System Architecture Document

Authors

Title of Project

October 2, 2018

Contents

1 Introduction

This introduction provides a brief overview of System Architecture Document for the current iteration of the Post Graduate Application Approval System. It consists of the purpose, scope, problem statement,project objectives, stakeholders and overview of the rest of the document.

1.1 Purpose

This document provides the reader with an architectural overview of the Post Graduate Application Approval System. The primary purpose of this project is to create a single integrated system that facilitates application review and decision making.

This document is intended to elucidate the major architectural decisions that have been made when designing and implementing the system. This is achieved by viewing the system architecture from various perspectives, called views. These views are intended to explain the system architecture through all levels of the development stack, from front-end to back-end.

1.2 Scope

The scope of this document is the design and implementation of the Post Graduate Application Approval System which consists of application upload by the Post Graduate Officer through to the final application decision made by the Post Graduate Coordinator.

1.3 Problem Statement

1.4 Project Objectives

1.5 Stakeholders

1.6 Overview

describe the structure and content of rest of report

2 Architectural Goals and Constraints

The architecture of the system has been designed to achieve the following objectives:

1. To assist the application approval process by having all application documents and information in a single digital repository.
2. To assist the Post Graduate Officer with the upload of application documents.
3. To simplify application reviews by having a single integrated view of all important information for each decision maker.

4. To prevent human error by providing notifications and weekly reminders about pending applications.

The significant constraints kept in mind when developing the system were as follows:

1. Security
2. Ease of Use
3. Paperless

3 Architectural Representation

3.1 Architectural Views

The development of the system has various contributors each with their own priorities and tasks. As such, the system needs to be documented from various perspectives to aid, and eventually validate, the completion of a contributor's tasks. The system architecture shall be represented from the following views:

1. Use Case View: This defines the high-level interactions between various actors and the system.
2. Design View: This contains the class and architecture diagrams of the system.
3. Process View: This displays the processes within the system that combine to perform the various interactions defined in the Use Case View.
4. Component View: This displays the User Interface of the system.
5. Database View: This contains the Entity-Relationship Diagram for the system database.

3.2 Architectural Design Patterns

ASP.NET Core framework was used in the implementation of the system. This follows the Model-View-Controller(MVC) design pattern. This design pattern separates the project into three distinct layers:

1. Model: Defines the data structures of the system and directly handles all logic and data within the system. A model class communicates exclusively with its controller.

2. View: A visual representation of a model. Typically in the form of a web page or a component of the web page. The view communicates exclusively with its controller.

3. Controller: Accepts user input and maps it to instructions for models, views and potentially other controllers. Directly responsible for communication between components of the system.

This framework and design pattern was chosen to enhance modularity of the system. This allows for:

- Parallel development
- Efficient code reuse
- Faster bug detection and tracking.
- Greater unit testing coverage.

3.3 Architectural Process

what process was used to design the system.

4 Architectural View Decomposition

4.1 Use-Case View

use case diagram plus short description of each use case.

4.1.1 Architecturally Significant Use Cases

describe in detail the use cases that use the most critical part of the system (possibly fully dressed use cases for this)

4.2 Design View

architecture diagram, class view

4.2.1 Overview

4.3 Process View

Activity diagram, ssds of significant use cases etc.

4.4 Component View

UI organization and organization of overall system.

4.4.1 Overview

4.5 Database View

ERD

5 Size and Performance

any metrics for the size and performance of the current system go here.

6 Quality

issues with system quality or concerns for future development go here.