**A picture containing text

Description automatically generated**

**Department of Informatics**

**University of Leicester**

**CO7201 Individual Project**

**Preliminary Report**

**Project Management System**

**Deepthi Vangapandu**

**[Your email address]**

**[Your student ID]**

**Project Supervisor: [XXXX]**

**Second Marker: [XXXX]**

**Word Count: [XXXX]**

**[Submission Date]**

**DECLARATION**

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people’s work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people’s work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: [Your name]

Date:[Date of submission]

Contents

[**1: Aims and Objectives** 3](#_Toc107136400)

[**2: Requirements** 3](#_Toc107136401)

[2.1: Essential 3](#_Toc107136402)

[2.2: Recommended 4](#_Toc107136403)

[2.3: Optional 4](#_Toc107136404)

[**3: Technical Specification** 4](#_Toc107136405)

[**4: Requirements Evaluation Plan** 4](#_Toc107136406)

[**5: Background Research and Reading list** 4](#_Toc107136407)

[**6: Time-plan and Risk Plan** 4](#_Toc107136408)

[**7: References** 4](#_Toc107136409)

# **1: Aims and Objectives**

Software is an essential part of life in the modern world. A lot of people work in the field of software development striving to solve problems or make work easier. This has led to existence of complex/sophisticated pieces of software which pose a challenge of understanding how they work. Software developers often move from projects to projects or from companies to companies and are required to get in a team and continue working. Having no prior knowledge of the existing project, new developers have to spend a lot of time and resources trying to wrap their head around the project for them to be able to contribute to the project. The learning period is often lengthened by the scattered sources of information like UML diagrams, ER diagrams and project documentations. Developers have to read through these different materials and consult with colleagues in order to get to a place of understanding the project.

To solve this kind of tedious work, there is a need for a platform where a project’s information can be bundled together to present a more finished product that presents all needed information about a project. This way a developer could get up to speed by viewing items like database tables and how they relate to each other, API endpoints and the description of what they do, Classes/Components used in code for both frontend and backend and any notes about the project written by team members.

This project is aimed at satisfying the need for such a platform and adds another functionality of team management utilities like organizing meetings, live chat and project progress management.

The success of this project solving all problems faces some challenges. There is a lot of information involved in software projects and although it would be very useful for all of it to be represented, not all can be accommodated to the project. Instead, the project focuses on the main pieces of information that form the foundation of the knowledge about a project. Despite focusing on main aspects of a project, sometimes those aspects can be complicated in nature. For example, there may be complex database tables relationships, and this poses a challenge on how to add and show them in the system.

# **2: Requirements**

## 2.1: Essential

1. Create, edit, delete users (admin, project manager, team members)
2. Create, edit, delete a project and Project information
3. Record database information, API endpoints, classes/components used in code for both frontend and backed
4. Add notes about the project organized in topics and subtopics
5. Present project’s information like project progress, team members, budget, technologies
6. Filter the project based on budget, team size, technology
7. Email functionality to email team members, setup meetings.
8. Define roles on what operations different types of users can do in the project

## 2.2: Recommended

1. Email Alert notifications and remainders to the meetings
2. Present the progress of the project
3. To allow chats among team members (live chat)

## 2.3: Optional

1. Present project information using diagrams drawn from entered data

# **3: Technical Specification**

This project will be implemented as a web application. The frontend will be done using ReactJs. The API will be done using SpringBoot and will utilise Microsoft SQL Server as the database. The programming/markup/styling languages involved will be Java, JavaScript, HTML and SASS.

# **4: Requirements Evaluation Plan**

Upon implementation, the project will be put under test to ascertain that it works as expected. The tests will involve all the personnel they will demand which may include me (the project implementor), users (friends, colleagues, lecturers) among others.

The testing will be done in the following categories

1. **End to end testing** 🡪 These tests will be done to monitor the workflow of the application to make sure everything works as expected
2. **Functional testing 🡪** these tests will be used to evaluate whether the system will be doing what it is supposed to be doing
3. **Integration testing 🡪** these tests will be used to evaluate whether the system meets its requirements
4. **Performance testing 🡪**  to evaluate the system’s speed, scalability, reliability and stability
5. **Security testing 🡪** to ensure there are no flaws that expose the system to risks of attack

# **5: Background Research and Reading list**

Background material (including a reading list and literature review where appropriate);

# **6: Time-plan and Risk Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Duration | Start | Finish |
|
| 1 | Database design | 4 days? | 6/14/22 8:00 AM | 6/17/22 5:00 PM |
| 2 | Database implementation | 5 days? | 6/20/22 8:00 AM | 6/24/22 5:00 PM |
| 3 | API design | 5 days? | 6/27/22 8:00 AM | 7/1/22 5:00 PM |
| 4 | API implementation | 10 days? | 7/4/22 8:00 AM | 7/15/22 5:00 PM |
| 5 | API Unit testing | 5 days? | 7/18/22 8:00 AM | 7/22/22 5:00 PM |
| 6 | Frontend design | 5 days? | 7/25/22 8:00 AM | 7/29/22 5:00 PM |
| 7 | Frontend implementation | 10 days? | 8/1/22 8:00 AM | 8/12/22 5:00 PM |
| 8 | Frontend Unit testing | 5 days? | 8/15/22 8:00 AM | 8/19/22 5:00 PM |
| 9 | End to end testing | 5 days? | 8/22/22 8:00 AM | 8/26/22 5:00 PM |
| 10 | Functional testting | 3 days? | 8/29/22 8:00 AM | 8/31/22 5:00 PM |
| 11 | Integration testing | 2 days? | 9/1/22 8:00 AM | 9/2/22 5:00 PM |
| 12 | Performance testing | 1 day? | 9/5/22 8:00 AM | 9/5/22 5:00 PM |
| 13 | Security testing | 1 day? | 9/6/22 8:00 AM | 9/6/22 5:00 PM |

A picture containing calendar

Description automatically generated

# **7: References**

The reference list should contain a mixture of books, research papers (if appropriate) and internet resources, and should not consist only (or mainly) of Internet resources;