Workflow](#_Faculty_Workflow) use case.

Sally is an associate physics professor in the College of Science and Engineering at Seattle University. She would like to fill out her APR form for the year 2048. She has goals from the past year she would like prefilled. Sally logs into the APR website and clicks create form. She then begins filling out her form. She needs to take a lunch break and hits save and closes the website. Sally knows that because the system supports incomplete forms, she can resume her progress later. After a tasty and filling lunch, Sally logs back into the system and resumes we work. Sally finishes her work and hits submit. The form is moved to be reviewed by the faculty chair.

Later, the faculty chair makes comments on the form. Sally logs back in and acknowledges the comments and hits approve. The form is marked completed. It is locked, and the dean and chair can view the completed document.

#### Chair / Dean Scenario

The workflow for the chair / dean scenario can be found in the [Chair / Dean Workflow](#_Dean_/_Faculty) use case.

Alex is an environmental science professor in the College of Science and Engineering and is also the faculty chair of the environmental science department. Alex logs into the APR system and selects the chair profile, which is separate from their faculty profile. Alex notices that Mark has submitted his APR form to be reviewed. Alex clicks Mark’s form and reviews it. They leave comments on several sections and approve it. Alex then looks in the in-progress list and sends an email to Lindsey to remind her to complete her APR as it is past due.

#### Administrator Scenario

The workflow for the admin scenario can be found in the [Admin Workflow](#_Admin_Workflow) use case.

Jorge is a System Administrator in the College of Science and Engineering at Seattle University. He oversees loading all the faculty chairs in for the year 2022. He logs into the APR system and is automatically taken to the Admin profile since it is his only profile. Jorge selects the assignment section and removes the old mathematics faculty chair. He then adds the chair for the mathematics department from a drop down of applicable faculty members. The faculty chair profile is added for that user.

Possible Use Cases are covered in under [Models/Use Cases](#_Use_Cases).

## System Overview

The CSE Annual Performance Review Automation System Architecture is designed to be built from components as opposed to a monolithic architecture used historically by many software projects. If desired these components can be containerized and distributed as desired.

The user facing code in the front end will be divided up into components individually rendered through client-side rendering in the browser. These will be built in JavaScript using React JS. This will take load of off the Universities servers. These front-end components will utilize shared components to create a uniform experience throughout the application.

The core logic will be composed of shared libraries written in .NET. These will be built into separate maintainable NuGet packages than can be distributed and utilized as needed. These components will be fully decoupled from each other. Additionally, the abstraction / interface layer will be fully decoupled from the implementation. The core libraries will also contain the business logic and models required for data processing and representation.

The API will be written using ASP.NET Core and will utilize the core libraries written in .NET. The API will contain the controllers to drive the API. The API will be stateless. Each controller will only reference the core libraries and will not be coupled to other controllers. This will allow the API to be componentized and broken up, as one might do in a microservice architecture, if the performance tradeoffs become worth it at a future date. Additionally, since the server is stateless, load balancing can be implemented easily if needed. The average load on this application is expected to be quite small due to the use cases the client has indicated, however, if this changes the server can easily be replicated.

The system diagrams can be found under [Models/System Diagrams](#_System_Diagrams_1).

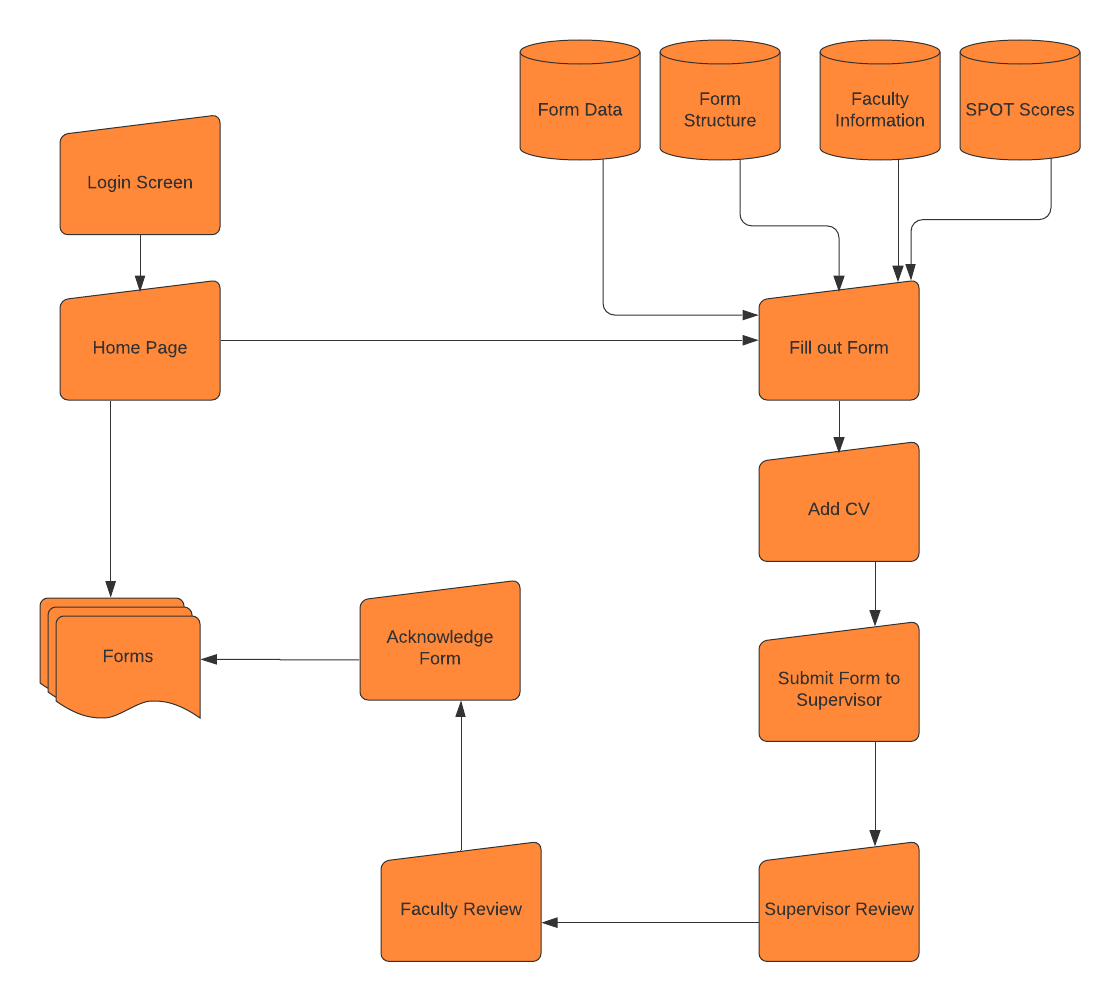
## Versioning

The core logic will utilize [Semantic Versioning](https://semver.org/) in order to update the libraries without breaking dependencies.

# Models

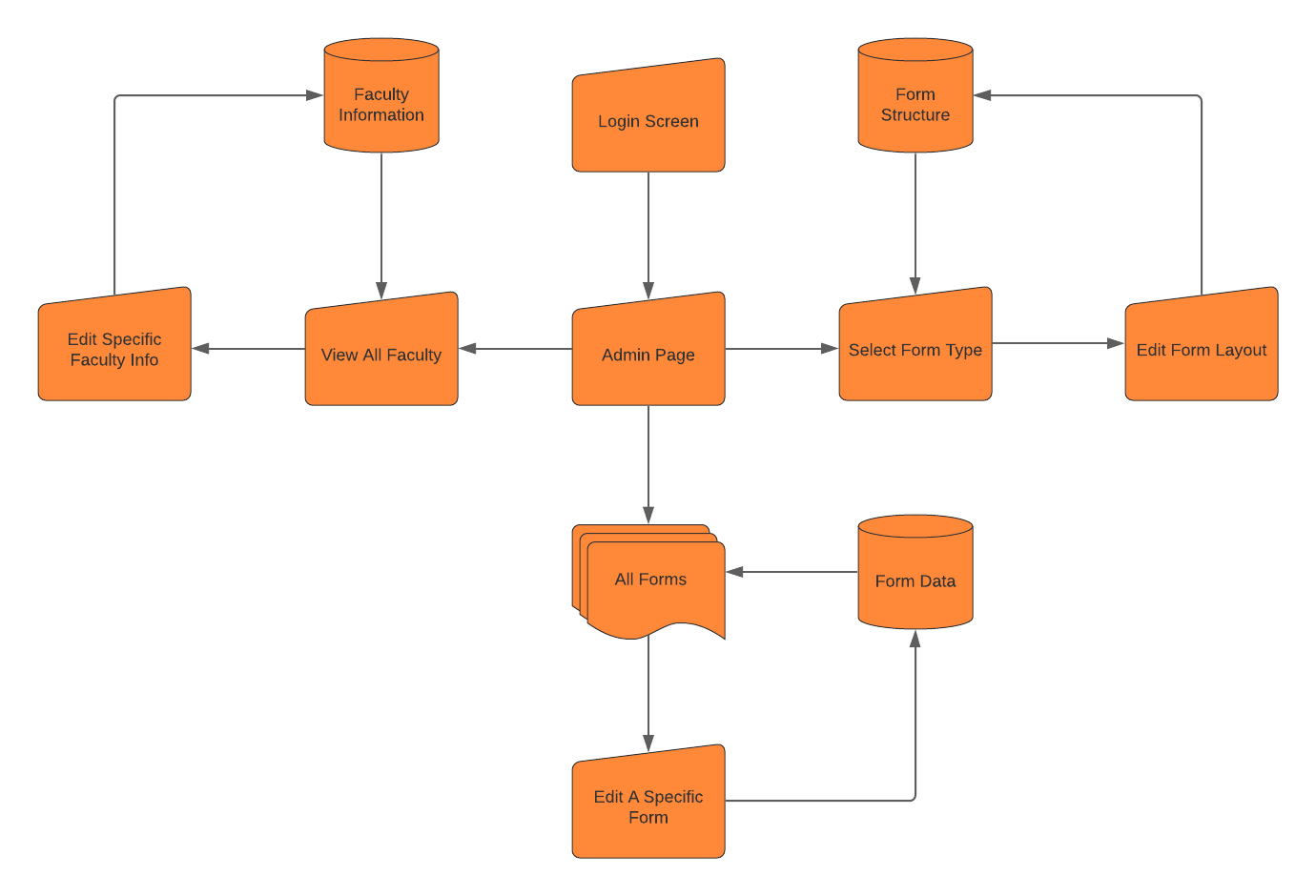
## Use Cases

### Faculty Workflow



### Dean / Faculty Chair Workflow

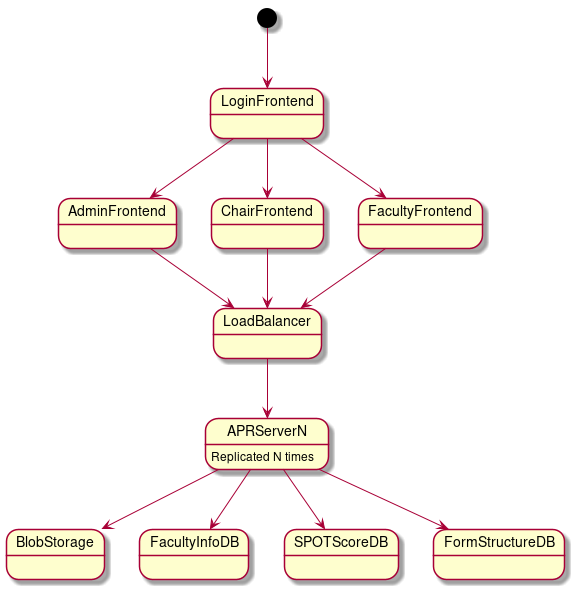
### Admin Workflow



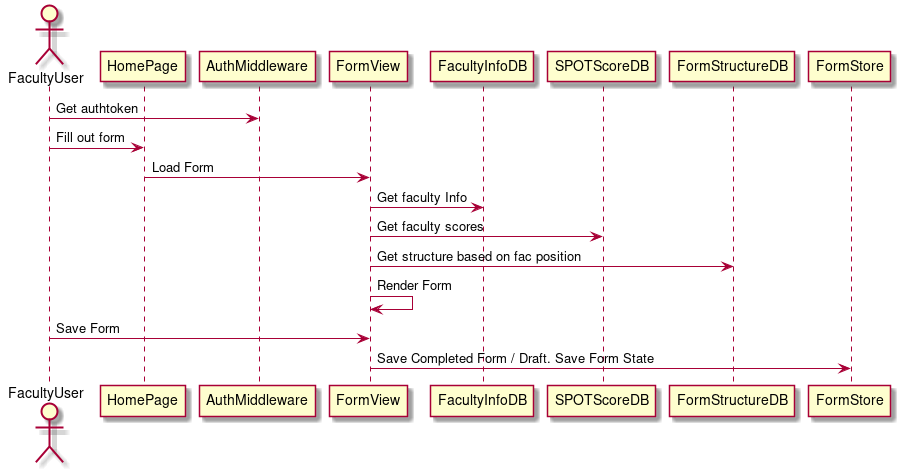
## System Diagrams

### System Diagram https://raw.githubusercontent.com/QuinnWass/CS-21.16/ArchDiagramDesc/docs/diagrams/arch_diagram.png?token=ADG5AR5NMCR7LVBWCWBBUDC7UDW4G

### System Diagram with Load Balancing



### Fill Out Form Sequence Diagram



### Dean or Chair Review a Faculty Form Sequence Diagram

A picture containing calendar

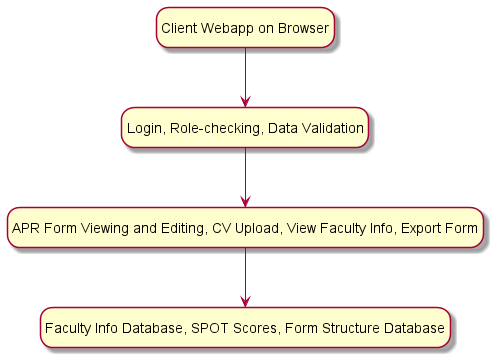
Description automatically generated

### Admin Login and Update Faculty Info Sequence Diagram

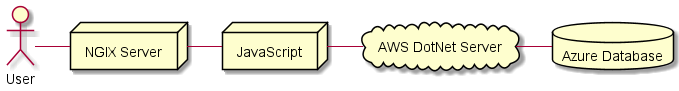
A picture containing diagram

Description automatically generated

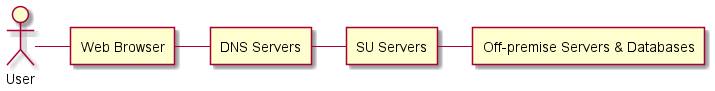
## Layer Architecture



## Deployment Diagram

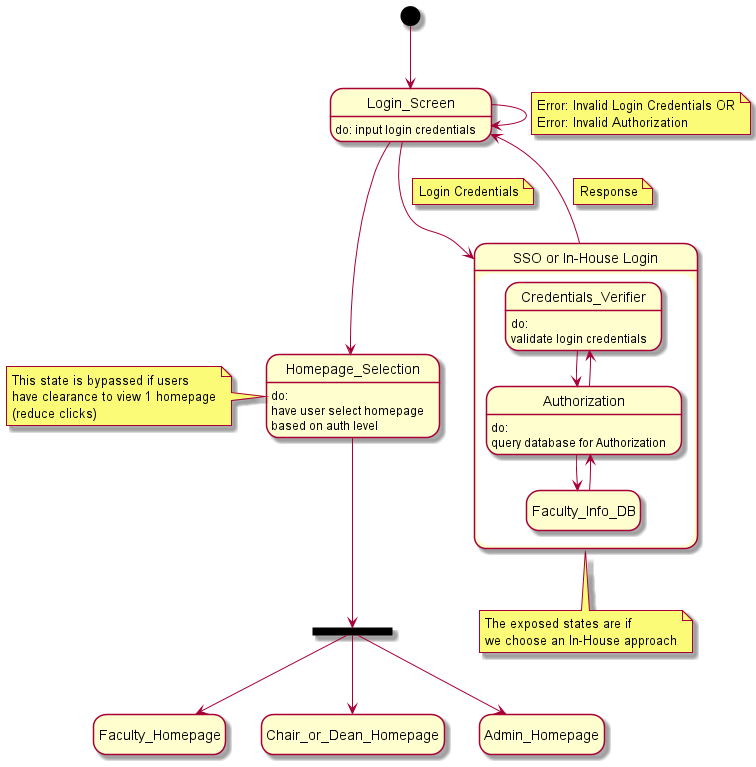


## Physical Diagram

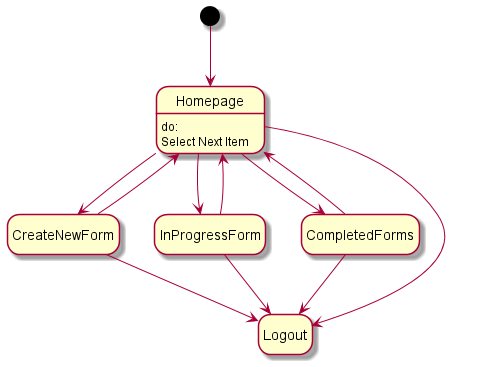


## State Diagrams

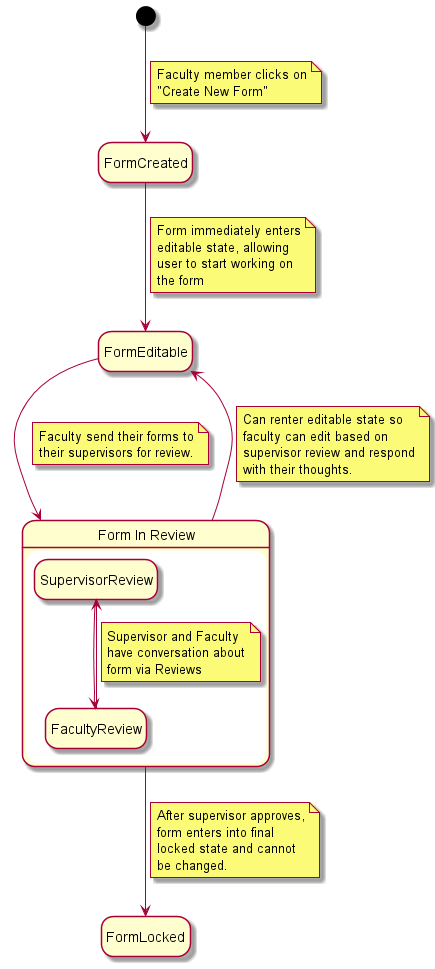
### Login



### Faculty Homepage



### APR Form



### The Dean or Chair Reviewing a Faculty Form

Diagram

Description automatically generated

### Admin Login and Update Faculty Info

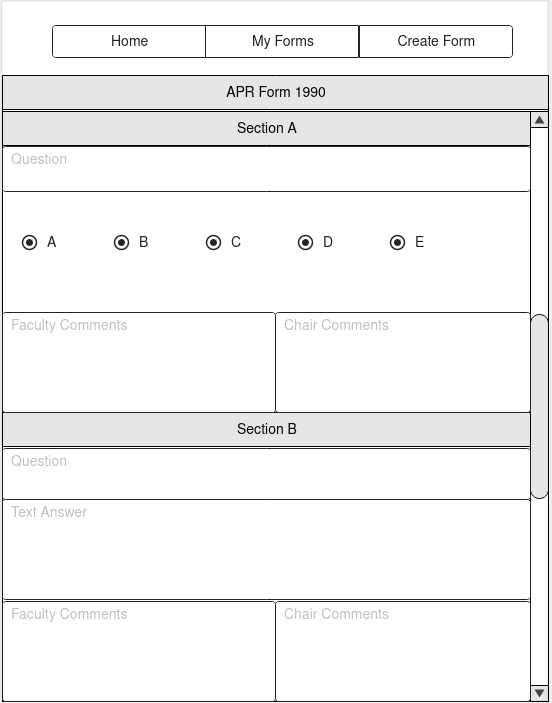
Diagram

Description automatically generated

## Wireframes

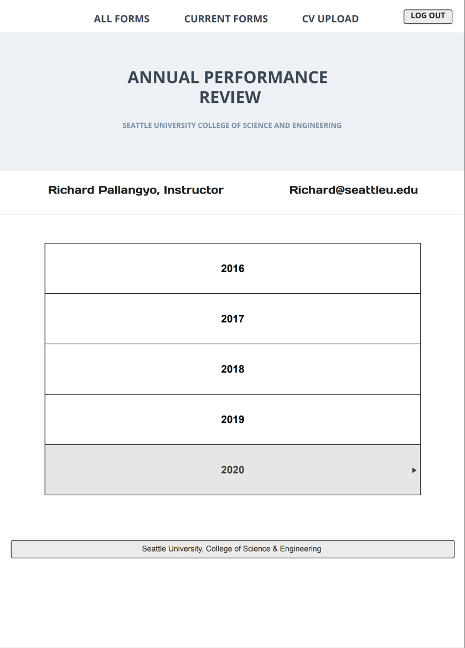
### Form Builder

This view demonstrates what filling out an APR form looks like. It aggregates all applicable sections and represents various input types depending on section.

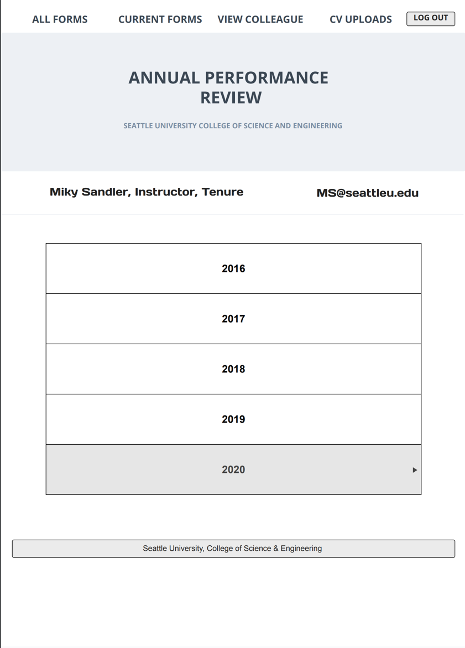


### Login Page

### Faculty Homepage



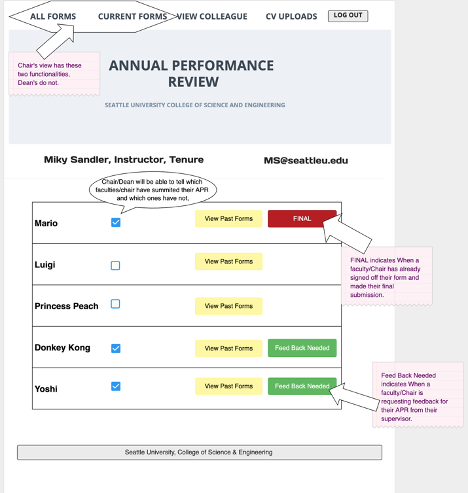
### Chair Homepage



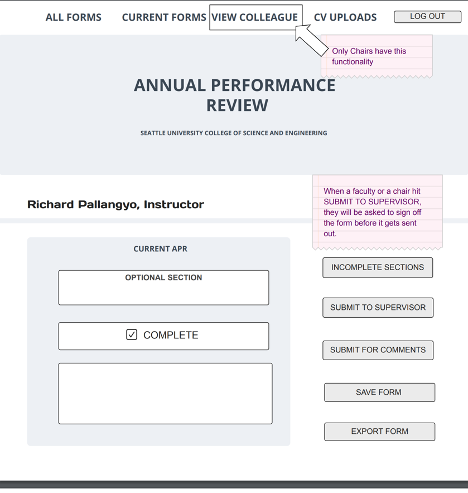
### Dean Homepage



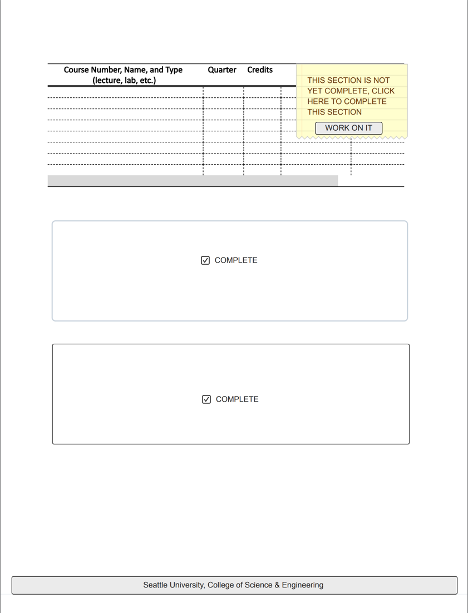
### View Colleague(Chair/Dean)



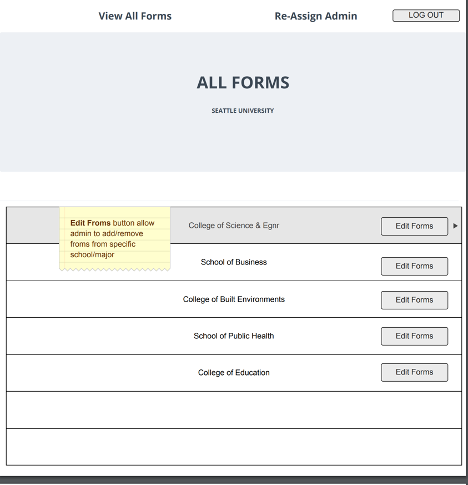
### Current Form (Faculty/Chair)



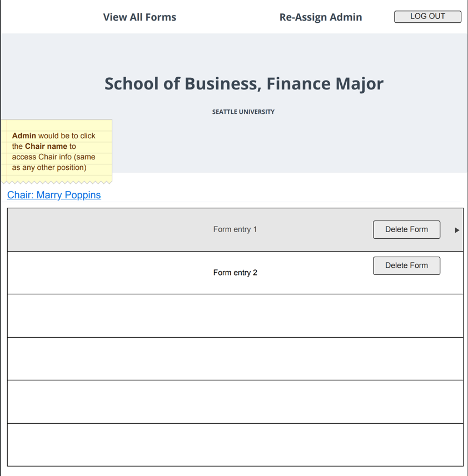
### Work in Progress(Faculty/Chair)



### Admin Homepage



### Individual Form Entries



# Version Control Table

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description of changes and person(s) responsible for changes** |
| 1.0 | 11/4/2020 | Initial contents for Design Doc finalized (Amanda Zheng, Richard Pallangyo, Malav Dipankar, Quinn Wass) |
| 1.0.1 | 11/8/2020 | Update the Requirements and NFR section. Quinn Wass |
| 1.0.2 | 11/9/2020 | Add Deployment and Physical Diagrams, update Physical View. Malav Dipankar |