HDSSD

Requirements Specification (RS)

Tick Task

Jean Redfearn

13/10/22

Table of Contents

1 Introduction **2**

1.1 Purpose 3

1.2 Project Scope 3

1.3 Definitions, Acronyms, and Abbreviations 4

2 User Requirements Definition **4**

3 Requirements Specification **4**

3.1 Functional requirements 5

3.1.1 Use Case Diagram 5

3.1.2 Requirement 1 : Register 6

3.1.3 Requirement 2 : Login 7

3.1.4 Requirement 3 : Add task 9

3.1.5 Requirement 4 : View all task 10

3.1.6 Requirement 5 : Edit task 11

3.2 Non-Functional Requirements 11

3.2.1 Performance/Response time requirement 11

3.2.2 Availability requirement 11

3.2.3 Recover requirement 12

3.2.4 Robustness requirement 12

3.2.5 Security requirement 12

3.2.6 Reliability requirement 12

3.2.7 Maintainability requirement 12

3.2.8 Portability requirement 12

3.2.9 Extendibility requirement 12

4 Interface requirements **13**

4.1 GUI 13

5 System Architecture **15**

6 System Evolution **15**

Introduction

1.1 Purpose

The purpose of this document is to set the specifications for developing the software of a to do list and planner web application. The web application will allow users to add, edit and view all tasks. To use this software, users will need to register or login to be able to view all their data. The intended customers for Tick Task are created for people who need to manage their personal tasks efficiently that are free of use and user friendly.

1.2 Project Scope

Project purpose

The project's objective is to create a web application that will allow users to manage their tasks effectively by being able to add tasks, edit tasks and also will be able to view all tasks logged. Logging into the system is required to be able to connect to the data.

Project scope

As this project is based on personal experience on wanting to develop a personal task manager. Research has been based on personal experience and from stakeholders to meet their personal needs that will be able to assist in managing their daily tasks.

Project objectives

This project aims to have the following functionalities

- Users will have to have to register to be able to login and view all data
- Users will be able to add task
- User will be able to edit tasks
- User will be able to view all logged tasks

Success conditions

The web application will be fully deployed with the above detailed functionalities with the basic functions that will be able to develop further in the future.

Project limitations

The development of this project is subject to several limitations, such as a short deadline which leads to minimal research and only being able to implement a few functionalities to the software.

1.3 Definitions, Acronyms, and Abbreviations

Use case
A scenario where the product could potentially be used
User
The individual who will be using the end product software
Class diagram
A UML diagram designed to model the software's structure, displaying attributes and operations of each class
Main flow
A basic scenario that describes user's interaction of the system
Alternate flow
An alternate flow that is different from the main flow but still leads to the same goal to the main flow
Exceptional flow
A flow that might cause a user preventing them from completing the goal
Precondition
The state that the system is required to start the use case
Post condition
The state that the system will be in after the use case
Functional requirement
Is a function that is required of the system for the users the accomplish their goals
Non-functional requirement
Defines the attributes of the system
GUI
Graphical User Interface such as computer monitors

2 User Requirements Definition

A few functionalities have been required from users but there will be 3 main functionalities implemented for this project to meet the basic requirements for now. The following functionalities listed below

- User requires an option where they can edit any of their tasks
- User requires a function where they will be able to view all tasks entered
- User requires a user friendly, minimal style design

3 Requirements Specification

This section outlines all specification requirements including functional and non-functional requirements and a use case diagram for the software for better visibility.

3.1 Functional requirements

These functions below demonstrates what the system will need to do to meet users requirement

- A non - registered user will be able to register
- A registered user will be able to login to their account
- A registered user will be able to add a new tasks to the list
- A registered user will be able to edit any tasks on the list
- A registered user will be able to view all current tasks

3.1.1 Use Case Diagram

The Use Case Diagram Figure 3.1.1 below provides an overview of all functional requirements.

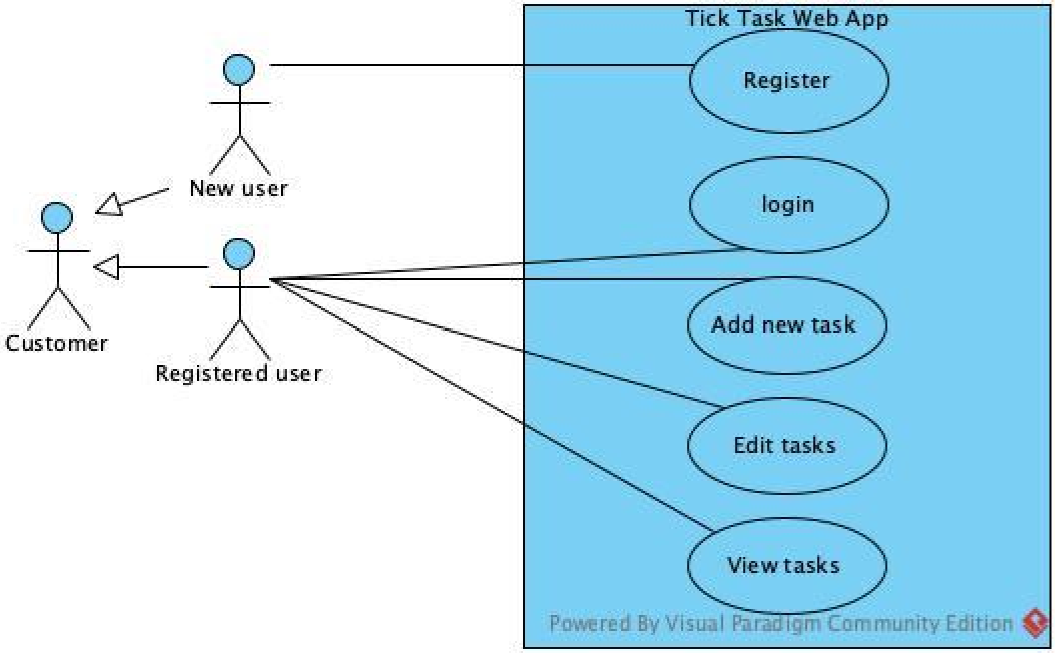


Figure 3.1.1 Use Case Diagram for Tick Task software

3.1.2 Requirement 1 : Register

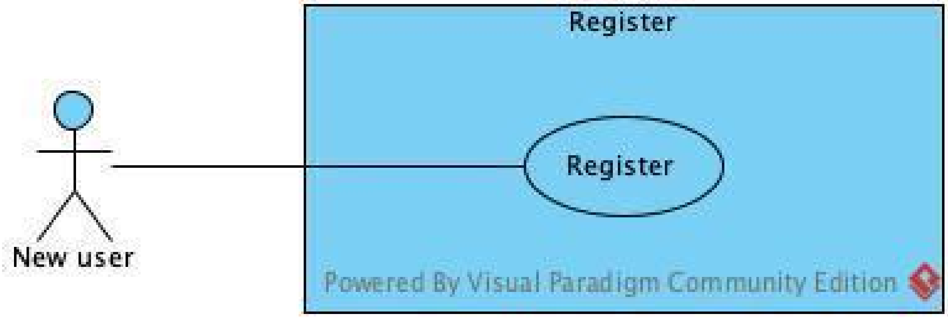


Figure 3.1.2 : Register use case

3.1.2.1 Description & Priority

Before a user will be able to use the login, registration is required to have login details. This is the 1st priority, without any registered account, the user won't be able to access their account and data.

3.1.2.2 Use Case

Scope

Registration for the software

Description

User needs to register before being able to login and access data to their task

Flow Description

Precondition

The user does not have any login credentials yet

Activation

The use case starts when the user clicks to register on the login page

Main flow

1. The user clicks the register option on the login page for users who don't have an account yet
2. The system will show the registration pages for user to fill in personal details
3. The user fills in all the details and click register button

4. The system will register the user details

Alternate flow

A1 : Input an existing registered email

1. The user input an email that is already registered in the system
2. The system will ask the user to check the details again
3. The main flow will continue from the main flow 3

Termination

The system displays a message that the registration has been successful

3.1.3 Requirement 2 : Login

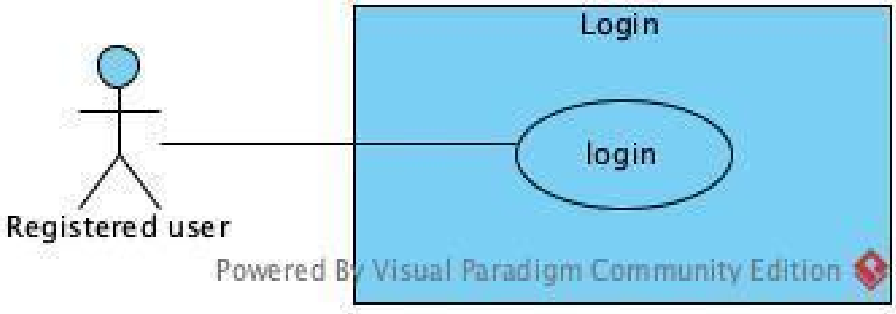


Figure 3.1.3 Login use case

3.1.3.1 Description & Priority

The user will have to login before being able to access their account. This is priority number 2, after registration is completed.

3.1.3.2 Use Case

Scope

The user needs to login to be able to access their account data

Description

A registered user, login to their account to access the software

<u>Flow Description</u>

Precondition

The user needs must registered previously and have login details

Activation

This use case starts when the user clicks on the register button from the login page

Main flow

1. The user input login details and clicks the login button
2. The system validates the details and signs the user in

Alternate flow

A1 : The user input incorrect login details

1. The system prompts the user that the details are incorrect and to input the correct details again
2. The user input the details again
3. The use case continues the main flow from 2

Exceptional flow

E1 : User forgot login details

1. The user enter incorrect login details
2. The system prompt the user that the details are incorrect
3. The user in correct details again, preventing user from logging in to their account
4. System will ask user to reset password

Termination

The system presents the tasks page with all users data that has previously been logged

3.1.4 Requirement 3 : Add task

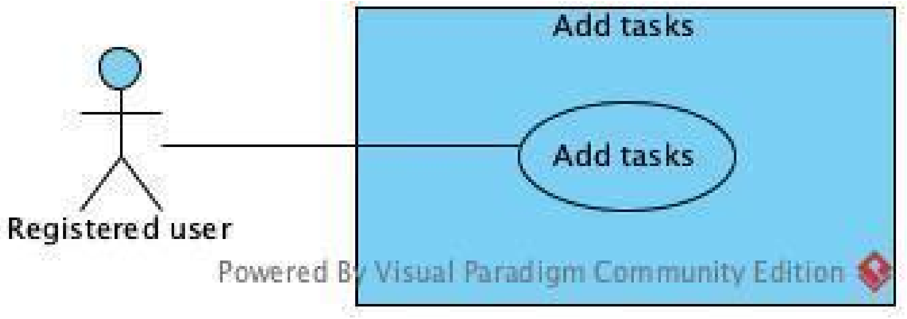


Figure 3.1.4 Add task use case

3.1.4.1 Description & Priority

Adding a task is the important functionality in the task page. In order to make other requirements 4 and 5 to work, this one has to be implemented first and it is the third priority of this software.

3.1.4.2 Use Case

Scope

The scope of this use case is to users will be able to add new tasks

Description

When the user first registers or login, user will need to add tasks to the software

Flow Description

Precondition

User must be logged in in order to add any tasks

Activation

This use case starts when an the user logins and clicks add new task

Main flow

1. User clicks add new task

2. System displays details for user to fill in tasks details

3. User fills in details and clicks submit

4. System saves the new task added and displays message "New task added"

Termination

The system displays the message " New task added" to the user

3.1.5 Requirement 4 : View all task

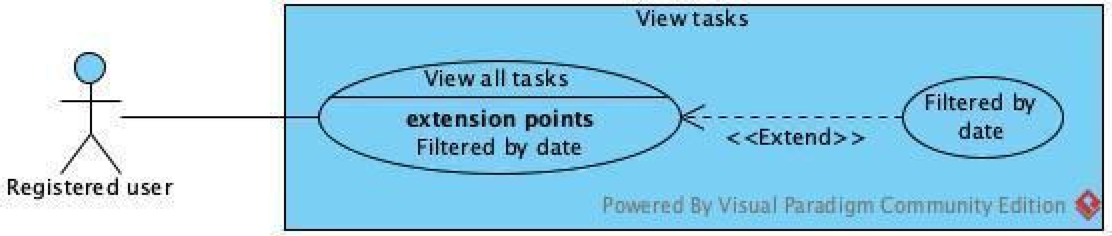


Figure 3.1.5 View tasks use case

3.1.5.1 Description & Priority

This requirement is the 4th priority in the software. The user will be able to view all tasks entered previously.

3.1.5.2 Use Case

Scope

The scope of this use case is users will be able to view all tasks entered

Description

This use case will allow users to view all added tasks previously

<u>Flow Description</u>

Precondition

The user is logged into their account

Activation

This use case starts when the user logs in to the task page

Main flow

1. The system identifies the user logged in to their account and display all tasks
2. User adds a new tasks and the tasks show up on the list

Termination

The system displays all tasks including the new added tasks

3.1.6 Requirement 5 : Edit task

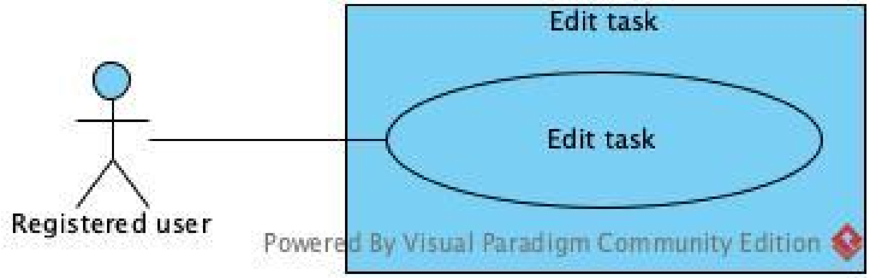


Figure 3.1.6 Edit task use case

3.1.6.1 Description & Priority

This requirement is the 5th priority. In order to implement this requirement, all other requirements must be implemented first.

3.1.6.2 Use Case

Scope

The scope of this use case is when user wants to edit a task

Description

This use case describes the function of when the user wants to edit a task that has been entered previously

Flow Description

Precondition

The system is in initialisation mode when user is logged in and has tasks entered previously

Activation

This use case starts when the user clicks on the edit button

Main flow

1. User clicks on edit button and edits the task
2. System updates the task to the update

3.2 Non-Functional Requirements

The non-functional requirements describes Tick Task's software quality attributes as below

3.2.1 Performance/Response time requirement

Depending on the users internet speed and device used, the response time shall take no longer than 2-5 seconds

3.2.2 Availability requirement

The web application will be deployed to cloud and the user will be able to access the software from anywhere and anytime as long as they have a device connected to the internet

3.2.3 Recover requirement

The software and all data will be uploaded and backed up to the cloud to help with recovery and prevent lost data.

3.2.4 Robustness requirement

Before the website will be deployed or any other new functionality, different types of testing will be made thoroughly to make sure no errors will be injected and cause any system failures. Any errors that may occur from users, error handling will be implemented to prevent any fails that may occur from user's input or use of the software.

3.2.5 Security requirement

To access personal data, users will have to pass the authentication by logging in to their account. In a later stage, when the web application is available for public use, network and data security will also be implemented.

3.2.6 Reliability requirement

User input validation, error handling, system testing are made regularly to help prevent system failure. Deploying software to the cloud and all data backed up can also help prevent any data loss.

3.2.7 Maintainability requirement

A maintenance or update may be required which shall be less than 30 mins each time.

3.2.8 Portability requirement

The software will be deployed on the cloud, therefore it should be accessible from any device with internet connection.

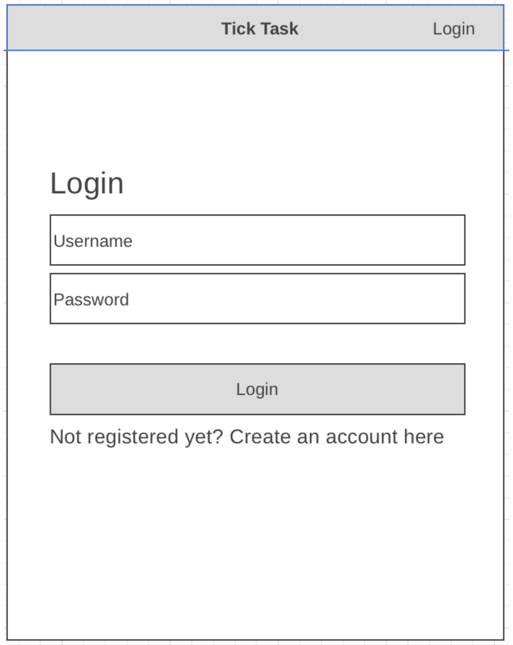
3.2.9 Extendibility requirement

Due to time limitation of this project, the software will not be open to public yet however it is designed to be able to add on more functionalities in the future and open for public use as freeware

4 Interface requirements

4.1 GUI

Three mock-up pages have been created to display the design for each key page. The first page is the login page where users can enter their login details and if the user does not have any login details, there will be a link provided below the login details for the user to link to register as image below Figure 4.1.1. The register page layout is a few details asking for user's details that are required for login as image below Figure 4.1.2. Once the user has logged in, the task page will show where the user is able to do all the other functionalities here which is add tasks, view tasks or search for any tasks as image Figure 4.1.3



Not registered yet? Create an account here

Figure 4.1.1 Login page

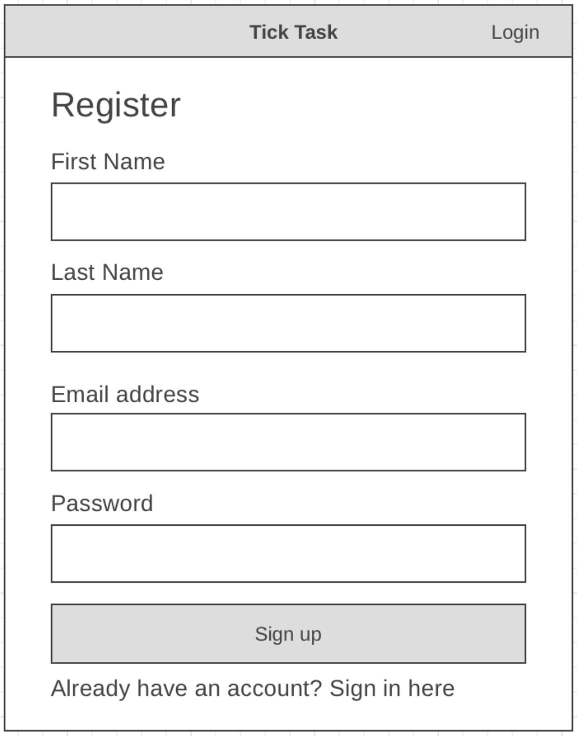
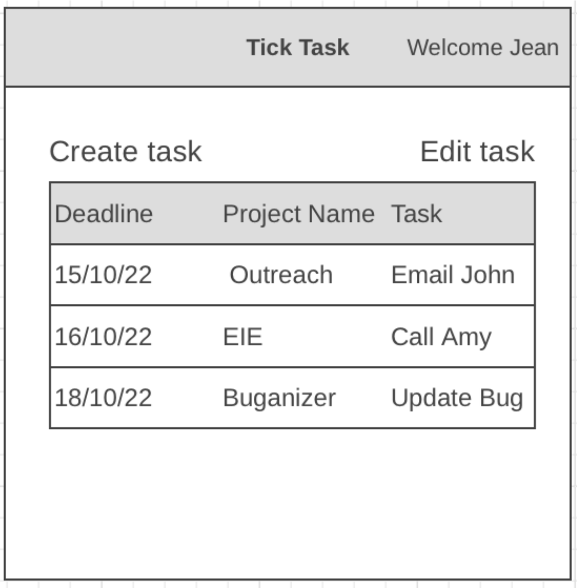


Figure 4.1.2 Register page



Deadline	Project Name	Task
15/10/22	Outreach	Email John
16/10/22	EIE	Call Amy
18/10/22	Buganizer	Update Bug

Figure 4.1.3 Task page

5 System Architecture

A class diagram Figure 5.1 has been created to help structure the software and to visualise the object oriented system. Each diagram displays attributes names, types of attributes and operations.

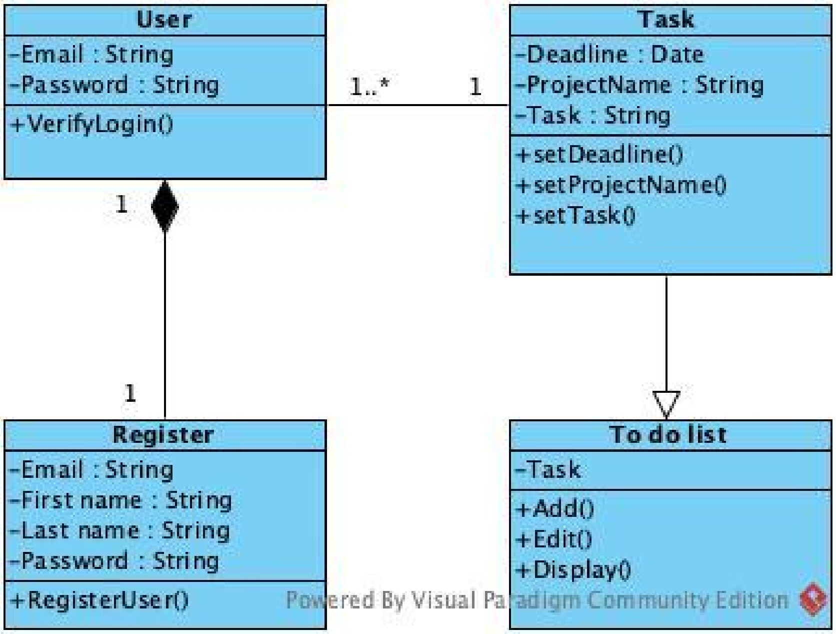


Figure 5.1 Class diagram for Tick Task software application

6 System Evolution

More functionalities will be added over time such as

1. Being able to add more functions for tasks such as search and delete tasks
2. Set a reminder for when the tasks are due
3. Mark tasks done and calculate tasks achieved