Statement of Work

Nomoth Project Management System
Date: February 17th, 2021
Drafted by: Keeth Spindler

Introduction

When involved in any collaborative environment, communication and organization are keys to success. Without these crucial skills, more work will inevitably be created from correcting recurring problems. The Nomoth Project Management System (The system) aims to mitigate these obstacles. By offering users a way to manage their projects and the tasks they commit to, this system allows teams of any size to stay organized and increase their work efficiency. With the additional capability of creating thorough defect tickets, developers will have a well-rounded tool which will increase the quality of their projects.

System roles

The system will house three types of user roles: Admin, Project Manager (PM), and Developer. Each role has a descending level of authorization respectively and has a specific range of functionality. Admins are the Superusers of the system, whereas PMs have the second highest level of authorization. developers have the lowest authorization and will have the fewest functionality workflows.

Scope of Work

This document identifies the required system capabilities at a high level. Later documents will specify the detailed requirements. These capabilities are required:

1. **The system must have role-driven functionality**

   Specialized functionality will be provided to users which is dependent on their role and authorization. The following is a list of the roles and their accompanying functionalities:
   - Admin
     - Register and remove users
     - Assign roles to users
     - Assign PMs and developers to projects
     - Create, modify and delete projects
     - Create, modify and delete tasks

    - Assign tasks to developers
    - Create, modify and delete tickets
    - Assign tickets to developers
- PM
    - Assign developers to projects
    - Create, modify and delete tasks
    - Assign tasks to developers
    - Create, modify and delete tickets
    - Assign tickets to developers
- Dev
    - Create and modify tasks
    - Create and modify tickets

2. **users must be register through email**

   In order to use the system, a User must be registered through an account using their email and a password.

3. **A User cannot have more than one role**

   In order to minimize an authorization security breach, each User will only be granted one role.

4. **Each Project must contain Agile development boards to contain their tasks**

   For the organization of Project tasks, all projects will contain an Agile development board with the following swim lanes:
   - Backlog
   - In Progress
   - Testing
   - Review
   - Complete

5. **Diagrams of a Project's ticket statistics must be displayed**

   Statistics about a Project's tickets will be displayed in detail on its Dashboard, such as the frequency of tickets created, their categories, and the frequency of ticket resolutions.

6. **A task must display information about itself**

   When viewing a task's description, the user should see the task's title, description, the date it was created, its user story's ID, the user story description and the user who is committed to implementing it.

7. **A task must list all of the tickets associated with it**

   When viewing the details of a task, a list of all tickets linked to it, whether they are opened or closed, will be displayed as historical data.

8. **A ticket must display information about itself**

   When viewing a ticket's description, the user should see the ticket's title, description, the date it was created, its category type, priority level, current status, the user who created it and the user who is assigned to resolving it.

9. **A ticket must save any changes made to it**

   A ticket's history will be preserved and displayed when viewing a ticket's details.

10. **A user must be able to comment on their project's tickets**

    When viewing a ticket within a project a user is assigned to, they will have the ability to comment on it and view others who have commented as well.

Expected System Technologies

The following structure is proposed for the construction and implementation of the system:

- API
  - .NET 5
- Database
  - SQLite
    - Version 3.34.1
- ORM
  - Entity Framework Core 5.0
- Client-Side
  - React
    - Version 17.0.1
  - Redux
    - Version 4.0.5

These technologies were chosen due to their popularity within the industry and light-weight infrastructure. Modification to this stack may occur, including additional technologies, the substitution of a technology, or version updates.

Protect Timeline

This project will be broken down into 4 phases made up of 2-week sprints. The following is an estimated structure that will be followed in order to accomplish the project:

Analysis Phase (10 Weeks):

- Sprint 1, 2
  - Statement of Work
  - Activity Diagram v1.0
- Sprint 3,4
  - Use-Case Diagram v1.0
  - System Specifications Document v1.0
  - Data-Model v1.0
- Sprint 5
  - Sprint 2 & 3 Work Review
  - User Story Matrix

Design Phase (10 Weeks):

- Sprint 6,7
  - Wireframe Construction
  - Wireframe Review
- Sprint 8,9
  - Prototype Construction
- Sprint 10
  - Prototype Revision

Management Review

Build Phase (TBD)

Implementation Phase (TBD)

Signature

I, the undersigned, hereby state that I understand the requirements given within this Statement of Work and agree to fulfill them to the best of my ability.

Developer: _____ Date: _____