A screenshot of a video game

Description automatically generated with low confidence

**COVID CARE**

**Software Requirement Specification**

Table Of Contents

[I. Overview 3](#_Toc98338925)

[1. Introduction 3](#_Toc98338926)

[2. System Functions 5](#_Toc98338927)

[a. Screen Flow 5](#_Toc98338928)

[b. Screen Details 5](#_Toc98338929)

[c. User Authorization 6](#_Toc98338930)

[d. Non-Screen Functions 6](#_Toc98338931)

[3. Common Requirements 6](#_Toc98338932)

[II. Functional Requirements 7](#_Toc98338933)

[1. Sign in - Sign up 7](#_Toc98338934)

[a. Sign in 7](#_Toc98338935)

[b. Sign out 8](#_Toc98338936)

[c. Sign Up 8](#_Toc98338937)

[2. CRUD 9](#_Toc98338938)

[a. User 9](#_Toc98338939)

[b. News 10](#_Toc98338940)

[3. Password Encryption 11](#_Toc98338941)

[4. Reset Password 11](#_Toc98338942)

[5. Email Confirmation 11](#_Toc98338943)

[6. Messenger 11](#_Toc98338944)

[7. Set timetable 13](#_Toc98338945)

# Overview

## Introduction

To reduce the pressure on the frontline healthcare workforce, the government now encourages Covid patients with mild or no symptoms and no underlying diseases to self-quarantine with the help of health stations and facilities at commune level. However, it would be hard for these patients and their family or friends to complete the care process and find support from medical forces. Besides that, health stations at the commune level would also find it difficult to connect and manage all the patients in their area without the help of technology.

**Covid Care** is a web platform providing free direct connection among patients, their family or friends and medical force as well as people who experience quarantine at home or even who are concerned about epidemic situations to get or supply health care services, advice and news.

With advanced features, the platform gives you the best healthcare experience by providing latest news, instant messages, high performance UI contact and steady speed.

Diagram

Description automatically generated

* 1. Product’s context

## System Functions

### Screen Flow

Graphical user interface, diagram, Teams

Description automatically generated

2-1. Screen FLow

### Screen Details

|  |  |  |
| --- | --- | --- |
| Feature | Screen | Description |
| Login | Login | Provide UI for login to system |
| Forgot Password | Login | Change password if forgotten |
| Sign Up | Sign up page | Provide UI for sign up |
| Manage User | Admin page | Manage user |
| Manage News | Admin page | Full CRUD’s permission to news |
| Show all News | Home page | List all news in the system |
| News Detail | User page  (Guest/ Patient) | News Detail |
| Chat/ Messenger | User page  (Patient/Hospital) | Chat |
| Edit profile | User page  (Patient/Hospital) | Edit user’s profile |
| Create timetable | User page  (Patient/Hospital) | Create timetable |
| Manage patient status | User page  (Hospital) | Update patient status |

### User Authorization

|  |  |  |  |
| --- | --- | --- | --- |
| Screen | Administrator | Member | Guest |
| Homepage | x | x | x |
| >> Login/Sign up |  |  | X |
| >> Finding and read news |  | x | X |
| >> Tools | x |  |  |
| UserPage |  | x |  |
| >> Edit profile |  | x |  |
| >> Manage own timetable |  | x |  |
| Management Page | x |  |  |
| >> Manage Members | x |  |  |
| >> Manage Moderators | x |  |  |
| >> Analyze Statistics | X |  |  |
| >> CRUD News | x |  |  |

### Non-Screen Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | Encrypt Password | Encrypt Password | Decrypt password by MD5 library |
| 2 | Authorization | Security | Use JWT to create a token for the login user and use it to determine if the user has permission to access api |
| 3 | Verify mail | Sign up | Use Spring mail to verify the mail of the user before enabling the account |

## Common Requirements

* Validation input:

+ Check valid user info before creating or modifying

+ Use pattern for password

* Check duplicate: duplicate news, user info

# Functional Requirements

## 1. Sign in - Sign up

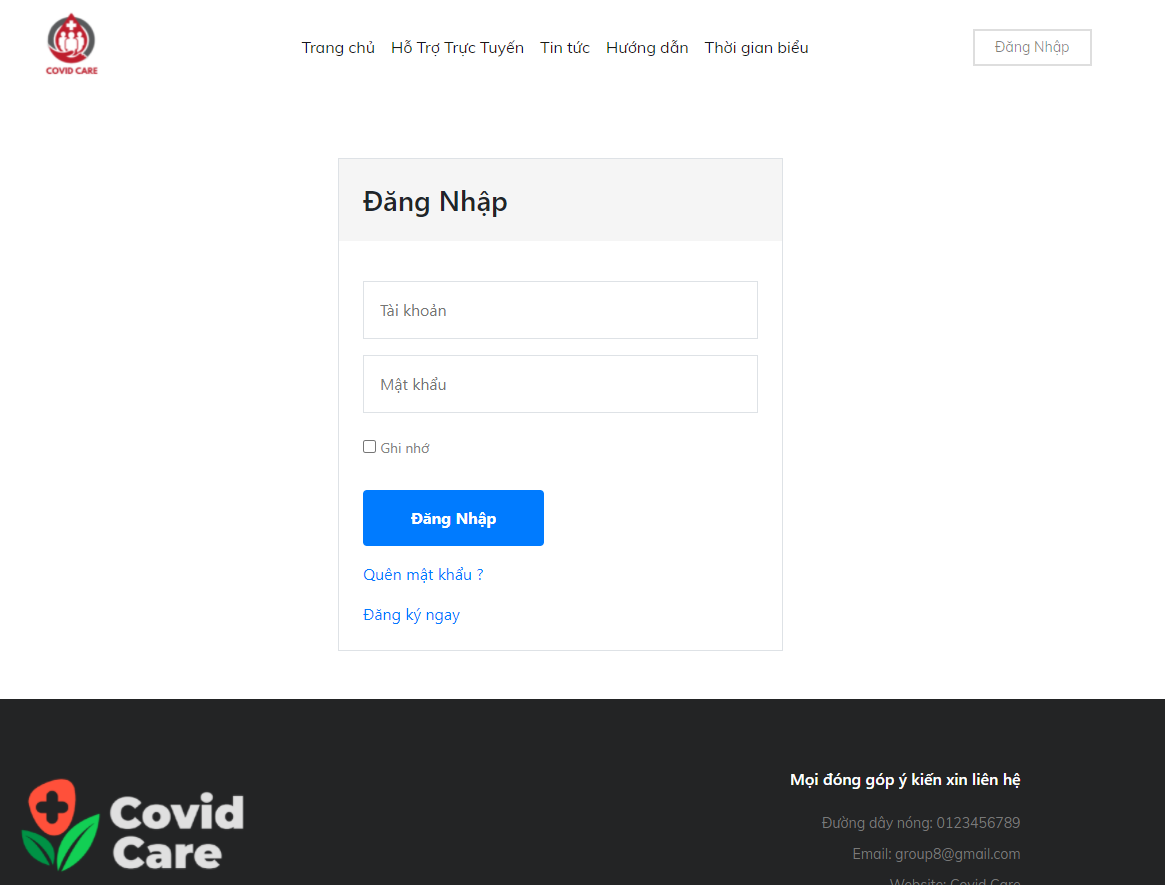
### a. Sign in

* *Function Trigger*: After access to homepage, clicking on a button for Login in the upper right corner (appear when not login) will redirect user to Login page
* *Description*:

+ Actor: User

+ Purpose: Function allows user to sign in with their Covid Care’s account

* *Screen*:



* *Function details*:

+ With username and password that user enter, the system will check database that if username exist or account is enabled:

If not: the system will notice the user for wrong username or for register

If existing: checking the password of this username in the database, and forwarding to the homepage with user information and token if the password is correct or notice for wrong password

+ Use a token to save user information and authorization

### b. Sign out

* *Function Trigger*: After access to homepage, clicking on a User avatar in the upper right corner will expand a menu for user setting, click to Log out
* *Description*:

+ Actor: User who already logged in

+ Purpose: Function allows user to log out of their accounts

* *Screen*:
* *Function details*:

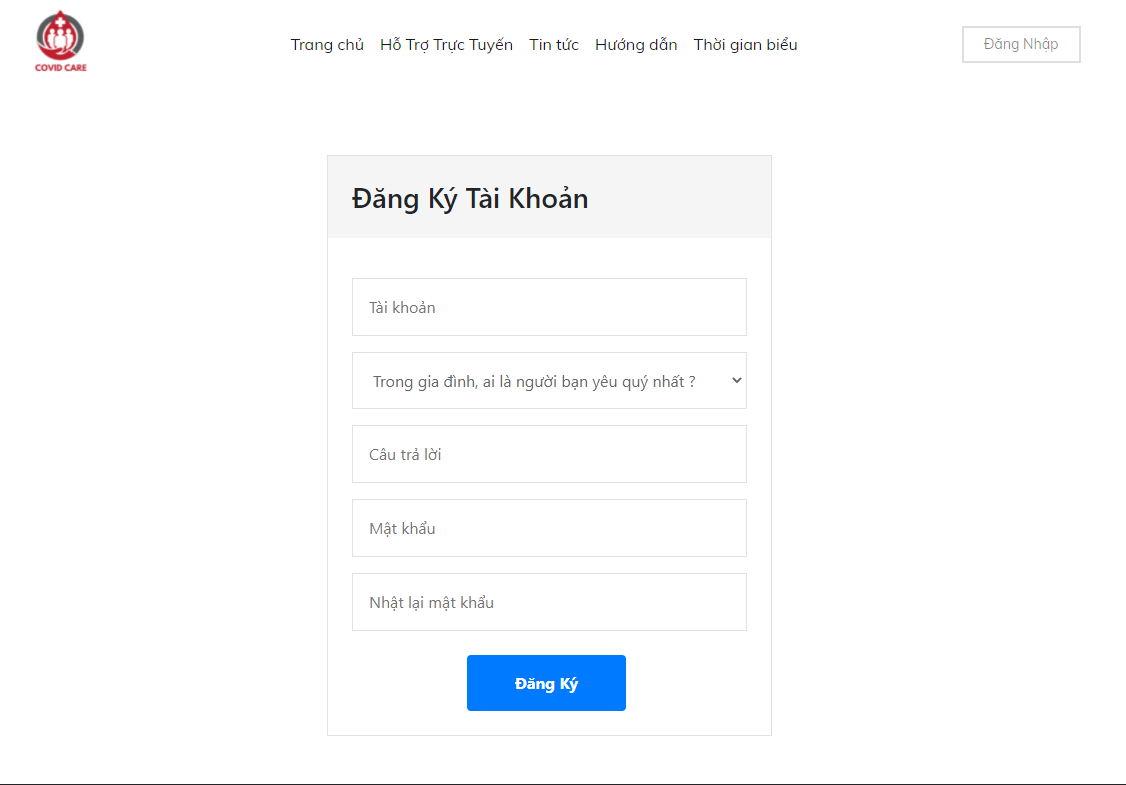
+ Remove current token

### c. Sign Up

* *Function Trigger*: After access to the homepage, clicking on a button for Sign up in the upper right corner (appears when not login) will redirect the user to the Sign-up page. Or click on Sign up for Covid Care in Login page
* *Description*:

+ Actor: User who want to create an account

+ Purpose: Function allows user to create their new Covid Care’s Account

* *Screen*:
* 
* *Function details*:

+ Sign up page will ask for necessary information of user (username, question, password) and create new account base on that information (after validating all input)

## 2. CRUD

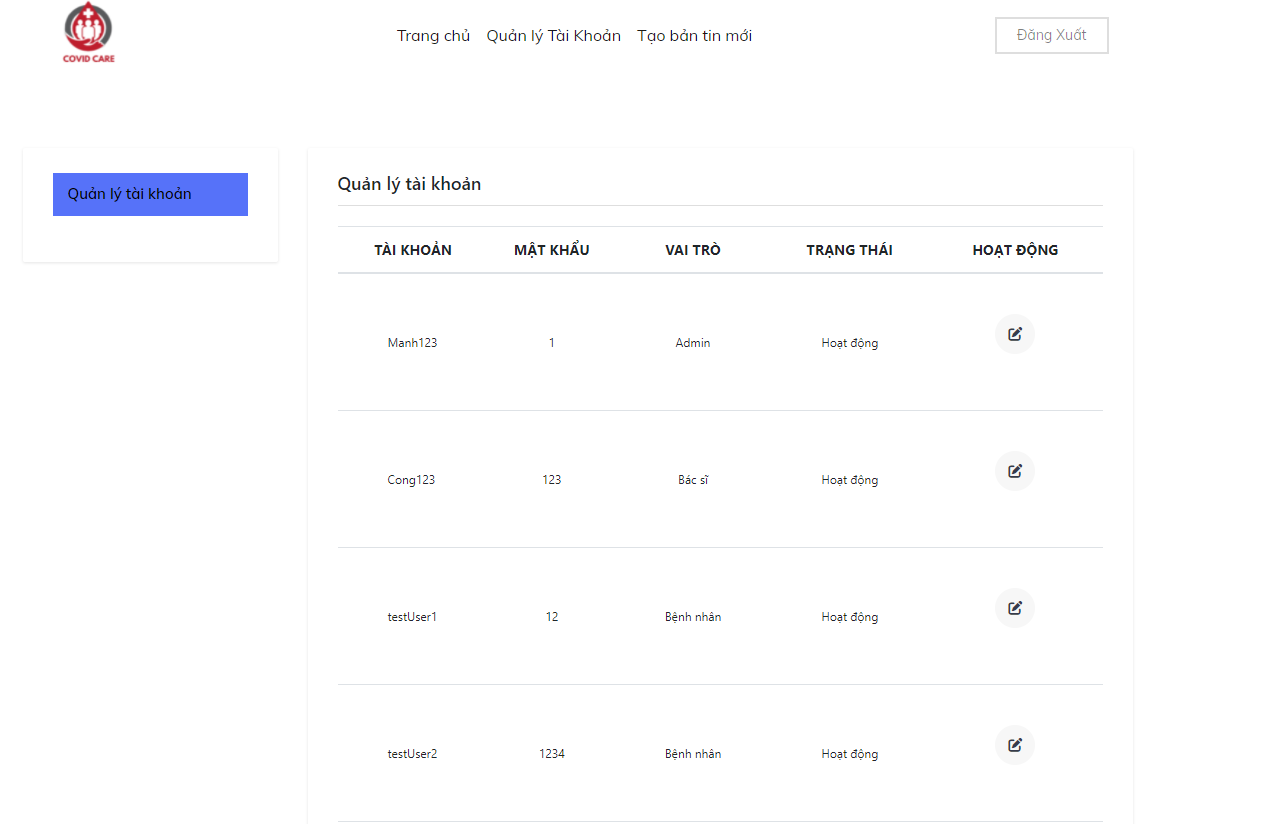
* System will validate all input before CRUD

### User

* *Function Trigger*: After login as admin or moderator and have access to the Management Page, clicking on the Member Management bar will direct to the CRUD Member Page
* *Description*:

+ Actor: Moderator for member accounts and Administrator for all users

+ Purpose: Manage and set roles for users

* *Screen*:
* 
* *Function details*:

+ Create: Admin and moderator can create a new user or a guest can register a new account in the Sign up page

+ Read: Retrieve information about user(s) from the database and display user(s) on the Management page

+ Update: Members can edit their own information, admin and moderator have access to modify all members, admin can provide roles to other users

+ Delete: Remove a user, moderator cannot remove admin

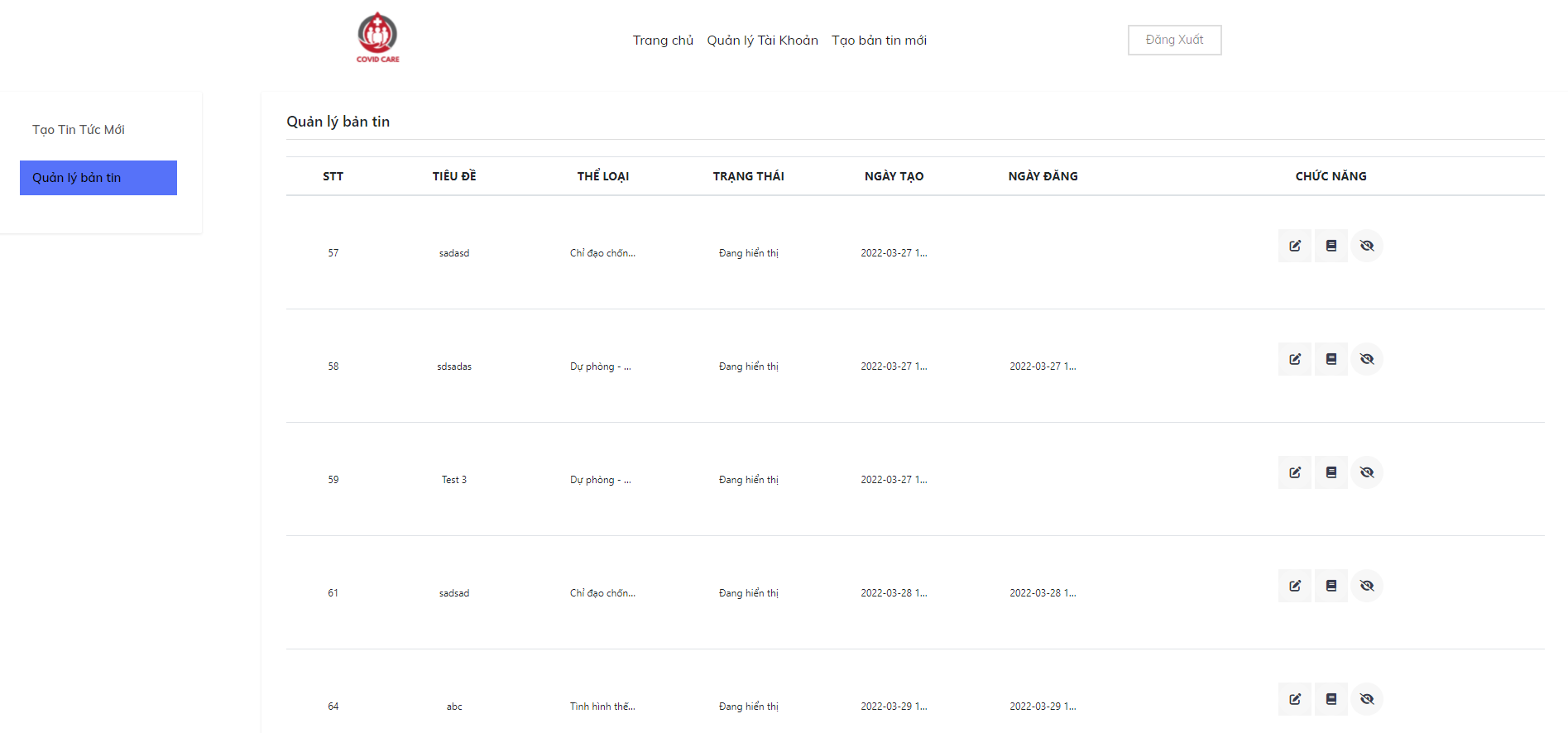
### News

* *Function Trigger*: After login as admin or moderator and have access to the Management Page, clicking on the News Management bar will direct to the CRUD News Page
* *Description*:

+ Actor: Moderator and Administrator for all news

+ Purpose: Manage news

* *Screen*:



* *Function details*:

+ Create: Use API to get what users input for their news and then the server creates new news in the database based on that information

+ Read: Retrieve all information about the news from the database and put them to the website by using API

+ Update: Change the information of a piece of news on the website and send the id to the server through an API, after that the server updates new information in the database. The new change will be displayed on the screen

+ Delete: Delete the news by providing the id to the server through an API, the server uses the id to delete the right piece of news in the database. After deleting the news, display the new list on the screen

## Password Encryption

* *Function Trigger*: After signing up successfully, the password will be encrypted then will be saved in the database.
* *Description*:

+ Actor: All users (admin, …)

+ Purpose: Increase the security for all users

* *Detail*: The server will use a the MD5 library to encrypt the password

## Reset Password

* *Function Trigger*: There are 2 ways to reset password. The first way users can reset in the profile page and the second way is on the login page
* *Description:*

+ Actor: All users (admin, ...)

+ Purpose: Reset password for users if the users forgot passwords

* Function details: When users click on reset password, the page will be redirected to other pages and requires users input email address, then the server will send an email to the users and verify. Users click the URL on the mail and redirect users to change passwords.

## Post vaccinate

* *Function trigger*: After logging in, the hospital can post the injection schedule and confirm the injection application from the patient, and the patient can view the injection schedule information and register for the injection.
* *Description*:

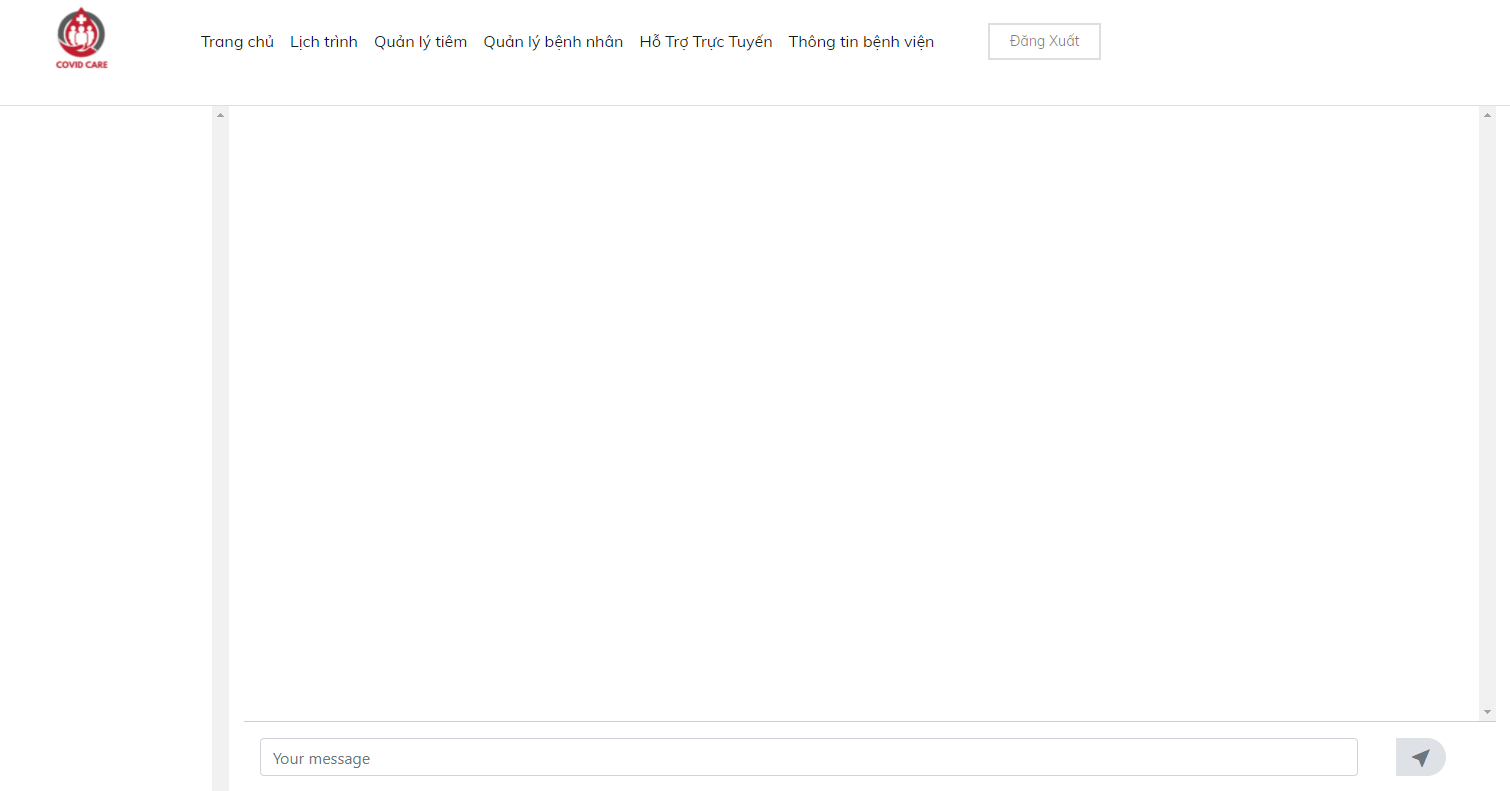
+ Actor: Patient/Hospital

+ Purpose: Information and registrations about injection schedules.

* *Screen*:

## Messenger

* *Function trigger*: When a patient clicks on the messenger button on their homepage, they can chat with the hospital that is responsible for taking care of them. In other way, the hospital can access to another Messenger screen that allows them to see all the patients that are currently taking care of.
* *Description*:
* Actor: Patient/ Hospital
* Purpose: exchange messages between patient and hospital
* *Screen*:



*messenger view*

* *Function details:*

+ When patient click on chat bubble on their site, they will connect to their nearest hospital in their area to get support; when they change their location, they will get connected to a new hospital with the medical record transferred from the former hospital

+ Hospital can view list of patients in their area

+ Patients and hospitals can chat to each other

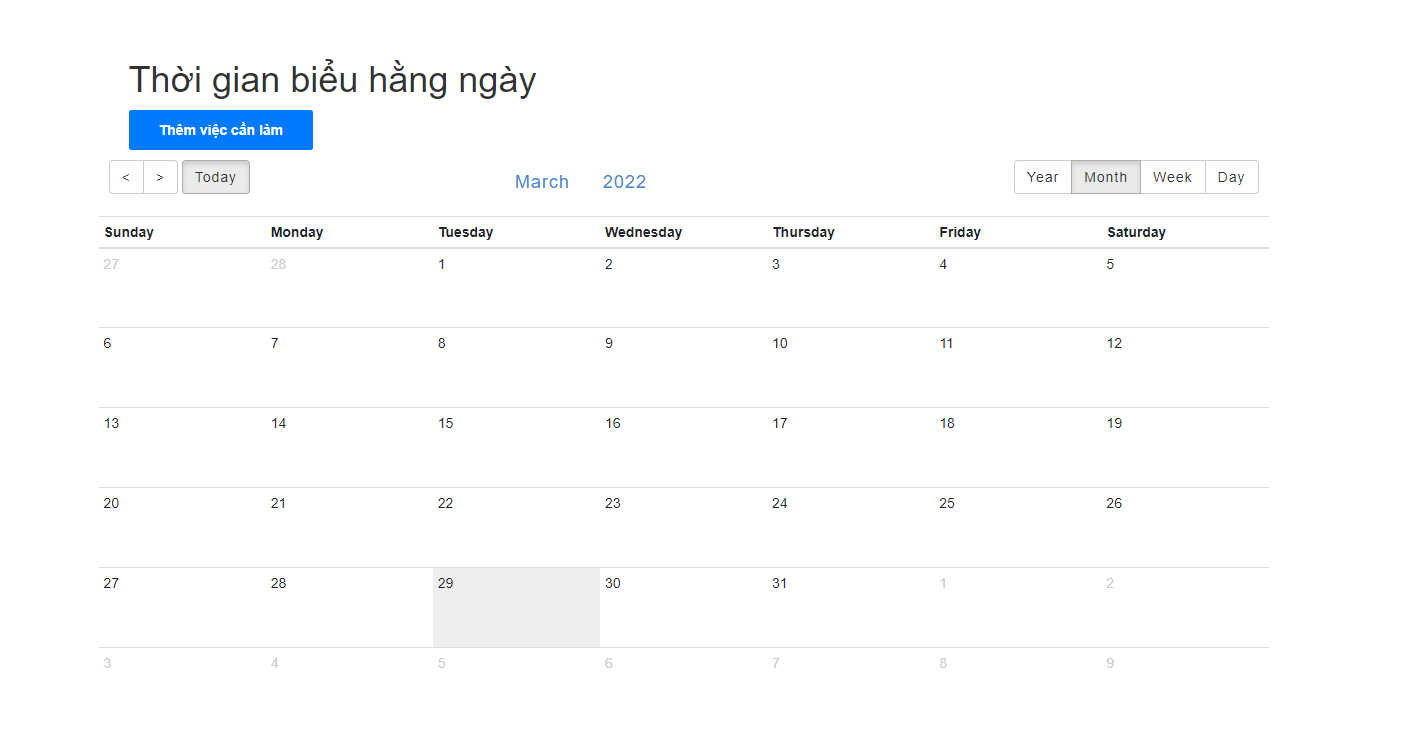
## Set timetable

* *Function trigger*: After login, patient or hospital are allowed to access to their timetable
* *Description*:

+ When a patient or hospital clicks on the timetable button on their site, they will redirect to their timetable that reminds their daily tasks

+ They can CRUD tasks on their timetable. Hospital can CRUD tasks to both them and their patients

* *Screen*:



* *Function detail*:

+ Create: Use API to get what users input for their tasks and then the server creates new tasks in the database based on that information

+ Read: Retrieve all information about the tasks from the database and put them to the website by using API

+ Update: Change the information of a tasks on the website and send the id to the server through an API, after that the server updates new information in the database. The new change will be displayed on the screen

+ Delete: Delete the tasks by providing the id to the server through an API, the server uses the id to delete the right tasks in the database. After deleting the news, display the task list on the screen