Tribhuvan University

Texas International College
A Final Year Project Report
On
AceTask

Under the Supervision of
Mr. Omkar Basnet
Academic Incharge CSIT/ BCA
Texas International College

Submitted To

Department of Computer Science & Information Technology
Texas International College

In partial fulfillment of the requirement for the bachelor's degree in Computer Science & Information Technology

Submitted By

Arjun Gautam (15566/074)
Kushal Giri (15579/074)
Sandip Gole (15596/074)

April 2022

SUPERVISOR'S RECOMMENDATION

I hereby recommend that the report prepared under my supervision by Arjun Gautam (TU Exam Roll No. 15566/074), Kushal Giri (TU Exam Roll No. (15579/074), Sandip Gole (TU Exam Roll No. 15596/074) entitled "**ACETASK**" in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

Mr. Omkar Basnet
Academic Incharge,
Department of CSIT/BCA
Texas International College
Chabahil, Kathmandu

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by ARJUN GAUTAM (TU Exam Roll No. 15566/074), SANDIP GOLE (TU Exam Roll No. 15596/074) and KUSHAL GIRI (TU Exam Roll No. (15579/074) entitled "**ACETASK"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Mr. Omkar Basnet
Project Supervisor
Academic Incharge, Department of CSIT/BCA
Texas International College
Chabahil, Kathmandu

Mr. Kumar Poudyal
HOD, Department of CSIT
Texas International College
Chabahil, Kathmandu

Mr. Sarbin Sayami
External Supervisor
Asst. Prof.
CDCSIT, Tribhuvan University
Kirtipur, Nepal

ACKNOWLEDGEMENT

The completion of the project depended on the teamwork and dedication of our team members. However, it would not have been possible without the kind support and help of various factors and individuals. Without the kind support and assistance of many individuals and organizations, the success of this project would not have been possible, and we are incredibly grateful to have achieved this during the lifetime of our project.

We would like to thank Texas International College for providing us this opportunity to explore our ideas and providing us the necessary working space that helped us to grasp the various functional as well as nonfunctional aspects of this project.

We would like to express our special thanks and gratitude to our supervisor (Mr. Omkar Basnet) for his continuous support and motivation. He stood beside us during this entire time with continuous inspiration and guidance. His direction and support were crucial for driving us towards the completion of the project.

Our sincere gratitude to the Head of department, Texas International College, Mr. Kumar Poudyal who is the real actor behind the completion of projects for all of us. Has provided us with all the technical and non-technical support and educators of many fields regarding the IT sector. To have someone so dedicated to the well-being of his students is quite an experience.

We are grateful and blessed enough to receive consistent support from our friends, families and every B.Sc. teaching staff. Department CSIT which helped us to complete our project successfully. We would like to extend our respects also to all B.Sc.'s non-teaching staff, Department of CSIT, for their timely assistance.

Sincerely,
Arjun Gautam (15566/074)
Kushal Giri (15579/074)
Sandip Gole (15596/074)

ABSTRACT

Ace Task is a web application built for the management of projects. It can be used to keep track of the project, add tasks for the team, keep track of the project progress and flow. Basically, it is focused on the small growing as well as the independent developers working on the various projects at a time.

Our main objective is to provide the facility of the project management tool with the simple interface and provide the premium service free of cost.

We used various sources for the data collection, with the study of the various existing products analyzing them to build our product. We have asked people about their view for this product and what change they would like if they would be using our product through the google form medium. The Internet played a major part for this as most of the research was done with the help of the internet.

Various tools and technologies are used for the development of the application. The software model is based on the Agile model. As for the tools, we have used software like Git, draw.io, MySQL workbench, Spring Boot, React.

The product can be made more advanced in the future with the growing number of users and feedback we get from our users. With the successful implementation of our web application the mobile version of our product is our next phase planning of this project which would give simple task as reminder and checking of the progress along with other features.

Keywords:
MySQL Workbench, JVM, RDBMS, Spring Boot, React, Agile

TABLE OF CONTENT

CHAPTER 1 1
INTRODUCTION 1
1.1. Background 1
1.2. Problem Statement 1
1.3. Objectives 2
1.4. Scope and Limitation 2
1.5. Development Methodology 2
1.6. Report Organization 3
CHAPTER 2 4
BACKGROUND STUDY AND LITERATURE REVIEW 4
2.1. Background Study 4
2.2. Literature Review 5
CHAPTER 3 6
SYSTEM ANALYSIS 6
3.1. System Analysis 6
3.1.1. Requirement Analysis 6
3.1.2. Feasibility Analysis 12
3.1.3. Object Oriented Analysis 14
CHAPTER 4 18
SYSTEM DESIGN 18
4.1. Design 18
4.2. Algorithm Details 20
CHAPTER 5 22
IMPLEMENTATION AND TESTING 22
5.1. Implementation 22
5.1.1. Tools Used 22
5.1.2. Implementation Details of Modules 23
5.2. Testing 25
5.1.2. Test Cases for Unit Testing 25
5.1.2. Test Cases for System Testing 30
5.3. Result Analysis 30

CHAPTER 6 31
CONCLUSION AND FUTURE RECOMMENDATIONS 31
6.1. Conclusion 31
6.2. Future Recommendations 31
REFERENCES 32
APPENDIX I 33
APPENDIX II 37

LIST OF FIGURES

Figure 1 : Kanban Board 3
Figure 2: Response for project management tools used 6
Figure 3: Response for ease of use in small projects 7
Figure 4: Response of Designation 7
Figure 5: Response for complexity of project management tools 7
Figure 6: Use Case Diagram of AceTask 8
Figure 7: Gantt Chart of AceTask 14
Figure 8: Class Diagram for AceTask 15
Figure 9: Sequence Diagram of ACETASK 16
Figure 10: Activity Diagram of AceTask 17
Figure 11 : Detailed Class Diagram of AceTask 18
Figure 12: Component Diagram of AceTask 19
Figure 13: Deployment Diagram of AceTask 20
Figure 14: Flow chart of Project Board Algorithm 21

LIST OF TABLES

Table 1: Table for functional Requirements of AceTask 9
Table 2: Test Case for User Registration 26
Table 3: Test Case for User Login 27
Table 4: Test Case for Project 28
Table 5: Test Case for Project Board 29
Table 6: Test Cases for System Testing 30

ABBREVIATIONS

API	Application Programming Interface
CSS	Cascading Style Sheet
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
JPA	Java Persistence API
JVM	Java Virtual Machine
JWT	JSON Web Token
ORM	Object Relational Mapping
RDBMS	Relational Database Management System
ROI	Return Of Investment
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1. Background

"AceTask" is a web-based software project management tool and task tracer used for managing personal projects. It can be used to keep the track of projects, and project tasks for management of projects within the specified time frame.

A software developer, freelancer can rely on this tool for managing a project to ensure that each task is completed on time and to balance workload for optimal time management. An individual can also be involved in various projects at a time, so to manage their time and ease the workload balance this tool can be used. This tool also helps to guide project managers or individuals to reach the specific goal set without wasting valuable resources such as time, money etc. As the modern workplace continues to evolve, so too must your project management skills. For any project to be successfully completed, better project management is essential. When you're involved in multiple projects at a time, it can be incredibly challenging to keep track of all individual deadlines. A project management software can provide you with easy timelines by keeping the priority order of specific tasks. This tool helps to visualize the project deadline report as well as

So AceTask focuses on breaking down the big project into smaller chunks of tasks which helps in the better and efficient workflow by keeping track of progress of the project. This tool is helpful for software developer as well as freelancer who are working on multiple projects at a time.

1.2. Problem Statement

An individual working on several projects at a time can have difficulty in managing multiple tasks at a time. For example, freelancers who are working on projects of multiple clients might feel difficulty in handling the tasks and project deadlines, in such cases this tool comes in place to solve the problem.

Software developer also sometime have problem of managing deadlines while working on multiple projects. By keeping the track of spent time and remaining time to complete the project we will solve the problem of time management.

1.3. Objectives

Following are the objectives of the project:

- To help users keep track of their project as well as project task for developers.
- To work on multiple projects concurrently.
- To make the project flow much more effectively and swiftly.

1.4. Scope and Limitation

A good management of the project can lead to a successful project and a good relationship with the client. A proper software management tool allows the user to manage, maintain their project, keep track of the project as well as project tasks including features and bugs.

Acetask helps to keep the track of multiple projects of individual developer at a time by keeping track of project as well as providing the information about the time to complete the project and time spent on project. Similarly for each project board will help to visualize the progress or project showing task on TODO, INPROGRESS and DONE section which will make the project flow more effective.

We have implemented the project for individual developers only. Role management feature can be implemented for better resource utilization. To improve user experience, drop and drag features in project board can be implemented.

1.5. Development Methodology

AceTask is developed using agile methodology specifically using Kanban methodology. We divided whole projects into small tasks. Each task is assigned to the Kanban board where the progress of the project can be visualized. We can have the TODO, IN PROGRESS and DONE section in Kanban board where tracking of the project becomes easy.

Kanban increases productivity by maintaining a consistent flow of tasks that culminate in the product. Kanban is a highly visual system. The production or development process is broken down into individual tasks which are each represented by a usually colorful and often color-coded card. We assigned various tasks to different members of the project and their progress is reflected in the Kanban board.

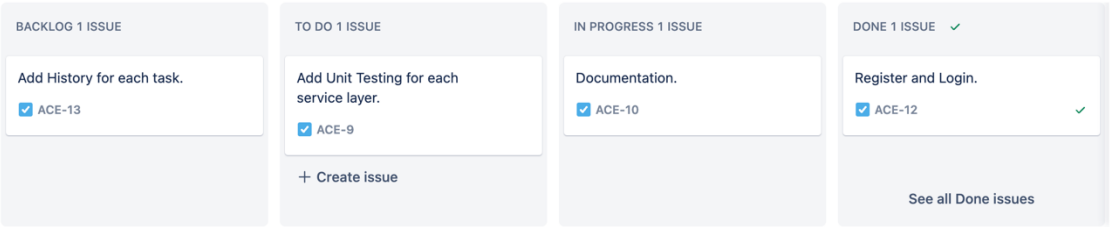


Figure 1 : Kanban Board

1.6. Report Organization

Chapter 1: Introduction
This chapter describes the project briefly with objectives, problem statement, scope, limitation, and development methodology.

Chapter 2: Background Study and Literature Review
The background section is the brief and descriptive first part of the report for the description of fundamental theories, general concepts, and terminologies related to the project. Literature review of the projects is used for review of similar or relevant projects, theories, and results by other researchers.

Chapter 3: System Analysis
This chapter is concerned with requirements and feasibility studies. Functional and nonfunctional requirements are explained and analyzed. Similarly, feasibility studies like technical, operational, economic, and schedule feasibility are analyzed.

Chapter 4: System Design
This chapter is concerned with the elaborated design of the model architecture, data used, and implementation process used throughout the project.

Chapter 5: Implementation and Testing
It includes software tools, dependencies and hardware tools used to implement the system. It also consists of different levels of tests carried out to test the model built for AceTask.

Chapter 6: Conclusion and Future Recommendations
This chapter includes the report's conclusion and the further work that can be done concerning the enhancement of the project.

CHAPTER 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1. Background Study

AceTask is a web application built for the management of projects. It can be used to keep track of the project, add tasks for each project, keep track of the project progress and flow. As the growing trend in the Information Technology sector there is an increase in the number of the personnel in this field. There are various indie developers working on their own projects, various startup companies which are operating on the low investment with the big dream of competing in the international market. We believe that with the birth of the new idea, passion and with AceTask no dreams are big. AceTask allows the user to login by their account, create tasks and assign among the members, list the task according to the priority of the task. Tasks can be searched according to its name or task id for ease.

In our country where this trend is still new and growing rapidly which is a good thing for sustainable manpower development, many small companies are opening. As a startup company, they can use AceTask easily for their project management with their small number of working teams and the easy interface of our product. It would be easier for them to manage their product and keep track of its progress. In this rapidly changing time, there is a huge requirement in the market, if you are not able to compete with this changing pace it is very difficult to run the business. As demands are changing quickly and pace is needed to be matched a good project management tool can be a game changer. Many independent developers are out there working on multiple projects at a time, AceTask can be the game changer for them as it is also focused for independent developers where they can set multiple projects and work simultaneously on them, making it easy for them to track their progress.

AceTask aims to provide equal opportunity to all as we provide our every feature for free to every user. We help to manage the project and task so that the workflow is not disturbed, and people can focus on the project rather than its management.

2.2. Literature Review

A good project management results in the proper development of the products. Among the first good examples of project management include the building and completion of the Great Pyramid of Giza in 2570 BC. [1] Even now, even the top archeologists are still arguing how the Egyptians were able to achieve this project. According to ancient records, managers were assigned for each of the Great Pyramid's four faces who were responsible for the coordination and completion of the task. In that sense, there was some form of planning, implementation, and control involved in finishing this project. Likewise, a good management of the project is a crucial part of the product development.

There are some of the existing software's like: JIRA [2], TRELLO [3], CLICKUP, ASANA, ZOHO PROJECTS and many more which have their own features and provide the services. But due to the complex UI and time it takes for learning these software's it can be admitted that these are a bit clunky and to be using this software daily is kind of frustrating. In most of this software there premium or all features are not available for everyone, they can only be accessed with the premium version of these which has to be paid and renewed in each time frame. From our software we will be providing these features of managing the software which can be feasible for big as well as the startup companies and indie developers for the quality services.

Jira software is an agile project management tool that supports any agile methodology, be it scrum, Kanban [4], or your own unique flavor. Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.

Trello is a collaboration tool that organizes your projects into boards. In one glance, Trello tells you what's being worked on, who's working on what, and where something is in a process. Imagine a white board, filled with lists of sticky notes, with each note as a task for you and your team. Now imagine that each of those sticky notes has photos, attachments from other data sources like Bitbucket [5] or Salesforce, documents, and a place to comment and collaborate with your teammates.

CHAPTER 3

SYSTEM ANALYSIS

3.1. System Analysis

3.1.1. Requirement Analysis

Data Collection

For the data collection of requirements and analysis of this project we did an online survey. We conducted an online survey as a part of a research tool for data collection. We prepared google form to collect data where questions were asked with different individuals to gather information about them, their experience and their thoughts about the services given by the existing system. Then we found the following data as a result of survey.

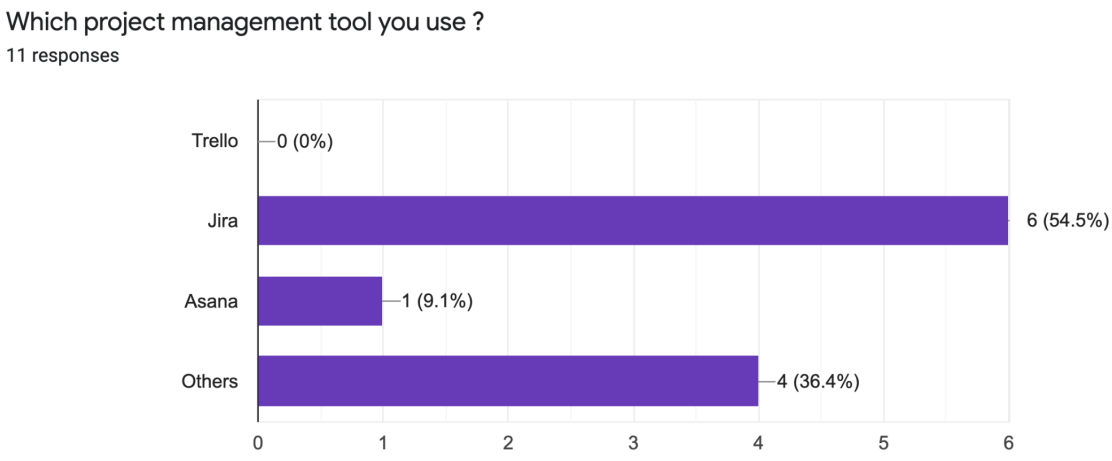


Figure 2: Response for project management tools used

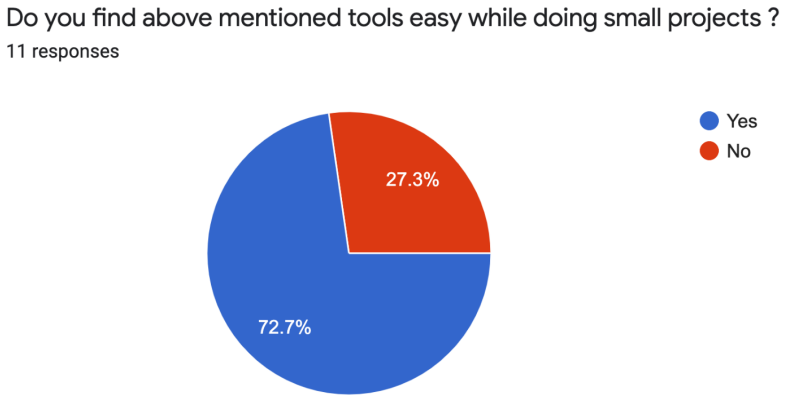


Figure 3: Response for ease of use in small projects

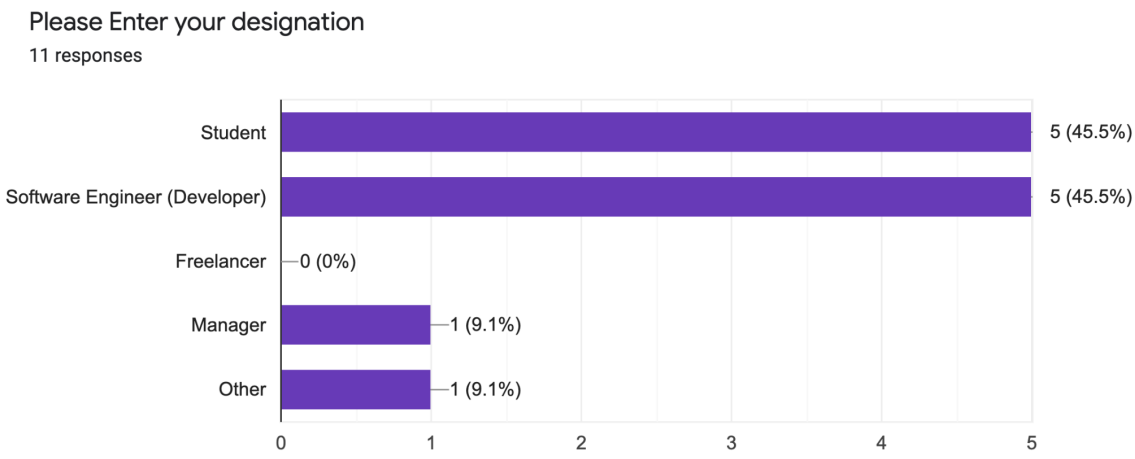


Figure 4: Response of Designation

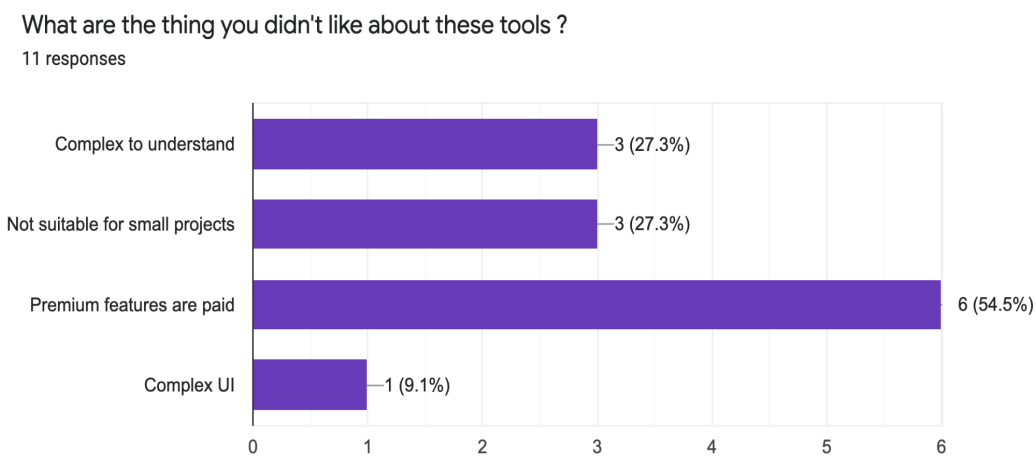


Figure 5: Response for complexity of project management tools

I. Functional Requirements

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs.

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.
The use case diagram for AceTask is as follow:

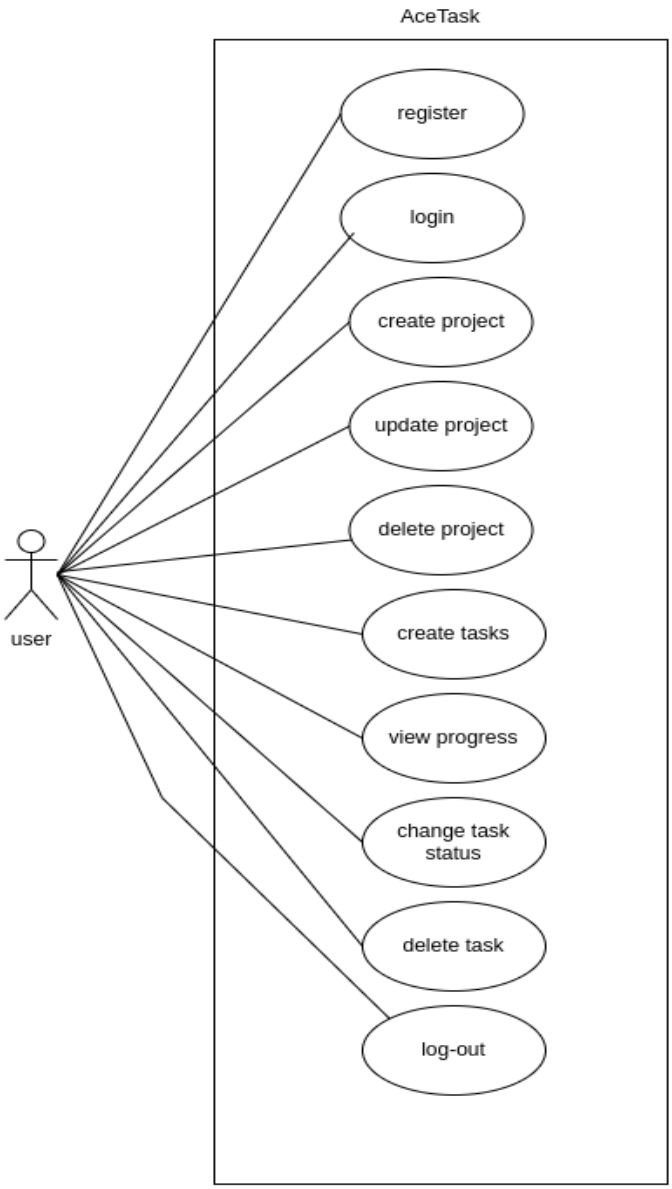


Figure 6: Use Case Diagram of AceTask

The functional requirement contains the following:

Table 1: Table for functional Requirements of AceTask

SN	Name	Use Case	Action/Process	Constraints	Dependency
1	User Registration	User should register in system	Fill the Form FullName: Email: Password: ConfirmPassword:	Should not leave required field empty	Must have valid email format and password and confirm password should match.
2	Login	User can login	Provide email and password	Must provide valid email and password	Must have registered user before proceeding login.
3	Project Dashboard	Users can create new projects and can see all existing projects	Login using valid credentials.	Must provide valid credentials to see project dashboard	Must be the user of the system and must be logged in to the system to view the project dashboard.
4	Create New Project	User can create new project	Fill the form Project Name: Unique Project Id: Project Description: Start Date: Estimated End Date:	Should not leave required field empty and unique project id must have 4-5 characters only	Must be the user of the system and must be logged in to the system to create a new project.
5	Project Operations	Update Project Details and	Fill the same form to update the project details after clicking	Should not leave required	Must be logged into the system

		Delete the project	on submit button and should click on delete button to delete project	field empty and unique project id must have 4-5 characters only	and should have projects to perform the project operations.
6	Project Board	Users can see the list of project tasks with task details which are listed in different states like TODO, IN PROGRESS and DONE. Users can move tasks from one state to another according to their status.	Users should click on update to change task status from one to another like from TODO to INPROGRESS or DONE.	Should not leave required field empty	Users must have a project to create new project tasks and to view on the project board.
7	Create New Project Task	Users can create project tasks adding necessary details of project tasks.	Fill the form Project Task Summary: Project Task Description: Due Date: Priority: Status:	Should not leave required field empty	Users must have a project to create new project tasks.
8	Project Task Operations	Users can read, update and delete the project tasks.	Users must click on the buttons like view, update and delete for corresponding operations.	Should not leave required field empty	Users must have a project task to perform correspondi ng task operations.
9	Search	Users can	Should provide	Should	Users must

		search existing projects and project tasks	some keywords to perform search operations	provide valid keywords to get results	have logged in into system
10	Project Deadline Report	Users can view project reports	Should navigate to the project deadline report tab	Must have a project to view project report	Users must have logged in into system
11	Logout	Users can log out from the system	Logged in users just must click on logout button	Users must have logged in into system	Users must have logged in into system

II. Non-Functional Requirements

The non-functional requirement contains the following:

Performance

AceTask is expected to perform smoothly for tracking the progress of a project. The project is divided into small project tasks and each task can track individually. AceTask has a simple user interface and the user with no experience with project management tools can operate easily.

Security

Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications. We have also implemented form validation for each input field. The Bcrypt [6] is used for the password encryption and JWT token based auth is used for the securing different API end points.

Usability

It is easier to use our application even without any guidance. We have tried to make this application simple and user friendly in every possible way by keeping its design simple and the language of our content simpler.

Scalability

AceTask can be considered as a scalable system. The system can be used by so many users. And will perform smoothly when there are so many users and requests in the system. Also, the system can adapt the new features and functionalities in the future version.

Availability

System works 24 hours a day so that users can access the system and the use of the system.

Reliability

AceTask can be considered as a reliable web app because we have implemented strong security. So, no need to worry about security. The user interface is so simple that every user can easily operate the system. The progress of the project can be seen on project dashboard. User can track their project task.

3.1.2. Feasibility Analysis

I. Technical

The proposed project will be a web-based application. The project is building using the latest, popular, and secured technology there are so many enterprise level software products that are made using this technology, so we do not expect there to be any technical problem for the proposed project. We are using JAVA programing language and spring boot framework for backend development along with different maven dependency and spring security framework. MySQL database is used for the storing the data in project. HTML, CSS, JS and React Library is used for the frontend development. These all technologies are technically feasible because these are all open-source technologies.

Tools Used:

IntelliJ IDEA: For writing JAVA code for backend development.

WebStorm: For writing HTML, CSS and JS code for frontend development.

Visual Studio Code: For writing HTML, CSS and JS code for frontend development.

MySQL Workbench: For database management.

Draw.io: For drawing different diagrams.

GitHub: For the management of the codes.

II. Operational

The user interface will be so simple and more effective to use the project would be using the most popular libraries and framework for development so, the project is expected to operate smoothly, and no user would face any problems. The system needs any browser and internet connection to operate.

III. Economical

The project will be economically feasible because of the following reasons.

1. The application will be a free application.
2. Project will be using open-source technologies.
3. Application will be platform independent.

Calculation of ROI (Return of Investment):

Development cost per month=15000

No of developer=3

Total development time=4 month

Total development cost=15000*4*3=180000

One time Domain registration cost=1500

Monthly server charge=1000

Yearly server charge=12*1000=12000

Total Investment=180000+1500+12000=193500

Yearly ad Income=70000

Yearly subscription-based Income=80000

Total Yearly Income =70000+80000=150000

Yearly income with 10% increase in service

0	-193500
1	150000
2	165000

Net Profit=165000+150000-193500

=121500

Average Annual profit=121500/2=60750

ROI= (profit/cost of investment) *100%

ROI= (60750/193500) *100%=31.39%

Payback period=2 year

Hence, we can conclude that our system will be economically feasible.

iv. Schedule (Gantt Chart)

Time management is the key for successful completion of any task. In this Gantt-chart we have represented our time allocation.

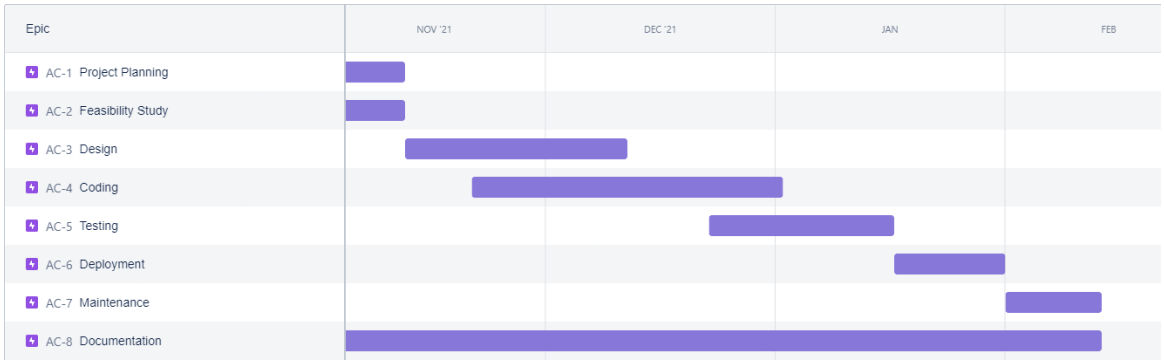


Figure 7: Gantt Chart of AceTask

3.1.3. Object Oriented Analysis

I. Object modeling using Class Diagrams

The class diagram is a static diagram that represents the static view of the application. It gives an overview of a software system by displaying classes, attributes, operations, and relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.

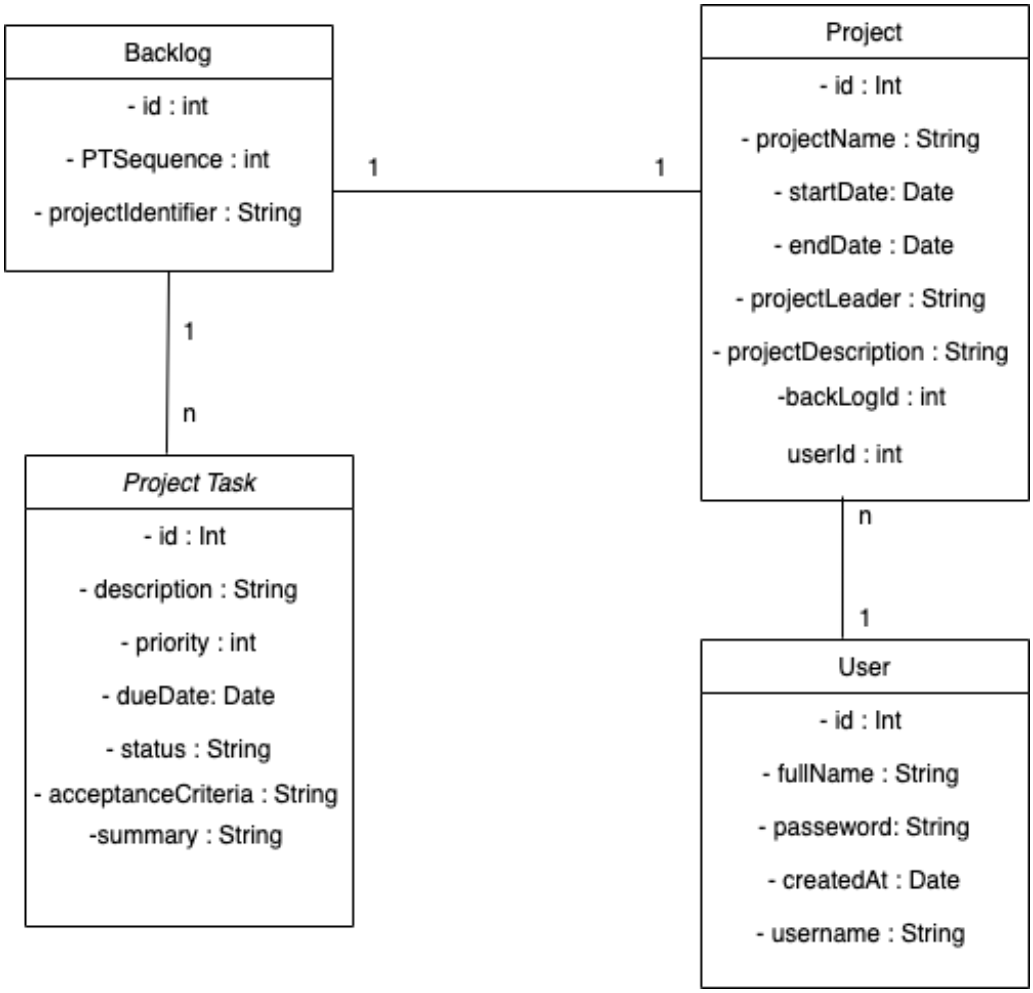


Figure 8: Class Diagram for AceTask

II. Dynamic modeling using Sequence Diagrams

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focused, and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent, and when. The below figure shows the sequence diagram of our system.

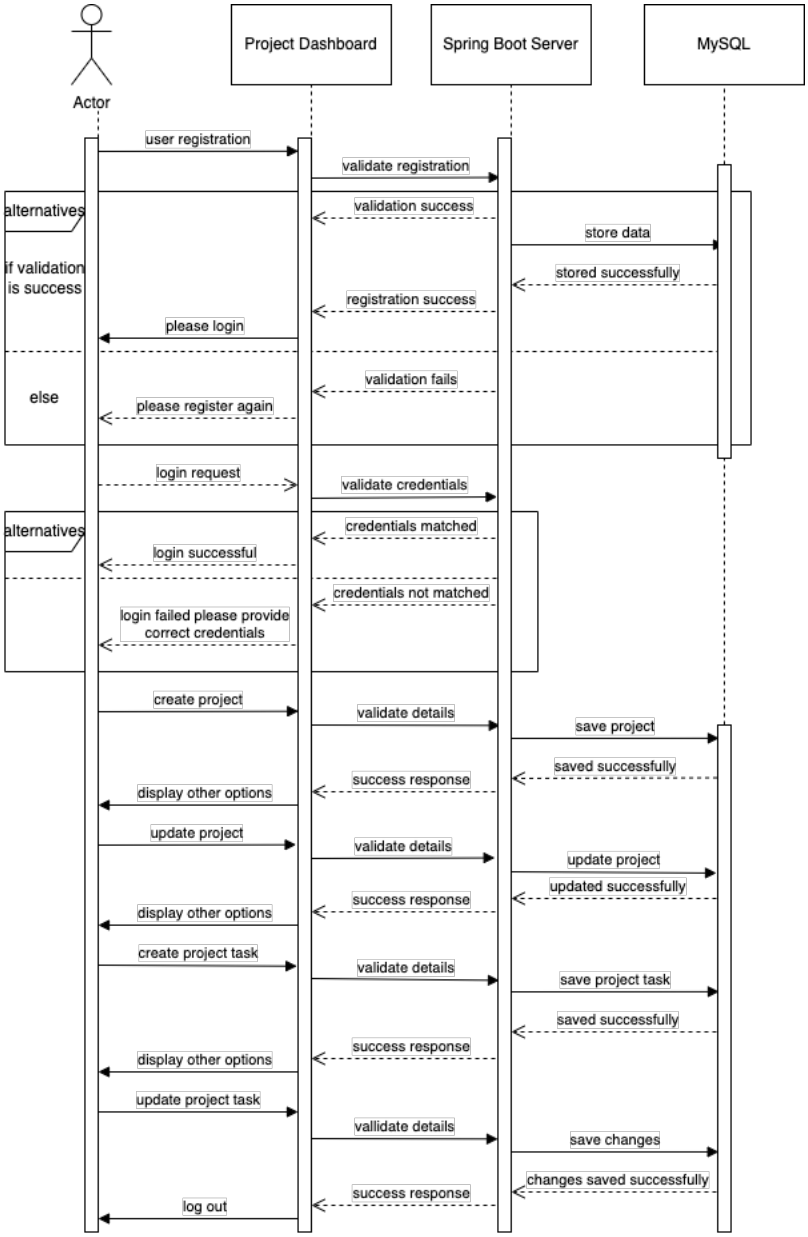


Figure 9: Sequence Diagram of ACETASK

III. Process modeling using Activity Diagram

The activity diagram is a UML diagram that describes the dynamic aspect of the system. It is basically a flowchart to represent the flow from one activity to another. They can be used to describe the business workflow or the operational workflow of any component in a system. The below figure shows the activity diagram of our system.

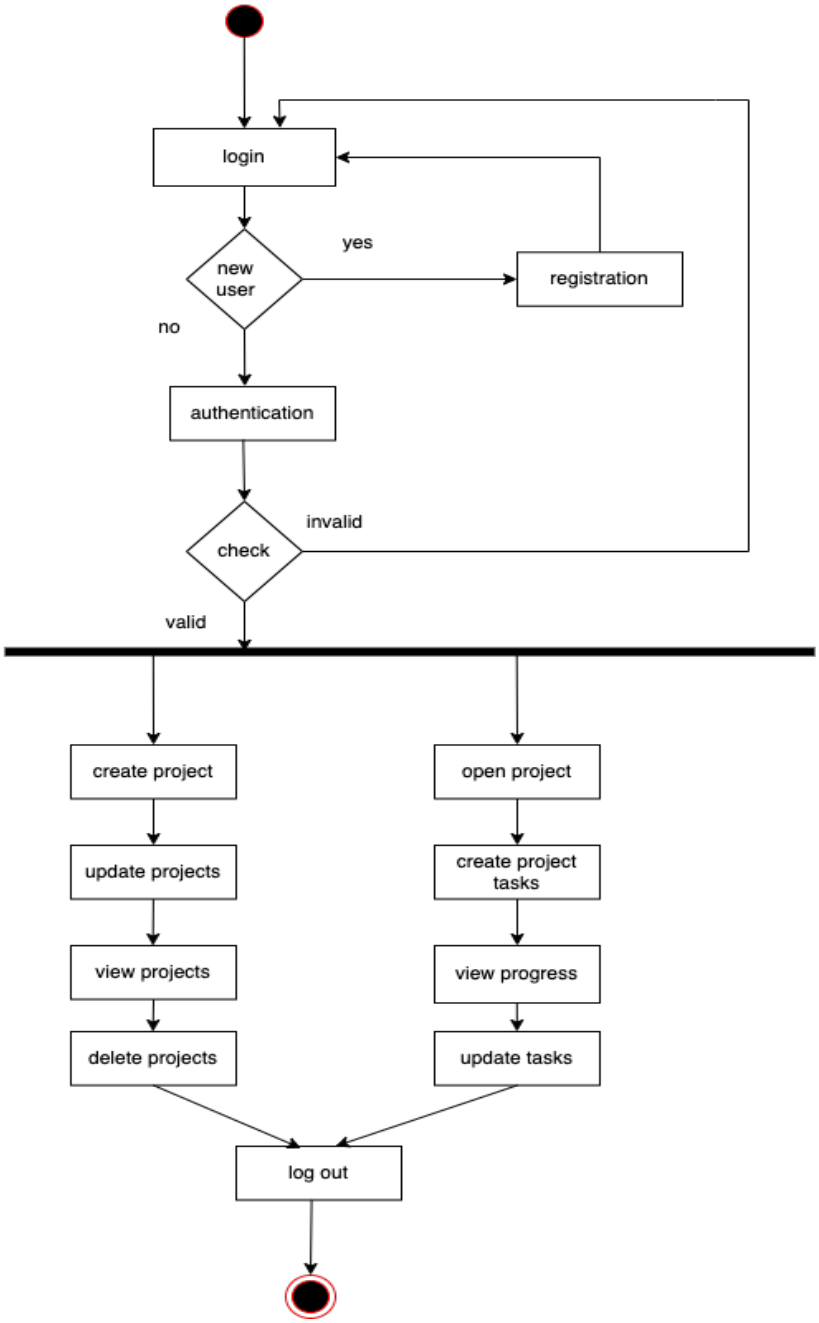


Figure 10: Activity Diagram of AceTask

CHAPTER 4

SYSTEM DESIGN

4.1. Design

Class Diagram

The static view of AceTask is represented by the class diagram containing all the fields and corresponding methods for each entity. The project and backlog has 1:1 relationship whereas user and project has 1: n relationship. Similarly backlog and project task has 1: n relationship which is represented below:

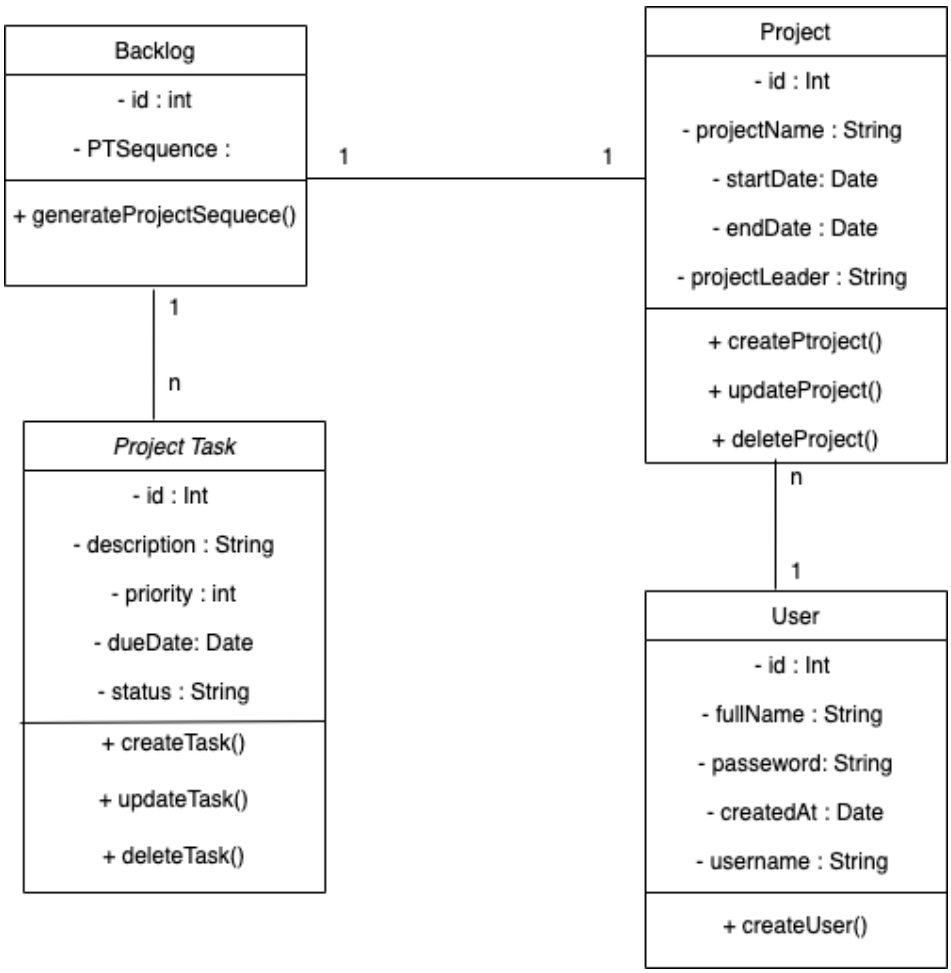


Figure 11 : Detailed Class Diagram of AceTask

Component Diagram

Component diagram describes the organization and writing of physical components in a system. Here the component diagram consists of all major components that are used to build the system. Our system has user interface, authentication, backend server, project management, project task management, persistence layer and database components. The component diagram for our system is shown below:

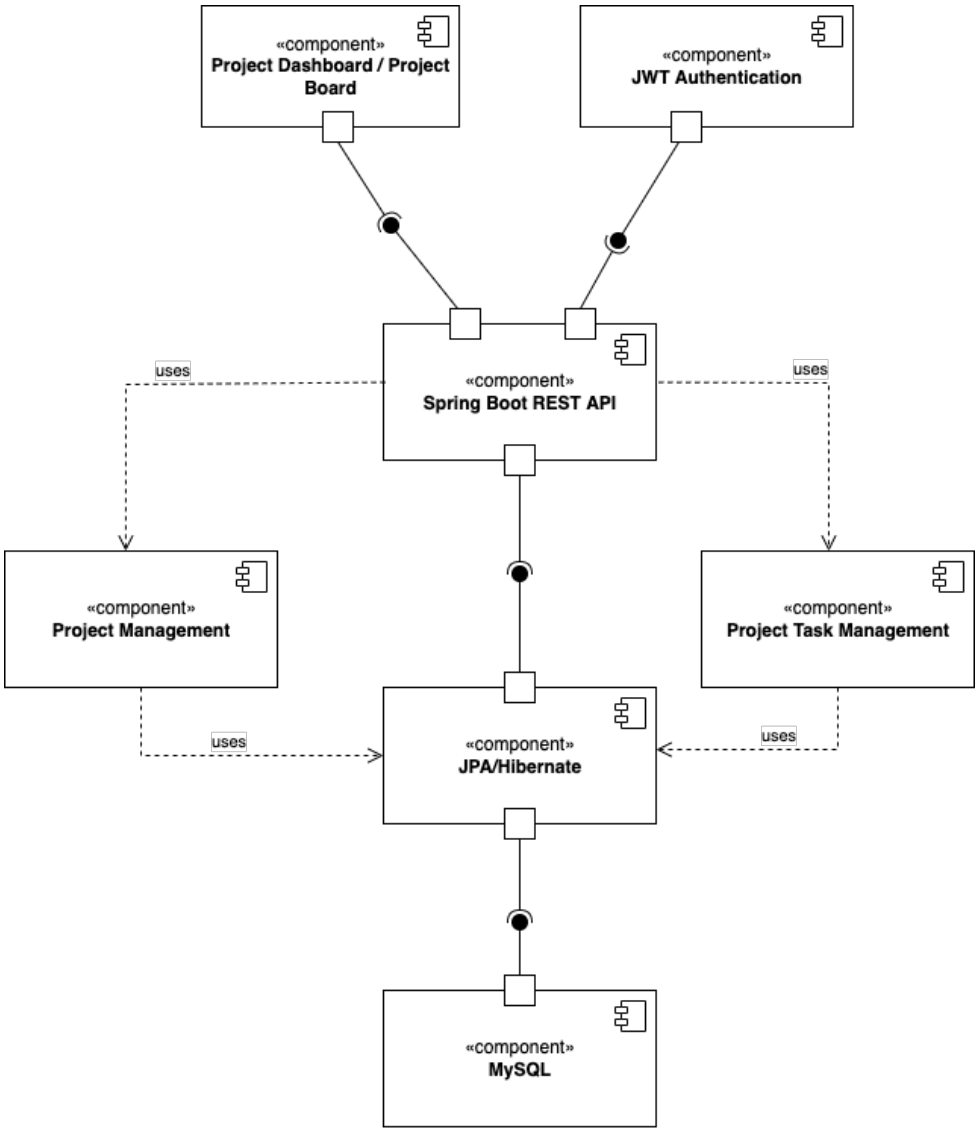


Figure 12: Component Diagram of AceTask

Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. The below diagram shows the deployment diagram of this system.

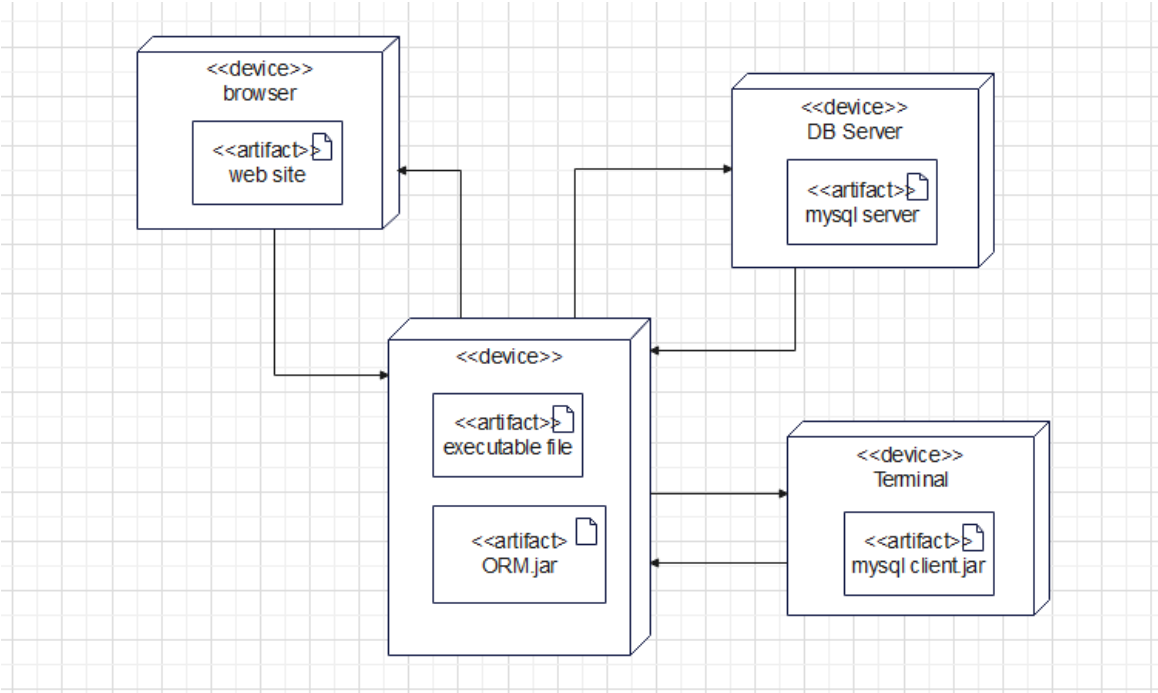


Figure 13: Deployment Diagram of AceTask

4.2. Algorithm Details

Binary Search Algorithm

We have used a **binary search algorithm** in our project for searching the project within the project dashboard as well as for searching the project task within the project board. [7]

Binary Search is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O (Log n).

The basic steps to perform Binary Search are:

- Begin with an interval covering the whole array.
- If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half.
- Otherwise, narrow to the upper half.
- Repeatedly check until the value is found, or their interval is empty.

Project Board Algorithm

We have used algorithms to provide a better user experience and given the name 'Project Board Algorithm'. Basically, what this algorithm does is when there are no project tasks existing for a specific project, it will display the user with the message of no project task on the board. If the specific project identifier is not present in the route or any user changes that it will show an error of no project Identifier does not exist.

Algorithm Steps:

Step 1: Start
Step 2: if length of project task > 1 goto step 3 else go to step 4
Step 3: if the routing is not correct
Display No Project tasks on this board
else
Display Project Id is not present
Step 4: Display Project tasks on Project Board
Step 5: Stop

Flow Chart of Project Board algorithm

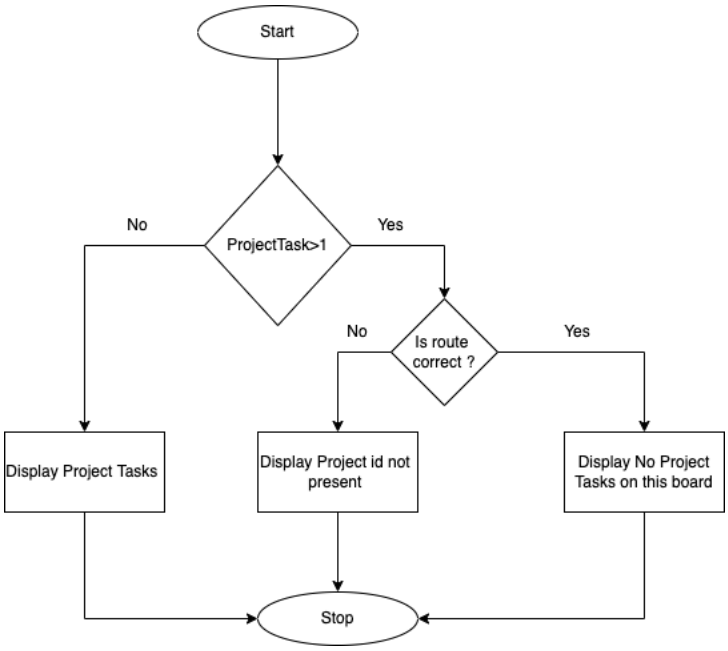


Figure 14: Flow chart of Project Board Algorithm

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1. Implementation

5.1.1. Tools Used

Backend End Tools

Java
Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of underlying computer architecture. We have used java-based framework Spring Boot to develop our application.

Spring Boot
Spring Boot is an open-source java-based framework used to create backend web services. It is developed by Pivotal Team and used to build stand-alone and production ready spring applications. We have used Spring boot to create backend web services, communicate with the database and implement security.

Front End Tools

JavaScript
JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation. In our application we have used JavaScript based library React to consume the backend API and for designing user interface.

React
React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in development of single-page web applications. We have used React in our application to consume the web services coming from the backend and manage the state of the user interface.

Database Tools

MySQL

MySQL is an open-source relational database management system (RDBMS). A relational organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. We have used MySQL as our primary database to store the details of user, project detail as well as tasks details.

5.1.2. Implementation Details of Modules

Sign up module

In the sign-up module, users can sign up for the site by providing their details like name, email and password for logging in later. After the validation process on the Server Side, the feedback is sent back to the client by a new dynamically generated web page. It is used to validate user input on Server Side because you can protect against the malicious users.

Login Module

In the login module, users can input their email and password. then users can have access to the dashboard where they get the information about their projects. Once they are logged in, they will have access to the dashboard and project board.

Homepage Module

This module in application is the accessibility of the main feature of the application after login. This feature in the module includes the dashboard where we can create multiple projects at a time. and search with a unique project identifier or project name.

Project Board Module

This module is the application that includes the tasks of the specific projects where progress of the project can be tracked. Users can create the project task with a certain description and put it into status TO DO and once he/she starts doing that project they can update the status and **IN, PROGRESS** and **DONE** once completed, also delete the task if not required.

Algorithm Implementations:

Binary Search Algorithm:

Code implementation of the binary search algorithm is as follows:

```
function binarySearchAlgorithm(sortedArray, key){
  let begin = 1;
  let end = sortedArray.length - 1;
  while (begin <= end) {
    let middle = Math.floor((begin + end) / 2);
    if (sortedArray[middle] === key) {
      // found the key
      return middle;
    } else if (sortedArray[middle] < key) {
      // continue searching to the right
      begin = middle + 1;
    } else {
      // search to the left
      end = middle - 1;
    }
  }
  // key wasn't found
  return -1;
}
```

Project Board Algorithm:
Code Implementation of project board algorithm is as follows:

Source Code:

```
const boardAlgorithm = (errors, project_tasks) => {
        if (project_tasks.length < 1) {
            if (errors.projectNotFound) {
                return (
                    <div className="alert alert-danger text-center" role="alert">
                        {errors.projectNotFound}
                    </div>
                )
            } else if (errors.projectIdentifier) {
                return (
                    <div className="alert alert-danger text-center" role="alert">
                        {errors.projectIdentifier}
                    </div>
```

```
            );
        } else {
            return (
                <div className="alert alert-info text-center" role="alert">
                    No Project Tasks on this board
                </div>
            );
        }
    } else {
        return <Backlog project_tasks_prop={project_tasks}/>;
    }
};
```

5.2. Testing

5.1.2. Test Cases for Unit Testing

Unit testing is the level of software testing where individual components/ units of software are tested. Unit testing is done to check whether the individual modules of source code are working are working properly or not. We have implemented using Junit and Mokito dependencies in Spring Boot.

In Code level we have implemented Unit Test for repository and Service level. The code snippet for unit testing for project repository is:

```
 @BeforeEach
   void setUp() {
      Project project=new Project();
      project.setProjectName("Demo");
      project.setProjectLeader("John");
      project.setProjectIdentifier("1234");
      project.setDescription("This is the project description");
      entityManager.persist(project);
   }
   @Test
   public void whenFindByIdentifier_thenReturnProject(){
      Project project=projectRepository.findById(1L).get();
      assertEquals(project.getProjectName(),"Demo");
   }
}
```

Similarly for Project Service code snippet for unit test is :

```
@BeforeEach
void setUp() {
    Project project = new Project();
    project.setProjectIdentifier("1234");
    project.setProjectLeader("John Doe");
    project.setProjectName("Demo");
    Mockito.when(projectRepository.findByProjectIdentifier("1234"))
            .thenReturn(project);
}

@Test
public void whenValidIdentifierOfProject_thenProjectShouldBeFound() {
    String projectIdentifier = "1234";
    String username = "John Doe";
    Project found = projectService.findProjectByIdentifier(projectIdentifier, username);
    assertEquals(projectIdentifier, found.getProjectIdentifier());
}
```

Table 2: Test Case for User Registration

Test case ID	Test Description	Test steps	Input test data	Expected result	Actual Result	Pass/ Fail
T-01	Signup with empty data	1. Click on Signup button 2.Leave input fields empty 3.Again click on register button	Fullname: Email: Password: Confirm Password:	Error messages for all fields will be shown with an error summary.	Error messages for all fields was shown with an error summary	Pass
T-02	Signup with valid data	1. Click on Signup button 2.Enter valid	Full Name:Arjun Gautam Email: arjungautam54	New user will be created successfully and redirect	New user was created successfully and redirect to the login	Pass

		data for all fields 3.Click on Signup button	31@gmail.com Password:Arjun12 Confirm Password:Arjun12	to the login page.	page.	
T-03	Signup with invalid data	1. Click on Signup button 2.Enter invalid data for any field or all fields 3.Click on Signup button	Email: sandipgole Fullname: Sandip Gole Password:1234 Confirm Password:1234	Error messages for all fields will be shown with an error summary.	Error messages for all fields were shown with an error summary.	Pass

Table 3: Test Case for User Login

Test case ID	Test Description	Test steps	Input test data	Expected result	Actual Result	Pass/ Fail
T-04	Login with valid data	1.Enter the valid email and password 2.Click on login button	Email:arjungautam5431@gmail.com Password:Arjun12	The login page should redirect to Project Dashboard	The login page redirect to Project Dashboard	Pass
T-05	Login with Invalid details	1.Enter incorrect email or password or both. 2.Click on login button	Email:kushalgiri@gmail.com Password:123	The login page should not redirect to project dashboard with login error message	The login page was not redirected to project dashboard with login error message	Pass
T-06	Login empty fields	1.Leave the entry fields empty	Email: Password:	The login page should not redirect to	The login page was not redirected	Pass

		2.Click on login button		homepage with login error message	to homepage with login error message	

Table 4: Test Case for Project

Test case ID	Test Description	Test steps	Input test data	Expected result	Actual Result	Pass/ Fail
T-07	Create project with valid data	1.Enter the valid project details. 2.Click on submit button	Project Name: Demo Unique Project Id: 1234 Project Description: Description of project Start Date: 03/22/2022 End Date: 04/12/202	The Project named demo should be created and redirected to the dashboard.	The Project named demo created and redirected to the dashboard.	Pass
T-08	Create Project with Empty Fields	1.Leave the project details empty 2.Click on submit button	Project Name: Unique Project Id: Project Description: Start Date: End Date:	The create project form should display the error message and should not redirect to the dashboard.	The create project form displayed the error message and didn't redirect to the dashboard.	Pass
T-09	Create project with invalid details	1.Enter the invalid project details. 2.Click on submit button	Project Name: Demo1 Unique Project Id:1 Project Description: Description of project	The create project form should display the error message and should not redirect	The create project form displayed the error message and was not redirect to	Pass

			Start Date:03/22/2022 End Date: 04/12/2022	to the dashboard.	the dashboard.	

Table 5: Test Case for Project Board

Test case ID	Test Description	Test steps	Input test data	Expected result	Actual Result	Pass/ Fail
T-10	Create Task with valid data	1.Enter valid project task details. 2.Click on submit button	Project Task Summary: Task 1 Description: Description of project Due Date :03/22/2022 Select Priority: High Select Status: TODO	The Project named demo should be created and redirected to the dashboard.	The Project named demo created and redirected to the dashboard.	Pass
T-11	Create Project with Empty Fields	1.Leave the project task summary empty 2.Click on submit button	Project Task Summary: Description: Due Date: Select Priority: Select Status:	Error message to include project summary should display.	Error message to include project summary displayed.	Pass

5.1.2. Test Cases for System Testing

Table 6: Test Cases for System Testing

Test case	Test Description	Expected result	Actual Result	Pass/ Fail
UI Validation	Validate that the UI is on point	UI should be easy to use.	The UI is easy to use.	Pass
Login page	Validate the login page functionality	User should be able to login and logout	Users can login and logout.	Pass
Dashboard	Validate that the dashboard page is working and button functionality.	The dashboard should be working properly.	The dashboard is working properly.	Pass
Project Board	Validate that the project board can be updated from status TODO, IN PROGRESS and DONE.	Project Task should be able to update and color code should work properly.	Project tasks are updated with proper color coding and in respective places of TODO, IN PROGRESS and DONE task are updated.	Pass
Input Data	Validate the form	The form should be editable and clickable.	The form is editable and clickable.	Pass

5.3. Result Analysis

As a result of our project a software developer or any person with multiple projects and tasks can handle their project and task to manage their workload and make the balance between multiple projects. We have completed from design to implementation for the whole scope we have defined. Though for multiple user and role management we haven't implemented this can be implemented in future.

Thus, as a result of our project we have properly built the **Project Management Tool** which can help a software developer as well as any technical person for managing multiple projects and related tasks to meet their deadline.

CHAPTER 6

CONCLUSION AND FUTURE RECOMMENDATIONS

6.1. Conclusion

The **AceTask** is a web application which is intended to facilitate Software developers and freelancers to make proper management of the projects and related tasks. To implement this project, we have created mainly two modules Project Module and Project Board modules. Project Module contains one or more projects and works like a dashboard where users can see the details of projects with project deadline report. Similarly, the Project Board of that related task contains the track of progress of project tasks with their details of Priority and Status.

The application has met all the criteria and goals of the project. The project will help to solve the problem faced by indie developers for daily time management to meet their deadline.

6.2. Future Recommendations

- New features such as an admin dashboard can be added.
- Role management features can be added.
- To improve user experience, we can add drag and drop features.
- Applications can be made mobile responsive or make separate Mobile applications.

REFERENCES

[1] M. J. B. Kabeyi, "khanacademy," Int J Sci Res, 12 April 2019. [Online]. Available: https://www.khanacademy.org/humanities/ap-art-history/ancient-mediterranean-ap/ancient-egypt-ap/a/old-kingdom-the-great-pyramids-of-giza. [Accessed January 2019].

[2] A. Mishra, "atlassian," ACM SIGSOFT Software Engineering Notes, 5 january 2013. [Online]. Available: https://www.atlassian.com/software/jira.

[3] H. A. Johnson, "trello," Journal of the Medical Library Association: JMLA, 20 march 2017. [Online]. Available: https://trello.com/en.

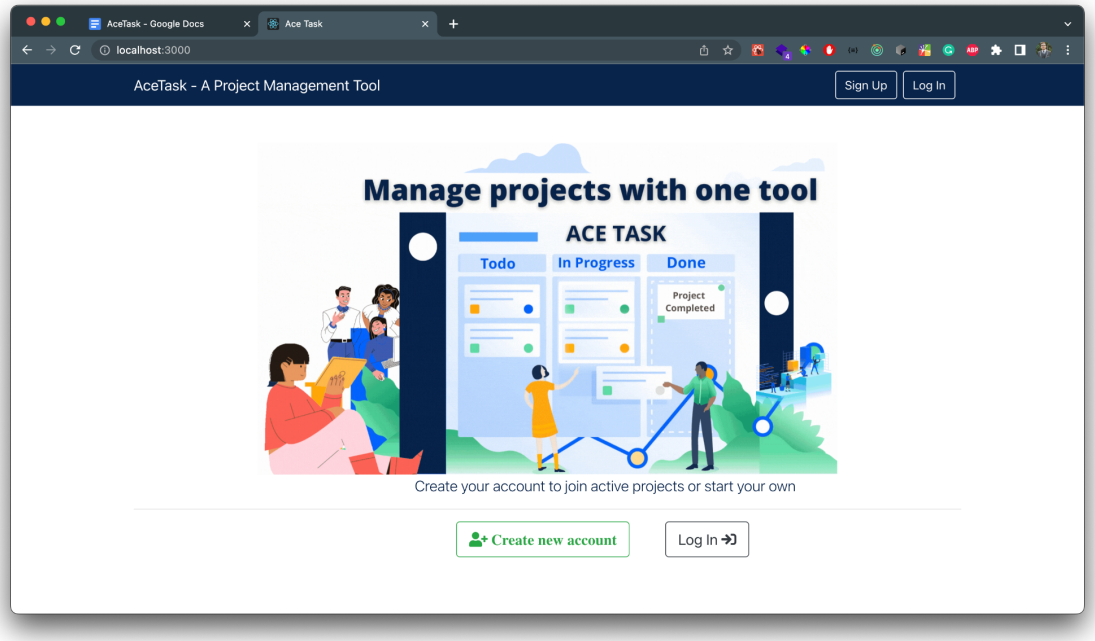
[4] R. B. R. Wakode, "kanbanize," IJSRD-International Journal for Scientific Research & Development, 20 April 2015. [Online]. Available: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban.

[5] M. Goeminne, "A historical dataset for the GNOME ecosystem.," Conference on Mining Software Repositories (MSR) , 25 December 2013. [Online]. Available: https://bitbucket.org/product/guides/getting-started/overview.

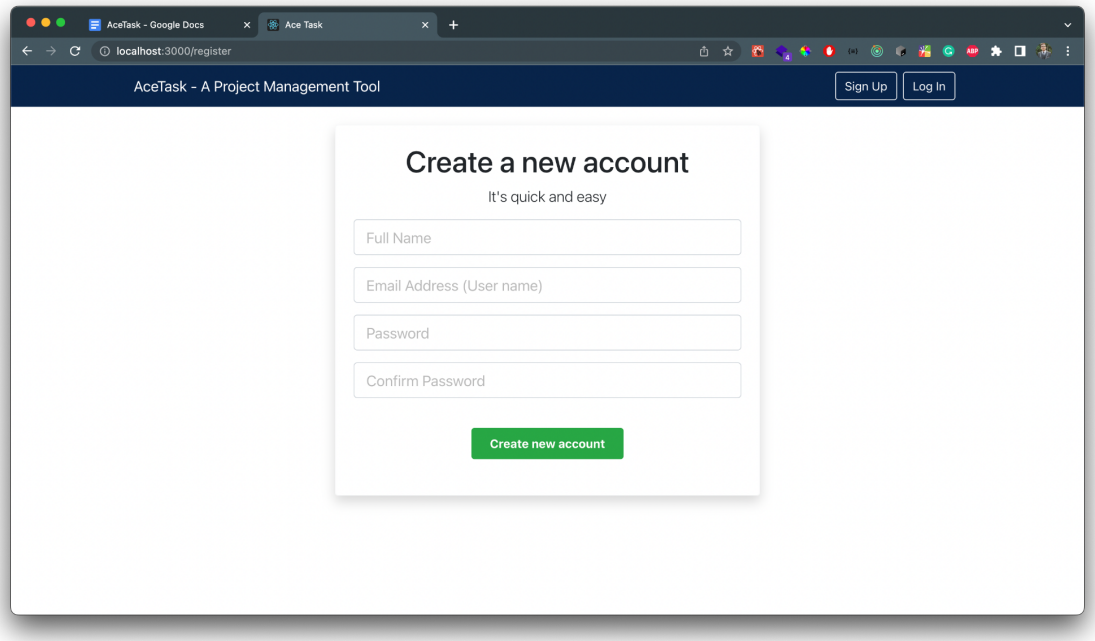
[6] N. &. M. Provos, "Bcrypt algorithm," USENIX., 1999. [Online]. Available: https://en.wikipedia.org/wiki/Bcrypt.

[7] D. Knuth, "Optimum binary search trees," Acta informatica, january 1971. [Online]. Available: https://www.cs.bgu.ac.il/~michaluz/seminar/Knuth71.pdf.

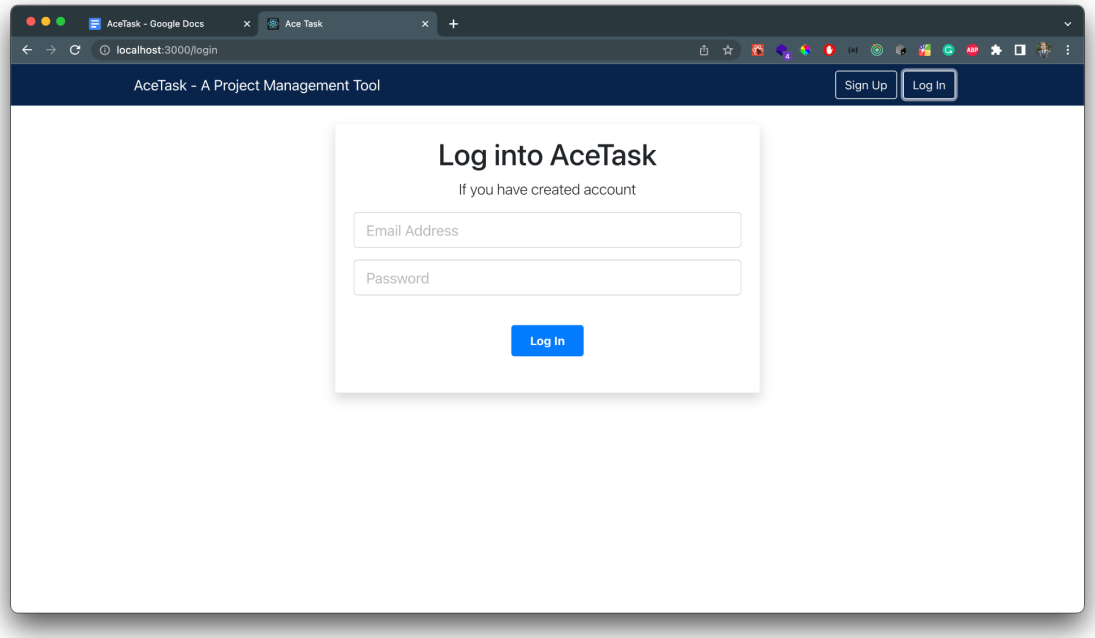
APPENDIX I



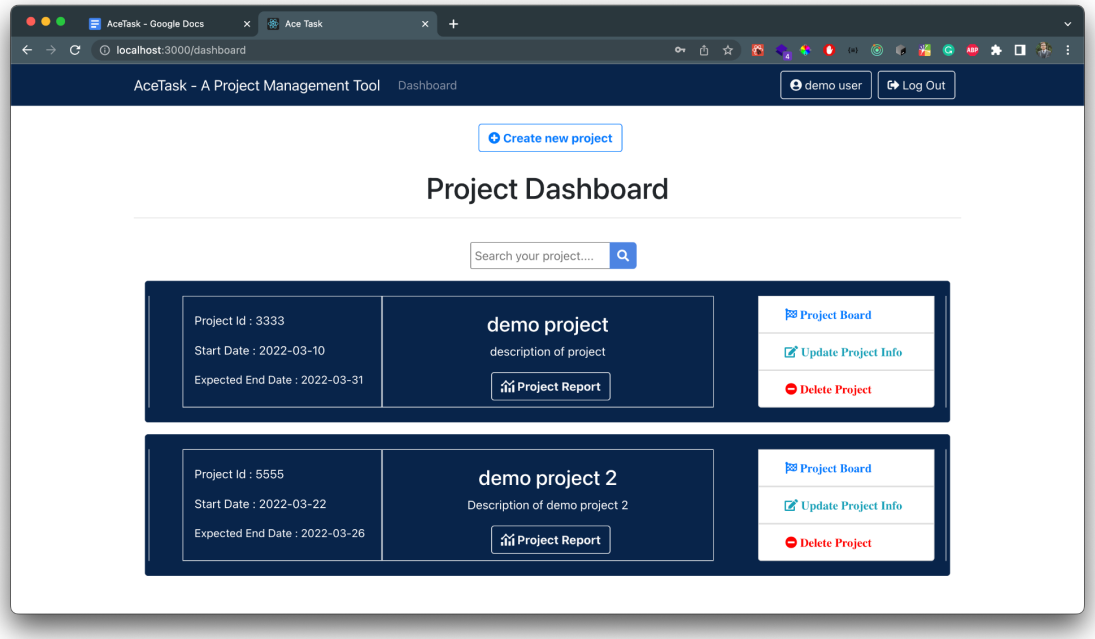
Landing Page



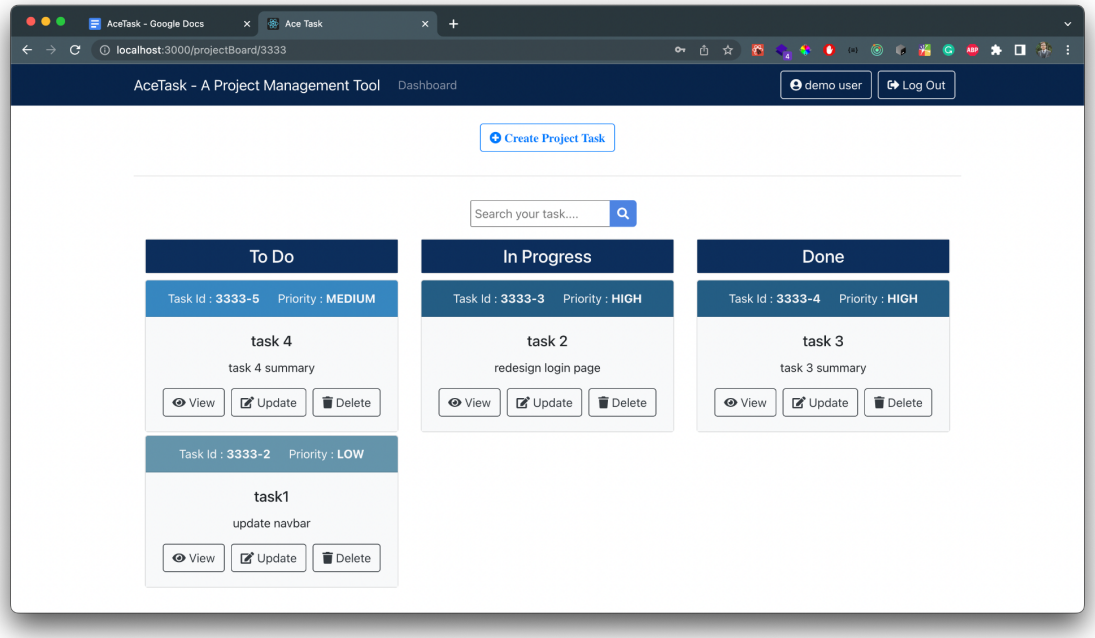
Signup Page



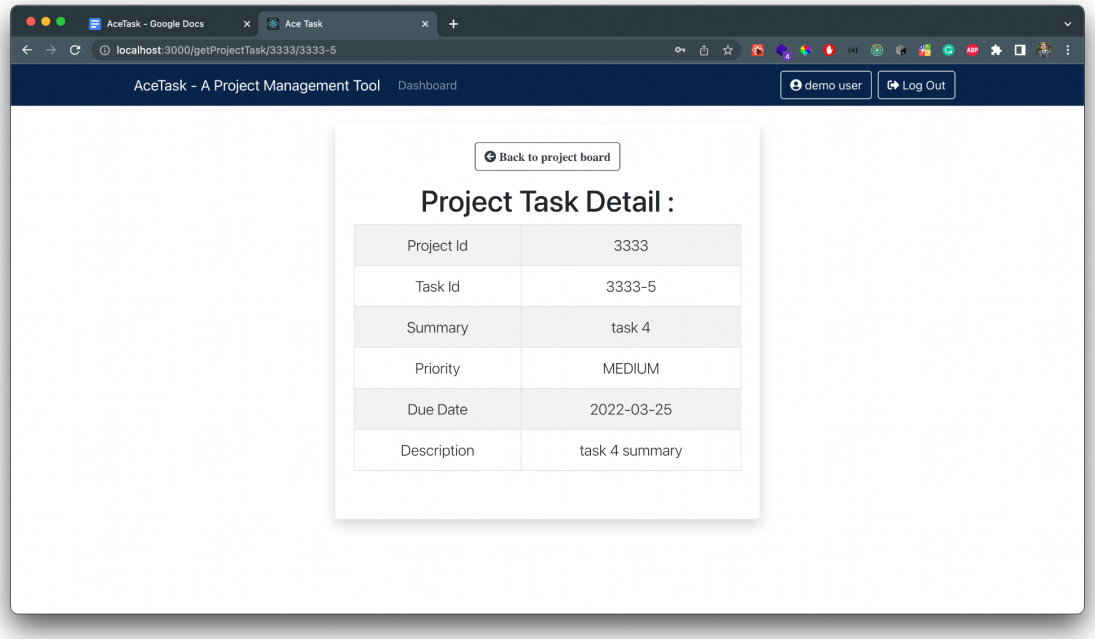
Login Page



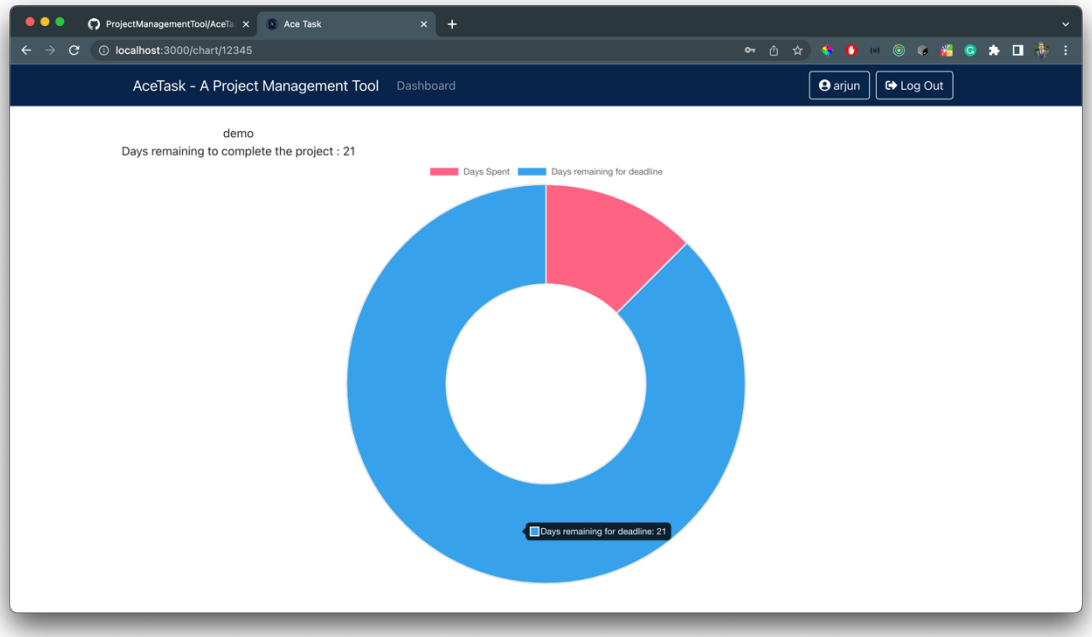
Project Dashboard



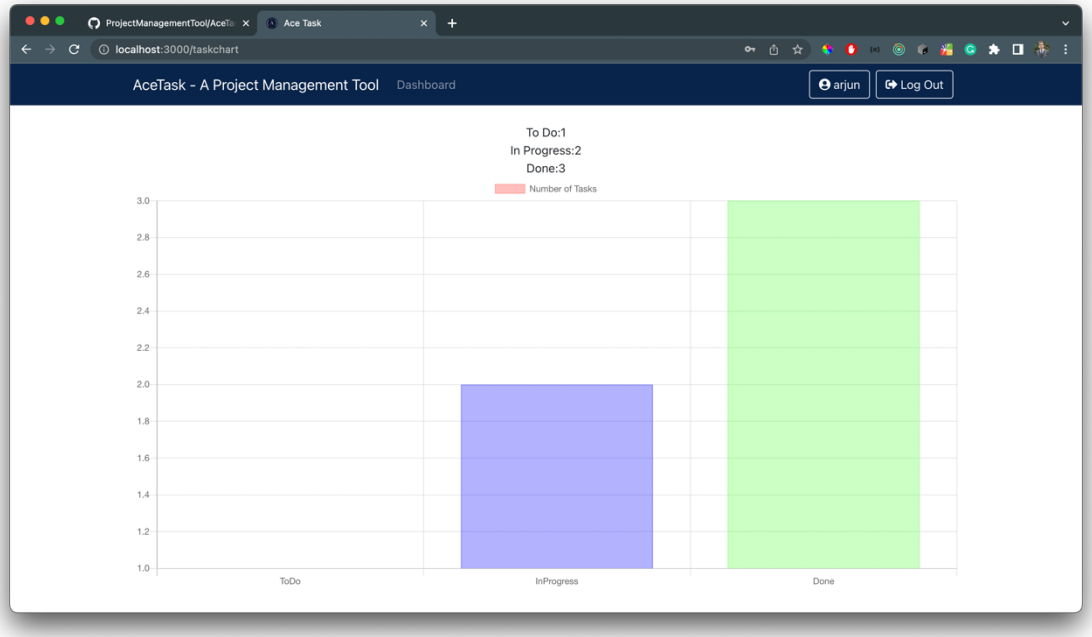
Project Board



Task Details Page



Project Deadline report



Project Task Report

APPENDIX II

Source Code of major modules for REST API Services:

UserService.java

```
package codes.laser.ppmtool.services;
import codes.laser.ppmtool.exceptions.UsernameAlreadyExistsException;
import codes.laser.ppmtool.model.User;
import codes.laser.ppmtool.repositories.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Service;

@Service
public class UserService {
   @Autowired
   private UserRepository userRepository;

   @Autowired
   private BCryptPasswordEncoder bCryptPasswordEncoder;

   public User saveUser(User newUser) {

      try {
         newUser.setPassword(bCryptPasswordEncoder.encode(newUser.getPassword()));

         //Username has to be unique(exception)
         newUser.setUsername(newUser.getUsername());

         newUser.setConfirmPassword("");

         return userRepository.save(newUser);
      } catch (Exception e) {
         throw new UsernameAlreadyExistsException("Username'" +
newUser.getUsername() + "'already exists");
      }
   }
}
```

ProjectService.java

```
package codes.laser.ppmtool.services;
import codes.laser.ppmtool.exceptions.ProjectIDException;
import codes.laser.ppmtool.exceptions.ProjectNotFoundException;
import codes.laser.ppmtool.model.Backlog;
import codes.laser.ppmtool.model.Project;
import codes.laser.ppmtool.model.User;
import codes.laser.ppmtool.repositories.BacklogRepository;
import codes.laser.ppmtool.repositories.ProjectRepository;
import codes.laser.ppmtool.repositories.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class ProjectService {

   @Autowired
   private ProjectRepository projectRepository;

   @Autowired
   private BacklogRepository backlogRepository;

   @Autowired
   private UserRepository userRepository;

   public Project saveOrUpdateProject(Project project, String username) {

      if (project.getId() != null) {
         Project existingProject =
projectRepository.findByProjectIdentifier(project.getProjectIdentifier());
         if (existingProject != null &&
(!existingProject.getProjectLeader().equals(username))) {
            throw new ProjectNotFoundException("Project not found in your account");
         } else if (existingProject == null) {
            throw new ProjectNotFoundException("Project with ID: '" +
project.getProjectIdentifier() + "' cannot be updated because it doesn't exist");
         }
      }

      try {
         User user = userRepository.findByUsername(username);
         project.setUser(user);
         project.setProjectLeader(user.getUsername());
```

```
            project.setProjectIdentifier(project.getProjectIdentifier().toUpperCase());

            if (project.getId() == null) {
                Backlog backlog = new Backlog();
                project.setBacklog(backlog);
                backlog.setProject(project);
                backlog.setProjectIdentifier(project.getProjectIdentifier().toUpperCase());
            }
            if (project.getId() != null) {

project.setBacklog(backlogRepository.findByProjectIdentifier(project.getProjectIdentifier
().toUpperCase()));
            }
            return projectRepository.save(project);
        } catch (Exception e) {
            throw new ProjectIDException("Project Id '" +
project.getProjectIdentifier().toUpperCase() + "'already exists.");
        }
    }

    public Project findProjectByIdentifier(String projectId, String username) {
        Project project =
projectRepository.findByProjectIdentifier(projectId.toUpperCase());
        if (project == null) {
            throw new ProjectIDException("Project Id'" + projectId + "'does not exist.");
        }
        if (!project.getProjectLeader().equals(username)) {
            throw new ProjectNotFoundException("Project not found in your account");
        }
        return project;
    }

    public Iterable<Project> findAllProjects(String username) {
        return projectRepository.findAllByProjectLeader(username);
    }

    public void deleteProjectByIdentifier(String projectId, String username) {

        projectRepository.delete(findProjectByIdentifier(projectId, username));
}
}
```

ProjectTaskService.java

```
package codes.laser.ppmtool.services;
import codes.laser.ppmtool.exceptions.ProjectNotFoundException;
import codes.laser.ppmtool.model.Backlog;
import codes.laser.ppmtool.model.Project;
import codes.laser.ppmtool.model.ProjectTask;
import codes.laser.ppmtool.repositories.BacklogRepository;
import codes.laser.ppmtool.repositories.ProjectRepository;
import codes.laser.ppmtool.repositories.ProjectTaskRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class ProjectTaskService {
   @Autowired
   private BacklogRepository backlogRepository;

   @Autowired
   private ProjectTaskRepository projectTaskRepository;

   @Autowired
   private ProjectRepository projectRepository;

   @Autowired
   private ProjectService projectService;


   public ProjectTask addProjectTask(String projectIdentifier, ProjectTask projectTask,
String username) {
      //Exceptions : Project not found


      //PTs to be added to specific project , !=project , backlog exists
      Backlog backlog = projectService.findProjectByIdentifier(projectIdentifier,
username).getBacklog();//backlogRepository.findByProjectIdentifier(projectIdentifier);
      //set the backlog to Project Task
      projectTask.setBacklog(backlog);

      //we want our project sequence to be like this IDPRO-1  IDPRO-2
      Integer BacklogSequence = backlog.getPTSequence();

      //Update the Backlog sequence
```

```
        BacklogSequence++;
        backlog.setPTSequence(BacklogSequence);

        //Add Sequence to project Task
        projectTask.setProjectSequence(backlog.getProjectIdentifier() + "-" +
BacklogSequence);
        projectTask.setProjectIdentifier(projectIdentifier);

        //Initial priority when priority null
        if (projectTask.getPriority() == 0 || projectTask.getPriority() == null) {//In the future
we need projectTask.getPriority()==0 to handle the form
            projectTask.setPriority(3);
        }

        //Initial status when status null
        if (projectTask.getStatus() == "" || projectTask.getStatus() == null) {
            projectTask.setStatus("TO_DO");
        }
        return projectTaskRepository.save(projectTask);


    }

    public Iterable<ProjectTask> findBacklogById(String id, String username) {
        projectService.findProjectByIdentifier(id, username);
        return projectTaskRepository.findByProjectIdentifierOrderByPriority(id);
    }

    public ProjectTask findPTByProjectSequence(String backlog_id, String pt_id, String
username) {
        //We are searching on the right backlog
        projectService.findProjectByIdentifier(backlog_id, username);
        //make sure that our task exists

        ProjectTask projectTask = projectTaskRepository.findByProjectSequence(pt_id);
        {
            if (projectTask == null) {
                throw new ProjectNotFoundException("Project task :'" + pt_id + "'not found ");
            }
        }

        //make sure that the backlog/project id in the path corresponds to the right project
        if (!projectTask.getProjectIdentifier().equals(backlog_id)) {
```

```
            throw new ProjectNotFoundException("Project Task '" + pt_id + "'does not exist
in project'" + backlog_id);
        }

        return projectTask;

    }

    public ProjectTask updateByProjectSequence(ProjectTask updatedTask, String
backlog_id, String pt_id,String username) {
        ProjectTask projectTask = findPTByProjectSequence(backlog_id, pt_id,username);

        projectTask = updatedTask;
        return projectTaskRepository.save(projectTask);
    }

    public void deletePTByProjectSequence(String backlog_id, String pt_id,String
username) {
        ProjectTask projectTask = findPTByProjectSequence(backlog_id, pt_id,username);
        projectTaskRepository.delete(projectTask);

    }
}
```