Deadass Default

Project Plan

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/9/2021 | 1.0 | Final release of project | Deadass |

Table of Contents

1. Introduction 4

1.1 Purpose of this document 4

1.2 Intended Audience 4

1.3 Scope 4

1.4 Definitions and acronyms 4

1.4.1 Definitions 4

1.4.2 Acronyms and abbreviations 5

1.5 References 5

2. Background and Objectives 5

3. Architecture and High-Level Design 5

4. Organization 5

4.1 Project group 5

4.2 Customer 6

5. Development process 6

6. Deliverables 6

7. Project risks 6

8. Communication 7

8.1 Collaboration 7

8.2 Git 7

9. Project plan 7

9.1 Time schedule 7

9.1.1 Remarks 7

9.2 Test plan 8

9.2.1 Testing Remarks 8

# Introduction

## Purpose of this document

The purpose of this document is to provide a detailed project description of the application called Deadass Default, which is designed to provide visualizations of enterprise data. This document includes details about organization, roles, deliverables, project risks, time plans and financial plans.

## Intended Audience

This document shall be used in all phases of the project as a guideline. Intended audiences of this project are all project stakeholders:

* Andrew Bond: Professor that this project is to be delivered to
* TAs: all teaching assistants assigned to this course
* Team members: the primary developers of the project
* Project Supervisor and Leader: Paul Nguyen

## Scope

This document defines the project plan of the Deadass Default application. The overview includes objectives of the project, organization of the project team, development process that is going to be used during the project, assessment of possible risks, communication used between project stakeholders and project plan that includes time schedule and activity plan.

## Definitions and acronyms

### Definitions

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
| Deadass Default | The name of the project |
| Project Supervisor | A person in charge of supervising the project |
| Project Leader | A person in charge of organizing the team and communicating with the project supervisor |
| Team Member | An active member of the team responsible for making the job done |
| Milestone | A time in a project that marks the end of a project phase or the completion of an important deliverable. |
| Git | Version control system that will be used in this project |
| Scrum | An iterative and incremental agile software development method for managing software projects and product or application development |
| AWS | A collection of services utilized in development |
| Scrum sprint | The basic unit of development in Scrum |
| Scrum master | Ensures the smooth working of the Scrum team and enforces Scrum practices |
| Product owner | Responsible for product management and its quality |

### Acronyms and abbreviations

|  |  |
| --- | --- |
| **Acronym or**  **abbreviation** | **Definitions** |
| AWS | Amazon Web Services |
| CI/CD | Continuous Integration/Continuous Deployment |
| SSO | Single Sign On |
|  |  |
|  |  |

## References

1. <https://www.scrum.org/>
2. <https://github.com>
3. <https://aws.amazon.com>

# Background and Objectives

The purpose of this project is to demonstrate understanding of enterprise software development tools and practices with the default project proposal provided by Andrew Bond for Software Engineering 272: Enterprise Software Platforms. As described, the proof-of-concept application will comprise of a web application to display a demo of a corporate database that includes Single Sign-On authentication, a code repository, a Jenkins integration to automate deployments, and additional tools that have yet to be considered. The flask app serves subsets of data extracted from the database provided for this project and is rendered for the user with D3 and jQuery for summary graphics

# Architecture & High-Level Design

The app consists of a few basic features. A login/register page that allows single sign on using Shibboleth via a Flask extension Flask-SSO and Flask’s built-in session module, a basic set of tables constructed with the DataTable jQuery plugin that displays company-wide data group by department, and an assortment of D3 visualizations. The tables can display from a selection of 10, 25, 50, and 100 entries per page, is sortable by all the indexed columns, and shows an alert window when displaying series information like an individual employee’s salary or department history. The gender visualization consists of 2 bar charts displaying the gender distribution by department and overall, a scalable scatter plot displaying the age distribution of employees, and a circle plot that shows the frequency of different salary values.

# Organization

## Project group

|  |  |  |
| --- | --- | --- |
| **Name** | **Initials** | **Responsibility (roles)** |
| Paul Nguyen | PN | Project Lead, Project Supervisor, Developer, Product Owner |
| Austin Wilson | AW | Developer |
| Sharad Nataraj | SN | Developer |

## Customer

The target customers are listed below:

* Andrew Bond
* TAs assigned to CMPE 272

# Development process

This project currently utilizes Flask, a lightweight Python microservice library to use as a webserver as the bulk of the project. To host our data, our database that we use is MySQL and the project is hosted on AWS Elastic Beanstalk using Amazon’s CodeDeploy to automate deployment and handle CI/CD via webhooks from GitHub which are automatically generated when commits are pushed to the master branch of the repository.

# Deliverables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **To** | **Output** | **Planned week** | **Promised week** | **Late +/-** | **Delivered week** | **Notes** |
| **Andrew Bond** | **Source code, demo, report** | **12/9** | **12/9** |  | **12/9** |  |
|  |  |  |  |  |  |  |

# Project risks

|  |  |  |
| --- | --- | --- |
| **Possibility** | **Risk** | **Preventive action** |
| Data processing speed | Med | Convert data type into a lighter format that requires less processing |
| Load Balancing | Low | Low audiences means that this app does not require much load balancing and can automatically be configured via AWS |
| Readability of data | High | Primary objective is to display the data in an attractive format |

# Communication

## Collaboration

For communication, all members are present in a discord server for general discussion of the class and project plans. Additionally, a webhook to GitHub is made in a separate channel as a notification system so everyone knows who and when changes were made regardless if email notifications were set up.

## Git

All source code and finished documentation will be uploaded to GitHub repository.

Repository URL: <https://github.com/paul1409/Deadass>

# Project plan

## Time schedule

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Milestone**  **Description** | **Responsible Dept./Initials** | **Finished week** |  |  |  | **Metr.** | **Rem.** |
|  |  |  | **Plan** | **Forecast** |  | **Actual** |  |  |
|  |  |  |  | **Week** | **+/-** |  |  |  |
| 01 | Complete Project | Devs | 12/9 | 12/9 |  |  |  |  |
| 02 | Complete Report | Devs | 12/9 | 12/9 |  |  |  |  |
| 03 | Complete Presentation | Devs | 12/2 | 12/2 | 0 | 12/2 |  |  |
| 04 | Set up CI/CD | Devs | 11/7 | 11/7 | 0 | 11/7 |  |  |

### Remarks

|  |  |
| --- | --- |
| **Remark Id** | **Description** |
| 01 | Project completed close enough to promised delivery date |

## Test plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | 001 | Phase: | 1 | Author: | PN | Date: 12/9 |
| Test Category: | | **Performance** | | |  |  |
| Software Product: | | Deadass Default | | | |  |
| Test Title: | | Analyse performance of loading dataset | | | | |
| Test Purpose: | | Determine what optimizations are required for site performance | | | | |
| Test Setup: | | Load datasets in the backend and have frontend access | | | | |
| Prerequisites: | | Datasets from Bond for default project purpose | | | | |
| Procedure: | | Load up site and wait for any table to load | | | | |
| Checks: | | Speed | | | | |
| Expected Results: | | <5s load time | | | | |
| Result: | | Loading inconsistent for dataset | | | | |
| Reason for Failure: | | Reading from unformatted text files is not optimal | | | | |
| Remarks: | | Discover different methods to load dataset | | | | |

### Testing Remarks

|  |  |
| --- | --- |
| **Remark Id** | **Description** |
| 01 | Text files were used to help create the front end and good enough for development usage |
|  |  |
|  |  |
|  |  |