

HDSSD

Requirements Specification (RS)

Tick Task

Jean Redfearn

13/10/22

Table of Contents

Definitions, Acronyms, and Abbreviations.....	2
1 Introduction	3
1.1 Purpose.....	3
1.2 Intended Audience	3
1.3 Intended Use.....	3
1.4 Project purpose.....	4
1.5 Project scope	4
1.6 Project objectives.....	4
1.7 Success conditions	5
1.8 Project limitations.....	5
2 User Requirements Definition	5
3 Requirements Specification	6
3.1 Functional requirements	6
3.2 Use Case Diagram.....	7
3.2.1 Requirement 1 – Register.....	8
3.2.2 Requirement 2 – Login	9
3.2.3 Requirement 3 - Add task	12
3.2.4 Requirement 4 - View all tasks	13
3.2.5 Requirement 5 - Edit task	14
3.2.6 Requirement 6 – Delete task	16
3.3 Non-Functional Requirements	17
3.3.1 Performance/Response time requirement.....	17
3.3.2 Availability requirement.....	17
3.3.3 Recover requirement	17
3.3.4 Robustness requirement.....	17
3.3.5 Security requirement.....	18
3.3.6 Reliability requirement	18
3.3.7 Maintainability requirement.....	18
3.3.8 Portability requirement.....	18
3.3.9 Extendibility requirement	18
4 Interface Requirements.....	19
4.1 GUI.....	19
5 System Architecture.....	23
6 System Evolution	24

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

1 Introduction

1.1 Purpose

The purpose of this document is to establish and define the specifications for a software utility that will improve upon the concept of a digital to-do list, through the development of a web-based planner application. This web application will allow users to create, edit, delete and view all of their tasks in one robust, easy to navigate digital environment.

To operate this software, users will first be required to register for the service, and then, will subsequently need to login to the web application to view all of their data (tasks).

The intended goal for Tick Task is to create a free to use and user-friendly environment for people who need to manage their personal tasks efficiently, and sometimes, on the go.

1.2 Intended Audience

This report is intended as guidance or a reference for any software developers who are creating, or researching the creation of, a web-based software utility, the coding involved, the troubleshooting required, and the implementation of the completed utility.

1.3 Intended Use

This software utility, once complete, is intended for use by any office-based staff who deal with multiple concurrently run tasks in their day-to-day work life.

This application format can be used by various business sectors, and even in a personal time-management capacity, however, the target demographic for the use of this software is staff from the technology sector, who deal with a wide variety of complex and semi-complex tasks relating to standalone projects, with defined time limitations, within their company, i.e. with internal accountability.

1.4 Project purpose

The project's main objective is to create a working web application that will allow users to manage manually added tasks effectively. This will be achieved through providing a platform upon which the user can add tasks, edit, delete their existing tasks, and track all the tasks that they log. Logging into the system is required to ensure data security and fidelity.

This project is influenced by the researchers own personal experience, wanting to develop a personal task manager for themselves and their work colleagues in order to aid efficiency in the business that they work.

Research has been performed directly with interested stakeholders, in order to find the best way to meet their personal needs, with a specific goal of assisting in assisting them in the management of their daily tasks.

1.5 Project scope

This project will utilize knowledge and experienced gathered throughout the researchers educational and career paths, to code a standalone software utility that can be hosted on, and used through, an easily accessible medium, the internet.

This project will record all aspects of the coding and construction process, and report on troubleshooting of the application while in Beta.

This project will limit itself to the creation of the software, as described above, and with the below objectives, and, it will not be formatted for use on other proprietary systems, through bridging or other means of cross-coding.

1.6 Project objectives

This project aims to create a utility with the following functionalities;

- 1) Users will have to have to register to be able to login and view all data

- 2) Users will be able to add tasks
- 3) User will be able to edit tasks
- 4) User will be able to view all logged tasks
- 5) User will be able to delete tasks

1.7 Succes conditions

The web application will be fully deployed and operable with the above detailed functionalities, with these basic functions forming a base of utility that will be able to be developed further in the future.

1.8 Project limitations

The development of this project is subject to several limitations, first and foremost of which is a short delivery deadline, this leads to the bare minimum of functional research, which, in-turn, allows for the implementation of only a few basic functionalities for the software.

2 User Requirements Definition

Many different potential functionalities for the software were identified through researching user needs with the potential users of the software. However, only 3 main key functionalities will be implemented for this project, to meet the basic requirements. Other functionalities may be added during a later iteration of the software.

The key functionalities are listed below;

- 1) User requires a function where they will be able to view all tasks entered
- 2) User requires the ability to create new tasks
- 3) User requires the ability to edit or delete any of their tasks
- 4) User requires a user-friendly, minimalist style design UI

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 of the intended functionality.

3.1 Functional requirements

The functions below demonstrate what the system will need to do to meet the user requirements identified in **Section 2**;

- An un-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 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
- A registered user will be able to delete any tasks on the list

3.2 Use Case Diagram

The 'Use Case' diagram, **Figure 3.2.A**, below, provides an overview of all functional requirements.

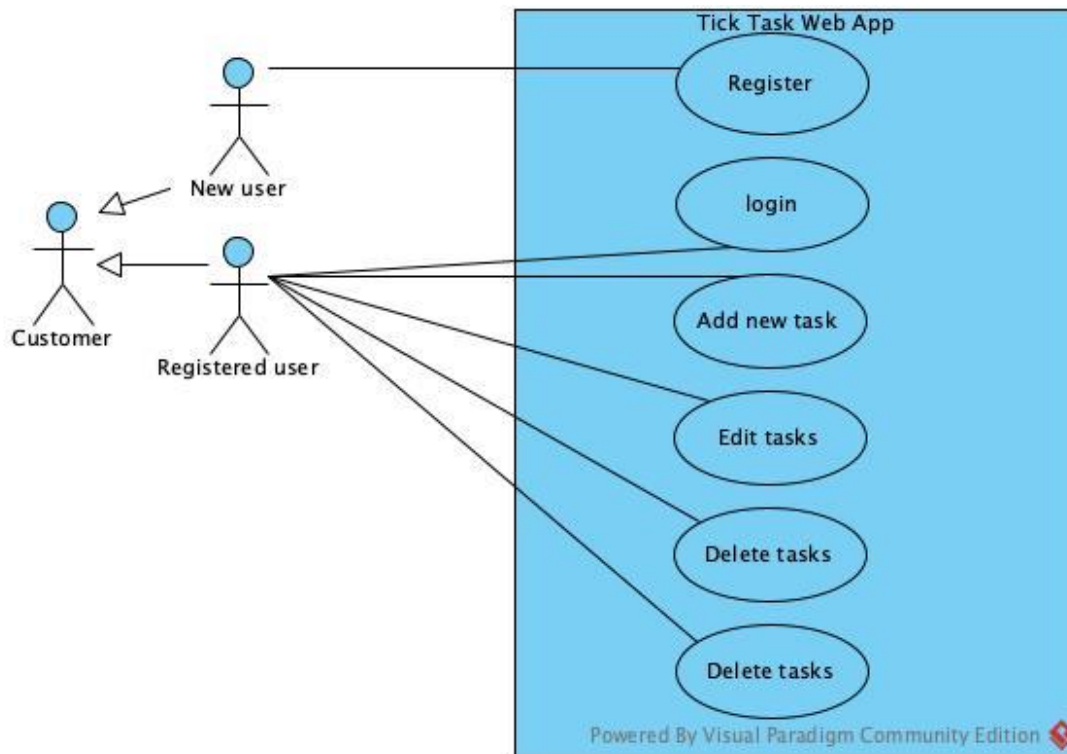


Figure 3.2.A - Use Case Diagram for Tick Task software

3.2.1 Requirement 1 – Register

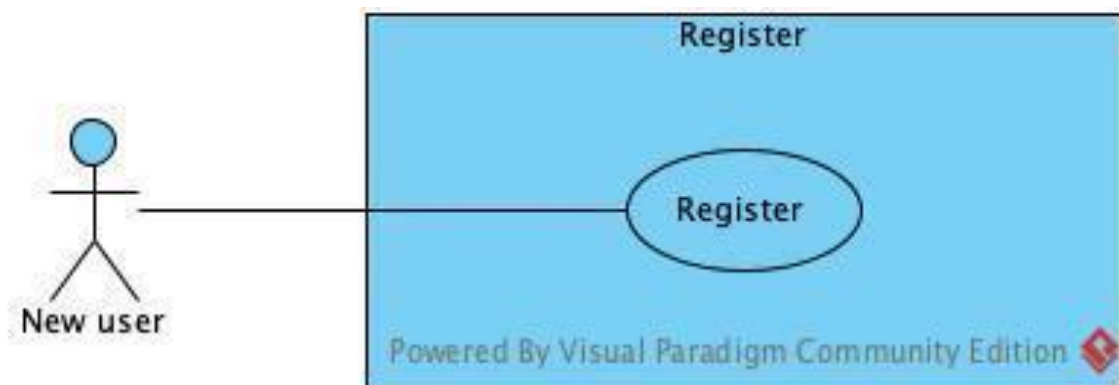


Figure 3.2.B - Register use case

Description & Priority

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

Use Case (Register)

Scope

Registration for the software

Description

User needs to register before being allowed to login and input or access data (tasks).

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 already have an account yet.
 - 2) The system will show the registration pages for the user to fill in personal details.
 - 3) The user fills in all the details required and clicks **register** button.
 - 4) The system will register the user details.

- *Alternate flow:*

- A1:**
- 1) Input an existing registered email address (login credentials).
 - 2) The user inputs an email that is already registered in the system.
 - 3) The system will ask the user to check the details again.
 - 4) The main flow will continue from the **Main Flow 3**.

- *Termination:*

The system displays a message that the registration has been successful.

3.2.2 Requirement 2 – Login

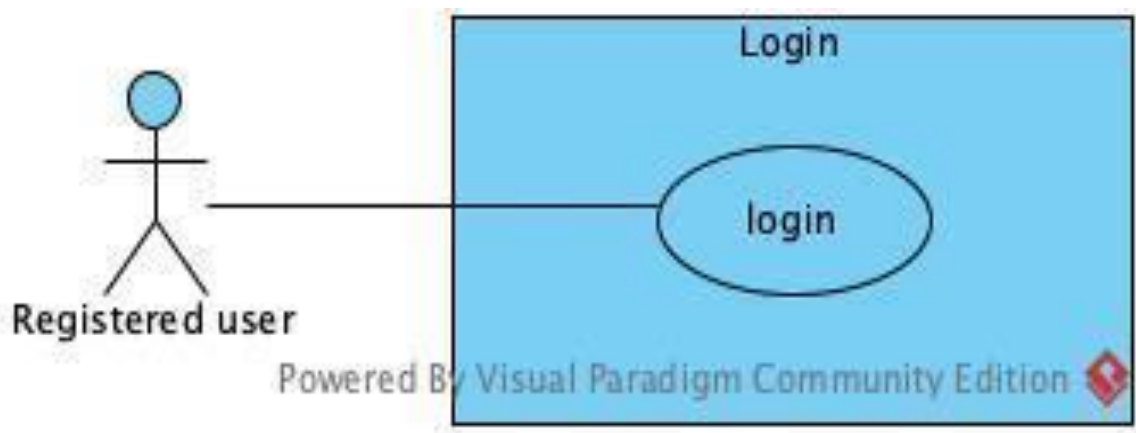


Figure 3.2.C – Login use case

Description & Priority

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

Use Case (Login)

Scope

The user needs to login to be able to access their account or input data.

Description

A registered user logs in to their account to access their data and use the software.

Flow Description

- *Precondition:*

The user must be already registered previously and have valid login details.

- *Activation:*

This use case starts when the user clicks on the **login** button on the login page.

- *Main flow:*

- 1) The user inputs login details and clicks the **login** button.
- 2) The system validates the details and signs the user in, granting access to the software.

- *Alternate flow:*

- A1 :**
- 1) The user inputs incorrect login details
 - 2) The system prompts the user that the details are incorrect and requests for the user to input the correct details again.
 - 3) The user inputs the new details.
 - 4) The use case continues the **Main Flow** from **3.2**.

- *Exceptional flow:*

- E1 :**
- 1) User forgot their login details
 - 2) The user enters incorrect login details.
 - 3) The system prompts the user that the details are incorrect.
 - 4) The user inputs incorrect details again, preventing the user from logging in to their account.
 - 5) System will ask user to reset their password.

- *Termination:*

The system presents the tasks page with all of the users data that has previously been inputted.

3.2.3 Requirement 3 - Add task

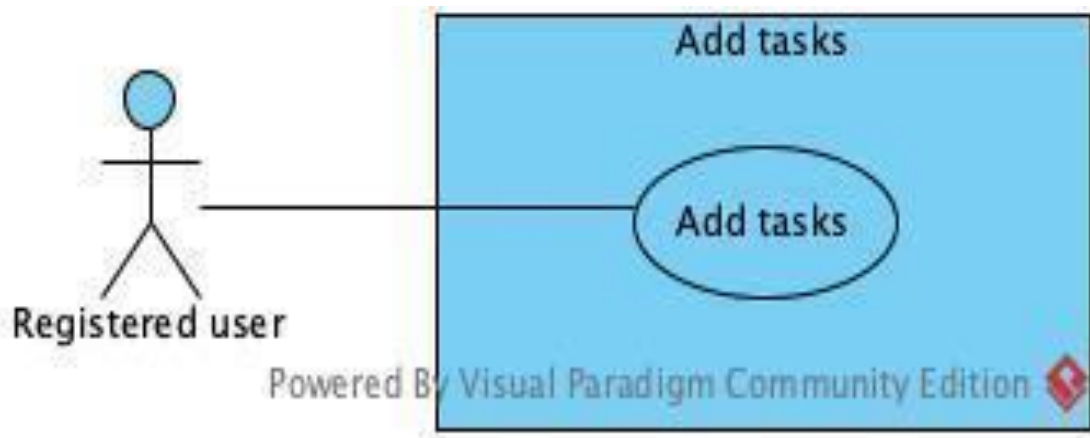


Figure 3.2.D - Add task use case

Description & Priority

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

Use Case (Add Task)

Scope

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

Description

When the user first registers and logs in, the user will need to input tasks in the software.

Flow Description

- *Precondition:*

User must be logged in in order to add any tasks.

- *Activation:*

This use case starts when and the user logs in 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 and navigates the user back to the task page.

3.2.4 Requirement 4 - View all tasks

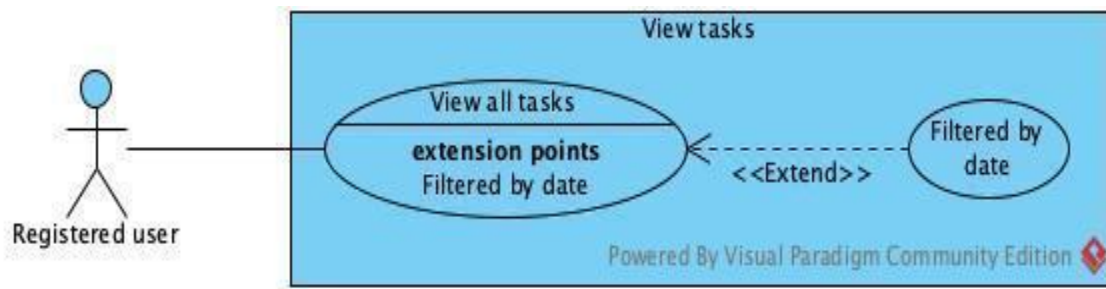


Figure 3.2.E - View tasks use case

Description & Priority

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

Use Case (View All Tasks)

Scope

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

Description

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

Flow Description

- *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 that the user logged in to their account and displays all of their tasks.

2) User adds a new task and the task shows up on the list.

- *Termination:*

The system displays all tasks including the new added tasks.

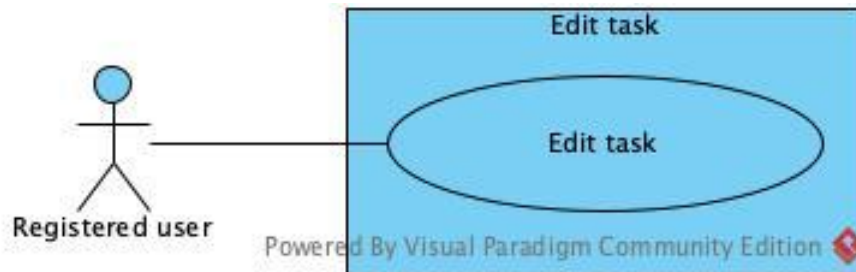
3.2.5 Requirement 5 - Edit task

Figure 3.2.F - Edit task use case

Description & Priority

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

Use Case (Edit Task)

Scope

The scope of this use case is that the user wants to edit a previously inputted 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 and list with the updated task information.

- *Postcondition:*

The task will be updated and will redirect back to the updated task list page

3.2.6 Requirement 6 – Delete task

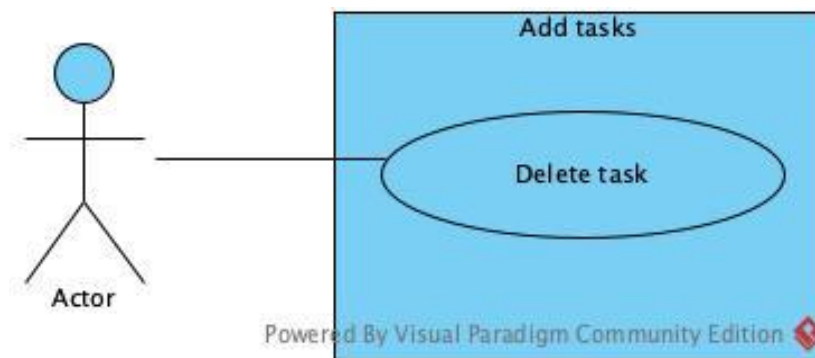


Figure 3.2.G - Edit task use case

Description & Priority

This requirement is the 6th priority. Any tasks that the user does not longer want on the tasks list, they can have the option to delete the task.

Use Case (delete Task)

Scope

The scope of this use case is that the user wants to delete a previously inputted task.

Description

This use case describes the function of when the user wants to delete 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 **delete** button.

- *Main flow:*

- 1) User clicks on **delete** button.

- 2) System updates deletes the task with the updated list of tasks.

- *Postcondition:*

The task list will be updated and will redirect back task page

3.3 Non-Functional Requirements

The non-functional requirements describe Tick Task's software quality attributes below;

3.3.1 Performance/Response time requirement

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

3.3.2 Availability requirement

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

3.3.3 Recover requirement

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

3.3.4 Robustness requirement

Before the website will be deployed or any other new functionality incorporated, different types of testing will be done, to thoroughly troubleshoot any issues, and to make sure no errors will be injected which may cause a system failure. Any errors that may occur during use will be addressed by error handling, that will be implemented to prevent any failures that may occur from user's input, or use of the software.

3.3.5 Security requirement

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

3.3.6 Reliability requirement

User input validation, error handling and system testing will be done regularly to help prevent system failure. Deploying software to the cloud and backing up all data will also help prevent any data loss.

3.3.7 Maintainability requirement

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

3.3.8 Portability requirement

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

3.3.9 Extendibility requirement

Due to time limitation of this project, the software will not be open to public at project completion, however, it will be designed to allow additions to functionality in the future, and, will be 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 register, as per the below ***Figure 4.1.A.***

The **register** page layout will consist of a few details, asking for user's details that are required for login, as per the below image ***Figure 4.1.B.***

Once the user has logged in, the **task** page will show all the other functionalities of the software, such as add tasks, view tasks, edit tasks, or search for any existing tasks, as per below image ***Figure 4.1.C.***

Tick Task	Login
<h2>Login</h2>	
<input type="text"/>	
<input type="password"/>	
<input type="button" value="Login"/>	
<p>Not registered yet? Create an account here</p>	

Figure 4.1.A - Login page

Tick Task	Login
<h2>Register</h2>	
First Name	
<input type="text"/>	
Last Name	
<input type="text"/>	
Email address	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="button" value="Sign up"/>	
Already have an account? Sign in here	

Figure 4.1.B - Register page

Tick Task			Welcome Jean
Create task		Edit task	
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.C - Task page

5 System Architecture

A class diagram **Figure 5.1.A** 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.

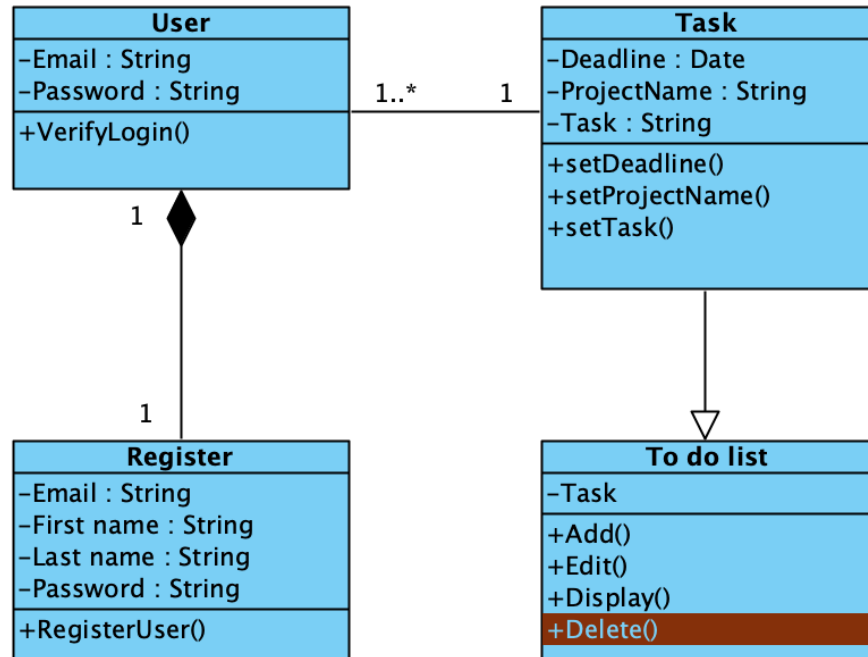


Figure 5.1.A - Class diagram for Tick Task software application

6 System Evolution

More functionalities will be added over time such as;

- Additional functions for tasks, such as search for tasks.
- Ability to set a reminder for when the tasks are due.
- Calculate total tasks achieved.
- Sync with user's email and calendar for meetings and appointments.