

# **RemindMe: A To-Do List Website**

## Software Requirements Specification Document

In partial fulfillment of  
the requirements for  
CMSC 128 Lab: Software Engineering 1

Submitted by  
Johanna Marie Quitoviera  
BS in Computer Science III

Submitted to  
Mr. Leonardo D. Pabroquez

## Table of Contents

1.	<b>Introduction</b>	1
1.1.	Purpose	1
1.2.	Intended Audience	1
1.3.	Intended Use	1
1.4.	Scope	1
1.5.	Definitions and Acronyms	2
2.	<b>Overall Description</b>	2
2.1.	User Needs	2
2.2.	Assumptions and Dependencies	3
3.	<b>System Features and Requirements</b>	3
3.1.	Functional Requirements	3
3.1.1.	<i>Register a new user in the database</i>	3
3.1.2.	<i>Login to the correct account of the user</i>	3
3.1.3.	<i>Navigate between the dashboard page and to-do page</i>	4
3.1.4.	<i>The dashboard page gives an overview of the tasks</i>	4
3.1.5.	<i>The to-do list page</i>	4
3.1.6.	<i>The logout button should properly log out the user</i>	4
3.2.	External Interface Requirements	4
3.2.1.	<i>User Interface</i>	4
3.2.2.	<i>Hardware Interface</i>	6
3.2.3.	<i>Software Interface</i>	7
3.2.4.	<i>Communication Interface</i>	7
3.3.	System Features	7
3.3.1.	<i>Accuracy</i>	7
3.3.2.	<i>Interoperability</i>	7
3.4.	Nonfunctional Requirements	7
3.4.1.	<i>Performance</i>	7
3.4.2.	<i>Usability</i>	8
3.4.3.	<i>Security</i>	8
3.4.4.	<i>Availability</i>	8

## 1. Introduction

This section will include the purpose, intended audience, intended use, scope, and definitions and acronyms that will be stated in this document. Furthermore, it will give a glimpse and outline of the project.

### 1.1. Purpose

The purpose of this document is to give a detailed description of RemindMe—a to-do list website. This website will help keep track of the users' tasks that they need to do for the day. Moreover, it will describe the overview and purpose of the project as well as the functions, interfaces of the website, features of the system, and system constraints under which it must operate. This document is intended for both the clients and developers of the website.

### 1.2. Intended Audience

The intended audience of this document are the clients, developers, project managers, and stakeholders. The users or clients of this website can be anyone who wants to use the website as a to-do list to keep track of tasks and this is accessible to all.

### 1.3. Intended Use

The intended use of this document is to organize and present the requirements of the website, what the website is supposed to do and how it will be expected to perform in order to address the user's needs. This document will serve as an overview for the clients, developers, and stakeholders.

### 1.4. Scope

The "RemindMe" is a website that intends to help the users to organize, keep track, and remind them of their tasks for today. The users can add the tasks that they need to do and delete the tasks that have already been completed. This will serve as a to-do list for the user but rather than writing it on actual paper, the website will store and show the user's pending tasks for today. In short, the website will do the task of storing and keeping track of the user's tasks which is

Having a handy to-do list, especially for students, will help increase productivity, help motivate the user, improve memory, and create order in one's life. The pile of chunks that seem overwhelming to do will be more manageable and help keep the user grounded on the things that are needed to be done. Moreover, the website is designed to give convenience to the users in their daily lives, accessible, and good storage for the user's tasks to store them in one place.

## 1.5. Definitions and Acronyms

Term	Definition
Clients	The users who will use the website.
Developers	Someone or a group of people who will build and maintain the website.
Stakeholders	Any person with an interest in the project who is not a developer.
To-do List	A list of tasks that are needed to be done.
User	Someone who interacts with the website.
User experience	How the user interacts with the system and overall experience.
User interface	The way for the user to interact with the system.
Website	A collection of interlinked Web pages that share a single domain name.

## 2. Overall Description

This section will include the user needs and assumptions and dependencies of the project. Furthermore, It will give an overview and description of the project.

### 2.1. User Needs

The primary users of the website will come from various backgrounds and social differences. Anyone can use the website with the goal of organizing and managing their tasks for the day in order to make the tasks for the day be bearable and manageable for the users. By using the website, the users will be able to register and log in to their personal account, add their tasks, delete their

completed tasks, and know the number of pending tasks that needs to be completed. Furthermore, users would want a to-do list website that is easy, accessible, intuitive, and convenient to use. Therefore, the website will be designed according to the needs of the users.

## 2.2. Assumptions and Dependencies

An assumption about the website is that a device should be connected to the internet and the browser supports HTML, CSS, and Javascript in order for the user to access and use the functionalities and features of the website to its full capacity. On the other hand, the website will be based on a previous project and will be reusing the components and features. But new enhancements will be done in order to improve the user experience of the existing website.

## 3. System Features and Requirements

This section will include the functional requirements, external interface requirements, system features, and nonfunctional requirements of the project. It will give a detailed description of the website and all its features and functionalities.

### 3.1. Functional Requirements

The following are the functional requirements of the website:

#### 3.1.1. *Register a new user in the database*

A user should be able to enter their information in the details field. Moreover, the website must be able to register or save the user's details to the database in order for them to join the community as a new user. The register or signup page should be able to check if the given email of the user already exists in the database. Also, to check and inform the user of their entered password's validity. The account password should contain and meet the conditions of a strong password to have a more secured account.

#### 3.1.2. *Login to the correct account of the user*

The website must be able to log in the user to their correct account according to the details entered. After the user details exist in the database, it should be able to redirect the user to the dashboard or

home page of the website. Else, the user shall be informed that they should register first.

**3.1.3. *Navigate between the dashboard page and to-do page***

The user should be able to navigate and switch between the pages with ease. The website should be able to redirect to the right pages once a specific page button has been clicked.

**3.1.4. *The Dashboard page gives an overview of the tasks***

The website must be able to show the pending tasks from yesterday and current tasks for the day. Also, it should be able to show and inform the user of the number of pending tasks that are needed to be done.

**3.1.5. *The To-do list page***

The user must be able to access and add their tasks, delete their completed tasks, and have the option to clear all their tasks. On the other hand, the website must be able to store the added task/s, show the current tasks and the number of pending tasks, and remove the tasks deleted by the user.

**3.1.6. *The logout button should properly log out the user***

The user must be able to log out their account from the system when they click the logout button. Therefore, the website must be able to end the user session and redirect the user back to the login page.

**3.2. External Interface Requirements**

The following are the external interface requirements of the website:

**3.2.1. *User Interface***

The user interfaces are the features that allow the user to interact with the system. For this project, users will be directed first to the login page of the website, see Figure 1. If the user is new, they should register first before continuing with using the website. This can be done by

clicking on the signup button on the login page. Clicking the button will then redirect the user to the signup page, see Figure 2. The user can now log in to the website after filling out the details. Moreover, users who are already registered in the database of the website can now enter their login details on the login page.

**LOGIN**

---

Email address:  
jpquitoviera@up.edu.ph

Password:  
\*\*\*\*\*

☐ Show Password

**LOG IN**

---

New User? **SIGN UP**

*Figure 1. The Login Page*

**SIGN UP**

---

First Name:  
Johanna Marie

Last Name:  
Quitoviera

Email address:  
jpquitoviera@up.edu.ph

Password:  
\*\*\*\*\*

☐ Show Password

**SIGN UP**

---

Already have an account? **LOGIN**

*Figure 2. The Sign Up Page*

After the successful login, the website should redirect the user to the dashboard page, see Figure 3. On the dashboard page, the user should be able to see the overview of the current tasks of the user. The navigation tabs should be able to navigate between the dashboard and the to-do page. On the other hand, the user should be able to add and delete tasks, see Figure 4. The website should be able to store and remove the tasks according to the user's actions. Lastly, clicking the logout button will end the session and redirect the user to the login page, see Figure 1.

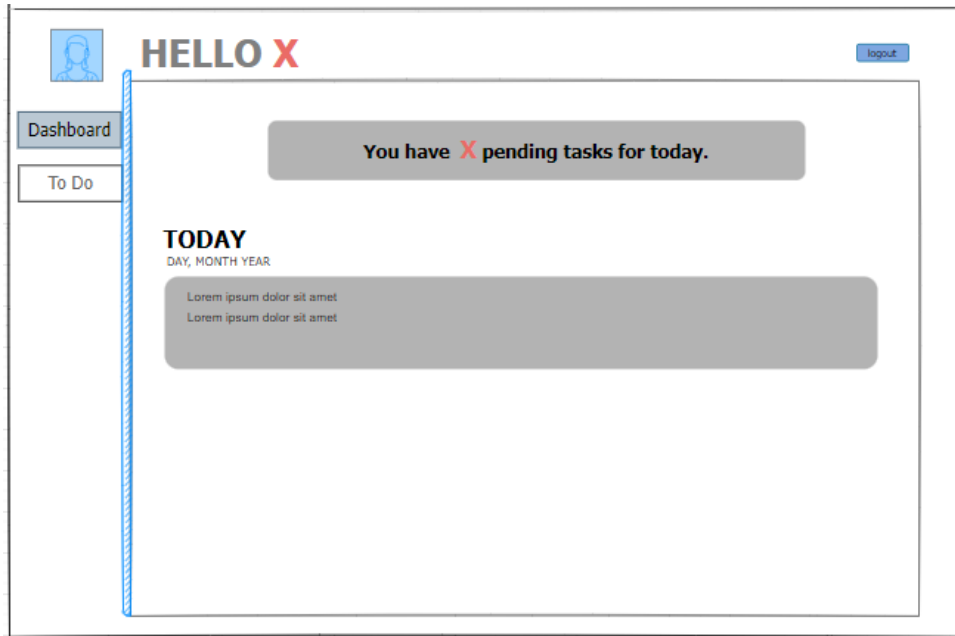


Figure 3. The Dashboard Page

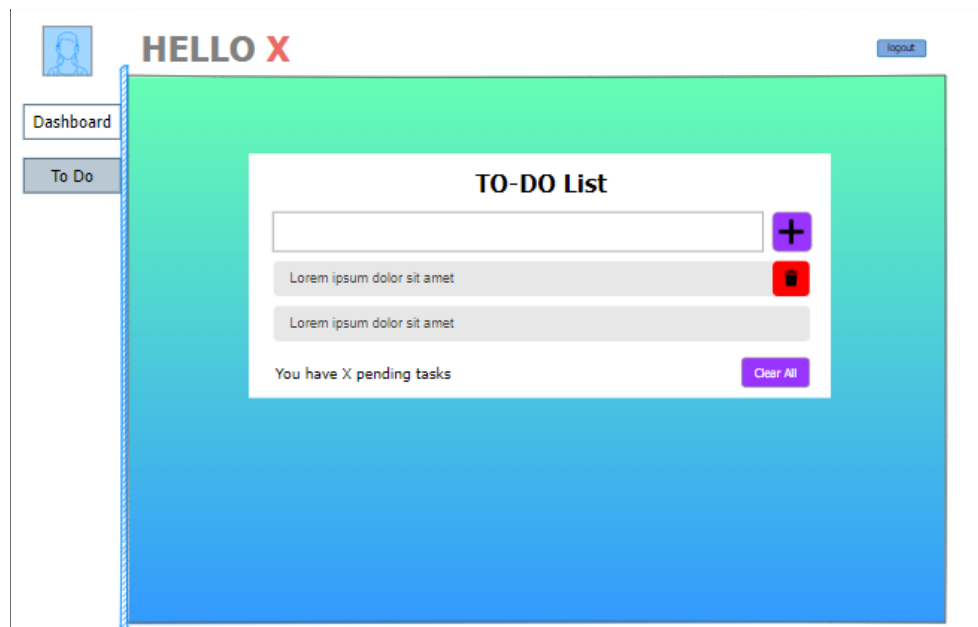


Figure 4. The To-do List Page

### 3.2.2. Hardware Interface

In order to access the website, the device used should be connected to the internet and the browser used to should be able to support HTML, CSS, and JavaScript. Furthermore, the operating system that is preferred to use is Windows for laptops or computers.



#### 3.2.3. *Software Interface*

The database serves as the storage area for the user's information in order for the user to log in to their account. The connection between the website and the database consists of operations concerning both reading and writing data. Furthermore, HTML, CSS, Javascript, and MySQL will be used to build, design, and communicate to the hardware interface.

#### 3.2.4. *Communication Interface*

The website will need an internet connection in order to communicate to the different parts of the interfaces. Also, in order to exchange data between the database and the website itself.

### 3.3. **System Features**

The following are the system features of the website:

#### 3.3.1. *Accuracy*

The website should be able to show the correct data that the user input. It should be consistent throughout the user interaction and what the user entered should also accurately reflect on its output. Users that added a task should be able to see what they entered into the system.

#### 3.3.2. *Interoperability*

The website must be viewable and accessible to all types of browsers (Google Chrome, Firefox, Internet Explorer, etc.). The general experience of the website must be relatively the same all throughout and in all of the browsers.

### 3.4. **Nonfunctional Requirements**

The following are nonfunctional requirements of the website:

#### 3.4.1. *Performance*

The website should be able to respond fast to various user interactions. The current load time of changing or navigating through web pages should be consistent all throughout. Furthermore, the transitions, adding tasks, and removing completed tasks should be quick and seamless in order to improve the user experience.

#### 3.4.2. *Usability*

The users should be able to intuitively navigate and use the website. The design, features, and interface of the website should be intuitive so that it could be easy to learn and understand. Using the to-do list feature should efficiently do the user's actions (adding tasks and deleting completed tasks) with no errors.

#### 3.4.3. *Security*

The website should be able to secure and protect the user's personal and important details entered in the website's database. It should be protected from security threats and should not be leaked outside of the system.

#### 3.4.4. *Availability*

The website should be available at any time and any day as long as the user needs to use the system. Also, it should be available and accessible to use on all browser platforms.