**CHAPTER 1**

**INTRODUCTION**

This document is a Software Requirement Specification (SRS) and it presents the main Objective of this project being to design a to-do list application. This is the initial draft for the SRS and it will be used for the extensions. Todo list application will allow a user to create new

tasks assign them a title and due date and choose a project for that task to belong to. They will

need to use a text-based user interface via the command-line. Once they are using the application, the user should be able to also edit,

mark as done or remove tasks. They can also quit and save the current task list to file, and then

restart the application with the former state restored.

 Also, this document is prepared by following IEEE conventions for software requirement specification.

**PURPOSE**

The purpose of this document is to build a to do list application. The application will allow a user to create new tasks, assign them a title, due date, choose a project for that task to belong to.  edit task,

mark as done or remove tasks, quit and save the current task list to file, and finally the user will be able to

restart the application with the former state restored.

**PROJECT SCOPE**

Presently the application web-based and provides a clean and user-friendly interface to the users.

**CHAPTER 2**

**LITERATURE SURVEY**

The to do list application is a new attempt to speed up the process of managing time of engineers and others working with computers and varied operating systems more often and also those in educational institute’s The existing systems are time-consuming and there are many difficulties faced by these groups of persons to get information about each individual and work schedules. This software provides a solution to these problems. It provides an interactive user interface and helps users in an organisation to get information immediately at that instant of time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | |  | | | | |

**Chapter 3**: **REQUIREMENTS**

**3.1 Functional Requirements**

● Model a task with a task title, due date, status and project

● Display a collection of tasks, sorted by date or filtered by project

● Support the ability to add, edit, mark as done, or remove tasks

● Support a text-based user interface

● Load and save task list to file

**3.2 Non- Functional Requirements**

**Safety Requirements**

The application should not provide catastrophic failure, such as a disk crash, and operating system incompatibility.

Also, the application will provide other Software Quality Attributes as USABILITY satisfactions

.

**3.3 Hardware Requirements**

* Computers

**3.4 Software Requirements**

● Mac, Windows, Linux operating systems

● Internet

**Model Implementation**

* Agile Methodology
* Github
* Slack
* Trello

**Chapter 4**: **System Architecture**

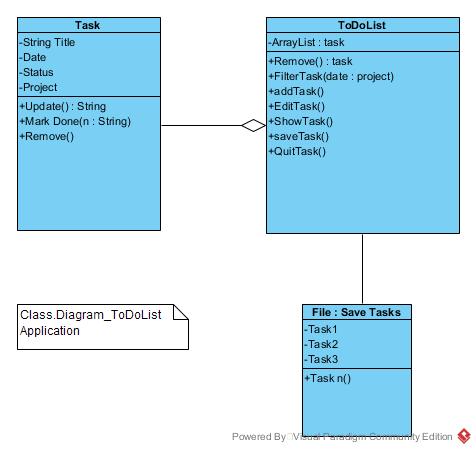
**4.1 Architecture**

Diagram:

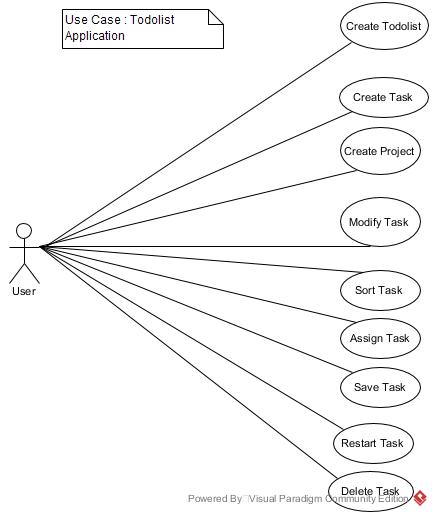
**Chapter 5**: **Design and Implementation**

**Product Features**

**5.1 class diagram design**



**5.2 Use case diagram**



**5.3 Sequence diagram**

**5.4 Activity diagram**

**5.5 Entity diagram**

**Chapter 6: SNAPSHOTS**

**Chapter 7: TESTING AND RESULT**

The reason behind testing is to find errors

**7.1Unit Testing**

Unit testing will be done on some methods to check if the outputs meets the requirements specified

**7.2 Acceptance Testing**

Acceptance test will be carried out using command line and based on the program interface.

Here the entire software will be executed to see that each module works individually and as a whole

**Chapter 8** : **TIME FRAME**

**Chapter 9**

**CONCLUSION**