**Requirements Specification**

**for Capstone:**

**Prepared by**

**Shadarien Williams**

# COP-2939-93453

# Instructor: Dr. Banisakher

**Date: September 11, 2024**

***Software Requirements Specification Page ii***

Table of Contents

[1. Introduction 1](#_Toc176938087)

[1.1 Purpose 1](#_Toc176938088)

[1.2 Intended Audience 1](#_Toc176938089)

[This document is intended for the development team, testers, system administrators, and instructors involved in reviewing the software project. 1](#_Toc176938090)

[1.3 References ( this is just example only) 1](#_Toc176938091)

[2. Overall Description 1](#_Toc176938092)

[2.1 Product Overview 1](#_Toc176938093)

[2.2 User Classes and Actors 1](#_Toc176938094)

[2.3 Program Environment 1](#_Toc176938095)

[2.4 Assumptions and Dependencies 2](#_Toc176938096)

[3. Functional Requirements 2](#_Toc176938097)

[3.1 Use Cases 2](#_Toc176938098)

[4. Domain Requirements 4](#_Toc176938099)

[5. Nonfunctional Requirements 4](#_Toc176938100)

[4.2 Accessibility and Usability Requirements 4](#_Toc176938101)

[5. Future Requirements 4](#_Toc176938102)

[Appendix A: Potential Changes and Constraints 5](#_Toc176938103)

[Appendix B: Example User Interface 5](#_Toc176938104)

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to detail the requirements for a Task Management System, which enables users to create, assign, and manage tasks efficiently. The system is built using Django and is designed to cater to individual users, managers, and administrators.

## 1.2 Intended Audience

## This document is intended for the development team, testers, system administrators, and instructors involved in reviewing the software project.

## 1.3 References ( this is just example only)

(<https://docs.djangoproject.com/en/5.1/>), Django Documentation

(<https://nishothan-17.medium.com/pbkdf2-hashing-algorithm-841d5cc9178d>), PBKDF2 password, (<https://developer.mozilla.org/en-US/docs/Learn>) HTML Documentation.

# 2. Overall Description

## 2.1 Product Overview

The Task Management System is a web-based application where users can create tasks, assign them to others, track progress, and manage deadlines. The system features user roles, task assignment, task filtering, and notification settings.

## 2.2 User Classes and Actors

* **Admin**: Manages users, roles, and overall system settings.
* **Manager:** Creates and assigns tasks to users, views reports.
* **User:** Receives tasks, tracks completion, and updates status.

## 2.3 Program Environment

The system is built on Django (Python-based web framework) and will be deployed on a web server. It requires a modern web browser (Chrome, Firefox, etc.).

It is assumed that *Functional Requirements*

## 2.4 Assumptions and Dependencies

Below is a set of normal use cases, their alternative flows, and screen descriptions from the user perspective when applicable.

# 3. Functional Requirements

## 3.1 Use Cases

**UC 1**

User Registration

Users can register for an account by providing their email, password, and other details.

A user (manager or admin) can create a new task, set a due date, and assign it to another user.

**UC 2**

Task Creation

**UC 3**

Managers and admins can assign tasks to users and manage assignments.

Task Assignment

**UC 4**

Users can view their assigned tasks and track their progress.

Task Tracking

**UC 5**

Task Editing

Users can edit or delete tasks based on their roles (admin/manager)

Admins can assign roles (Admin, Manager, User) to registered users.

**UC 6**

Role Management

# 4. Domain Requirements

These are the requirements imposed by the domain of the program*.*

* Tasks can only be assigned to registered users.
* User data must be secure and comply with password security best practices (PBKDF2).
* Admins have full control over the system, including task creation, assignment, and user roles.

# 5. Nonfunctional Requirements

**5.1 System and Platform Requirements**

* The system must operate on all major web browsers.
* It will be hosted on a Linux or Windows server with Python and Django installed.

## 5.2 Accessibility and Usability Requirements

The window may be clicked and dragged

**5.3 Safety and Security Requirements**

The program will not collect any personal.

**5.4 Maintenance and Reliability Requirements**

The chosen frameworks and APIs will be with future Windows operating systems and devices.

# 6. Future Requirements

Potential features may be added in the future. These include may change complex design, features are added.

# Appendix A: Potential Changes and Constraints

* Changes may be required to accommodate more complex user roles and task assignment workflows.
* Constraints include potential limitations of web hosting server or issues related to scaling.

# Appendix B: Example User Interface

Below is an initial layout of the home screen, which appears upon the opening of the program, or the click of the Home button.

* Main Page: Users can see their assigned tasks with a “Create new Task button”

A screenshot of a computer

Description automatically generated

* Task Creation Page: A form to input task details such as a title, description, due date, and assignee.

A screenshot of a computer

Description automatically generated

* User Registration: Enables new users to register at the time of login.

A screenshot of a computer

Description automatically generated

* Login Page: Allows users to authenticate.

A white background with black text

Description automatically generated