Оглавление

[Введение 2](#_Toc69133658)

[3. Разработка структуры приложения 2](#_Toc69133659)

[3.1. Требования к предоставлению приложения 2](#_Toc69133660)

[3.2. Описание компонентов и модулей системы 2](#_Toc69133661)

[4. Руководство пользователя. 3](#_Toc69133662)

[4.1 Назначение программы 3](#_Toc69133663)

[4.2. Описание интерфейса. 3](#_Toc69133664)

[5. Руководство программиста. 7](#_Toc69133665)

[5.1. Организация ввода данных в программу и вывода результатов 7](#_Toc69133666)

[5.2. Структура сайта 7](#_Toc69133667)

[5.3. Описание компонентов и модулей системы 8](#_Toc69133668)

[Приложение 20](#_Toc69133669)

# Введение

# 3. Разработка структуры приложения

### 3.1. Требования к предоставлению приложения

Веб-приложение будет состоять из страниц: авторизация и регистрация, главной страница с турами, страница с чатом, страница с уведомлениями, страница редактирования туров, страница редактирования заявок.

### 3.2. Описание компонентов и модулей системы

Приложение будет состоять из следующих модулей: модуль авторизации и модуль основной части. Модуль авторизации включает в себя компоненты регистрации и логин. Модуль основной части включает в себя все остальные необходимые страницы и еще страница выбора между менеджерами постов и заявок. Папка shared хранит компоненты, использующиеся в нескольких компонентах, какие-то по несколько раз, такие как header, footer, пост и сообщение. Папка models хранит модели пользователя, поста, сообщения, заявки. Папка services хранит сервисы системы. В качестве БД выступает файл db.json, с которым будут взаимодействовать сервисы и прописанные в них методы.

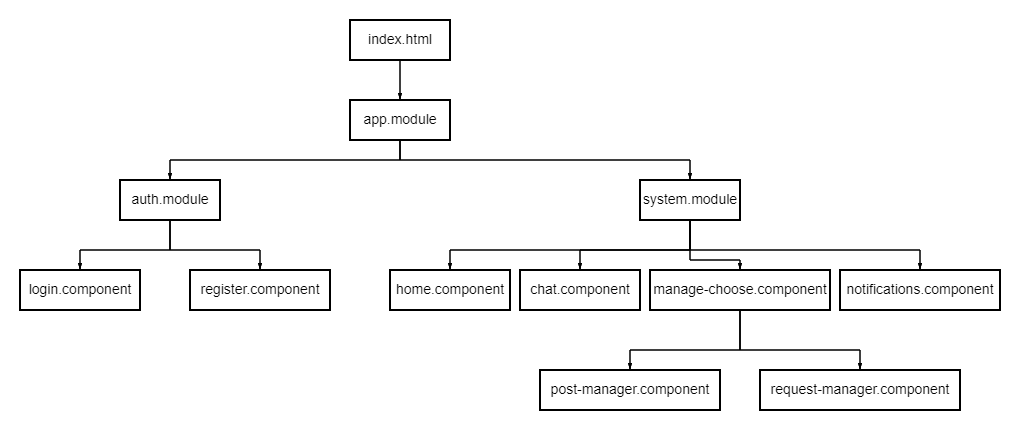


Рис.1 – Описание модулей приложения

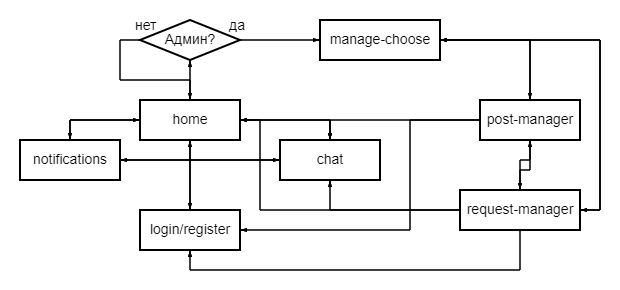


Рис.2 – Описание переходов приложения

# 4. Руководство пользователя.

### 4.1 Назначение программы

Веб - приложение предназначено для переписки между пользователями. Можно отправлять любому зарегистрированному пользователю любое сообщение и он сможет его увидеть, а также увидеть кто его отправил. У каждого пользователя свой профиль, который можно изменить или удалить на странице настроек. Можно добавить/изменить фото, дополнительную информацию о себе. Можно изменить имя, почту и пароль. Есть функция сортировки анкет по алфавиту для более удобного поиска внутри сайта.

### 4.2. Описание интерфейса.

При первом запуске приложения пользователь видит страницу с турами, а также в шапке кнопки «Вход» и «Регистрация».

Чтобы войти в систему и получить возможность записаться на нужные туры, нужно либо зарегистрироваться, либо авторизоваться.

После успешной аутентификации пользователя отправляет обратно на главную страницу, где теперь у туров включились кнопки записи. В шапке появились кнопки: чат с админом, уведомления, имя пользователя, кнопка выхода. Кнопка уведомлений показывает сколько в данный момент их есть у пользователя.

На страницу чата можно обратиться к админу. На странице уведомлений написаны имеющиеся изменения заявок.

У админа страница чата выглядит иначе. Перечислены все пользователи, можно открыть чат с каждым из них. Имеется сортировка пользовательских чатов по количеству сообщений.

Админ имеет возможность редактировать посты и заявки. Для этого созданы отдельные страницы-менеджеры. Выбор одной из них делается на странице ссылок на менеджеры.

Страница редактирования постов позволяет создать новый тур или изменить/удалить существующий.

Страница редактирования заявок позволяет изменить тур в заявке на другой или удалить заявку.

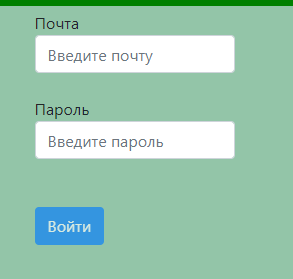


Рис.3 – Страница с формой авторизации

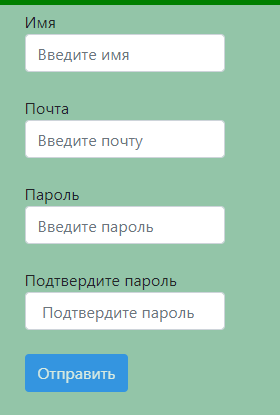


Рис.4 – Страница с формой регистрации

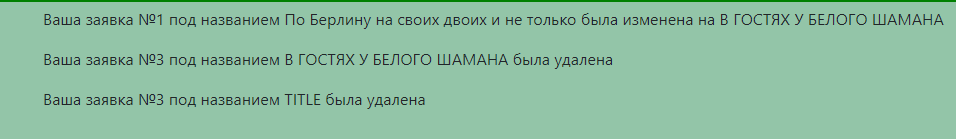


Рис.5 – Страница уведомлений

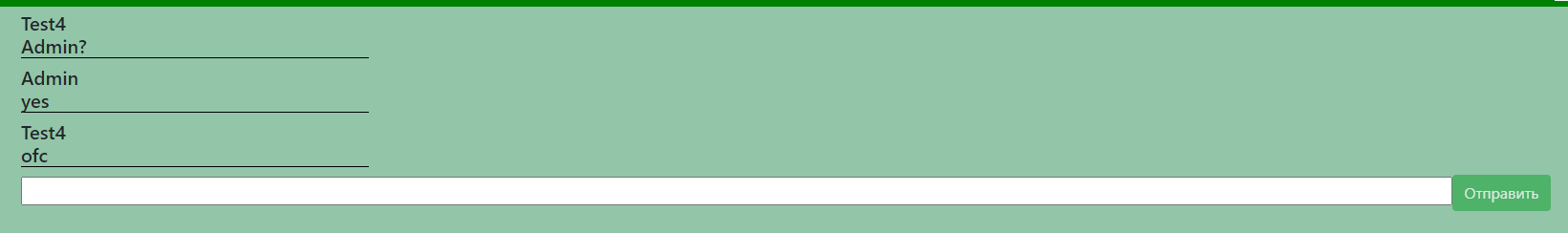
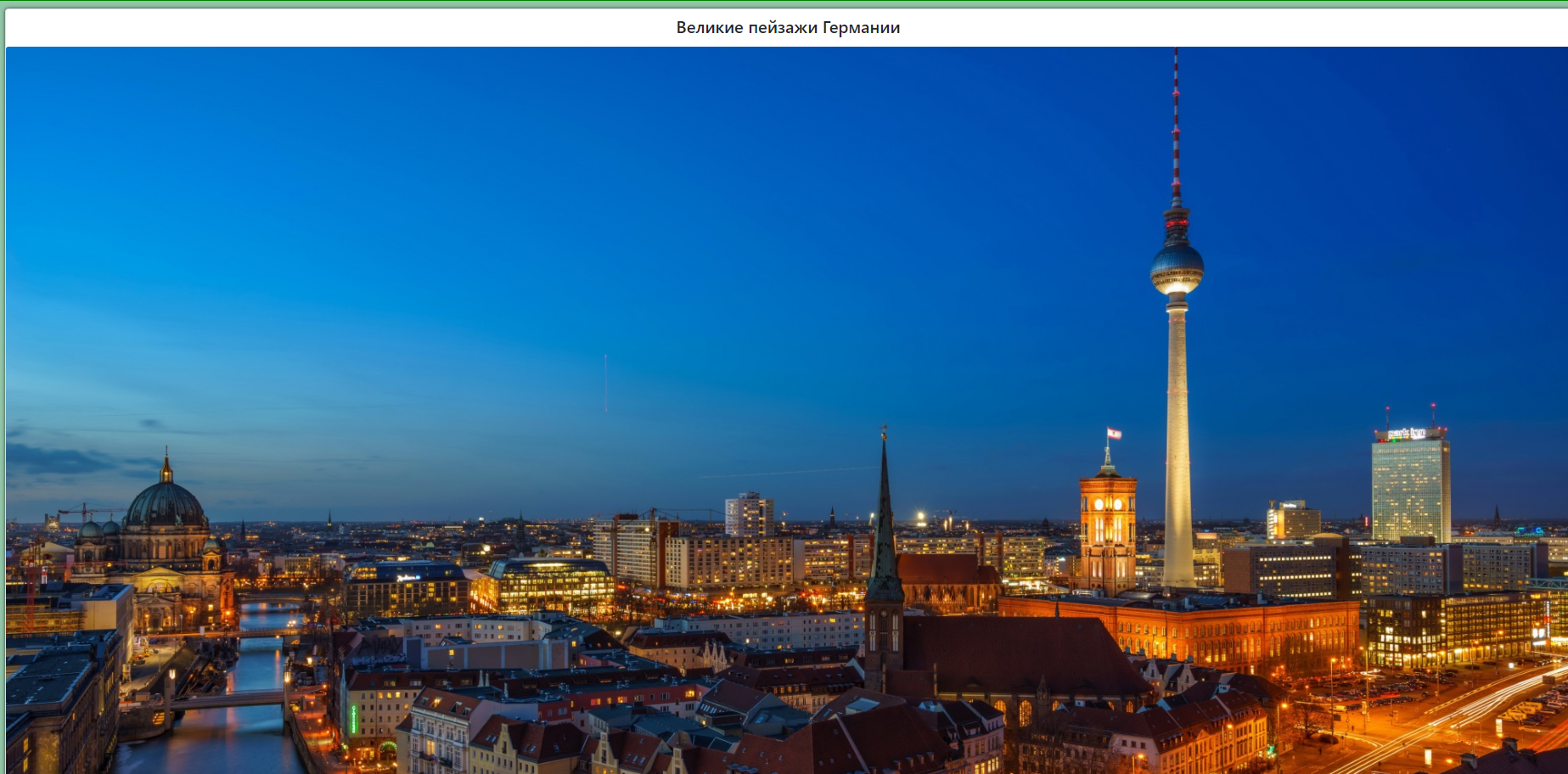


Рис.6 – Чат у пользователя с админом



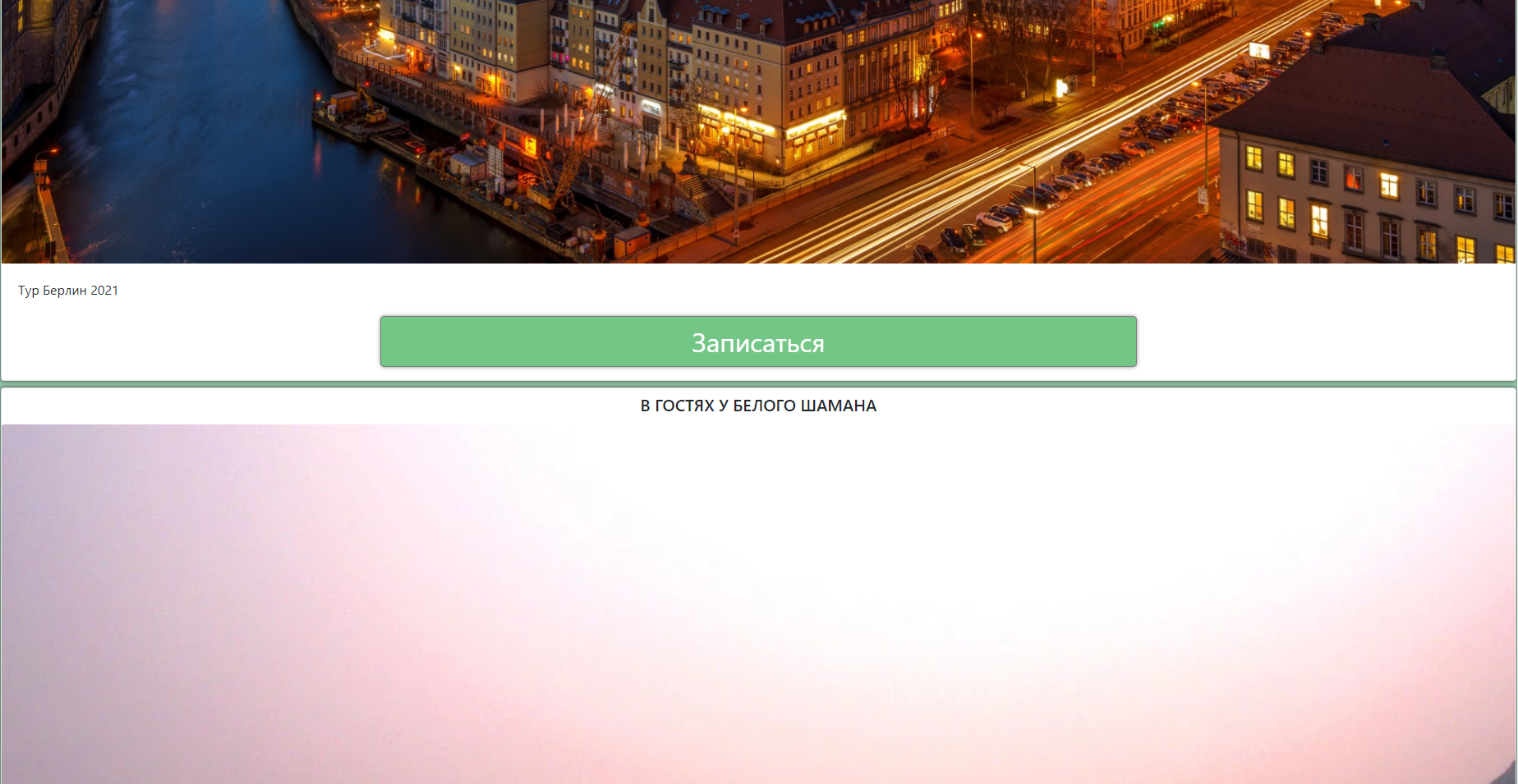


Рис.7 – Главная страница



Рис.8 – Кнопки пользователя

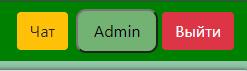


Рис.9 – Кнопки админа

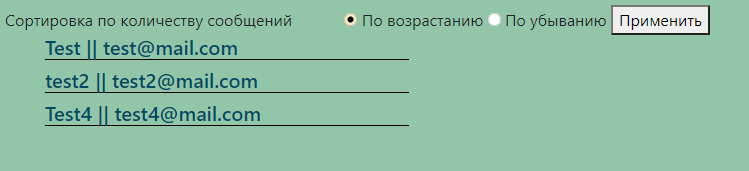


Рис.10 – Чат админа и пользователи

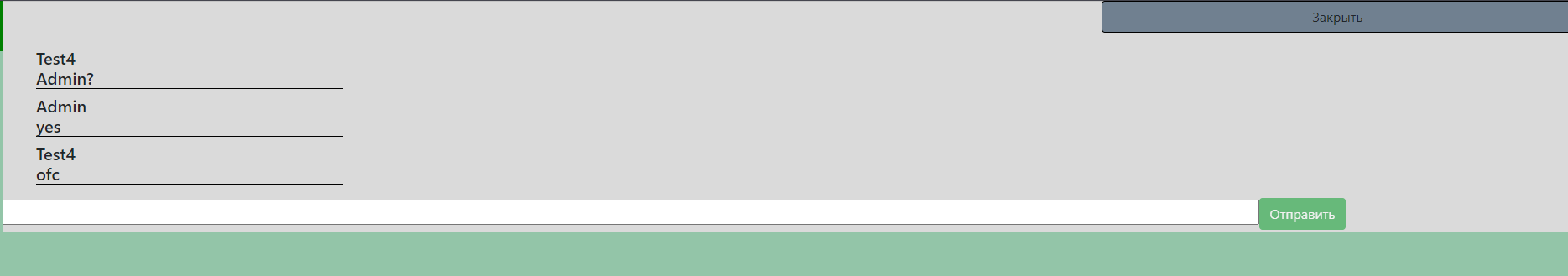


Рис.11 – Чат админа с пользователем

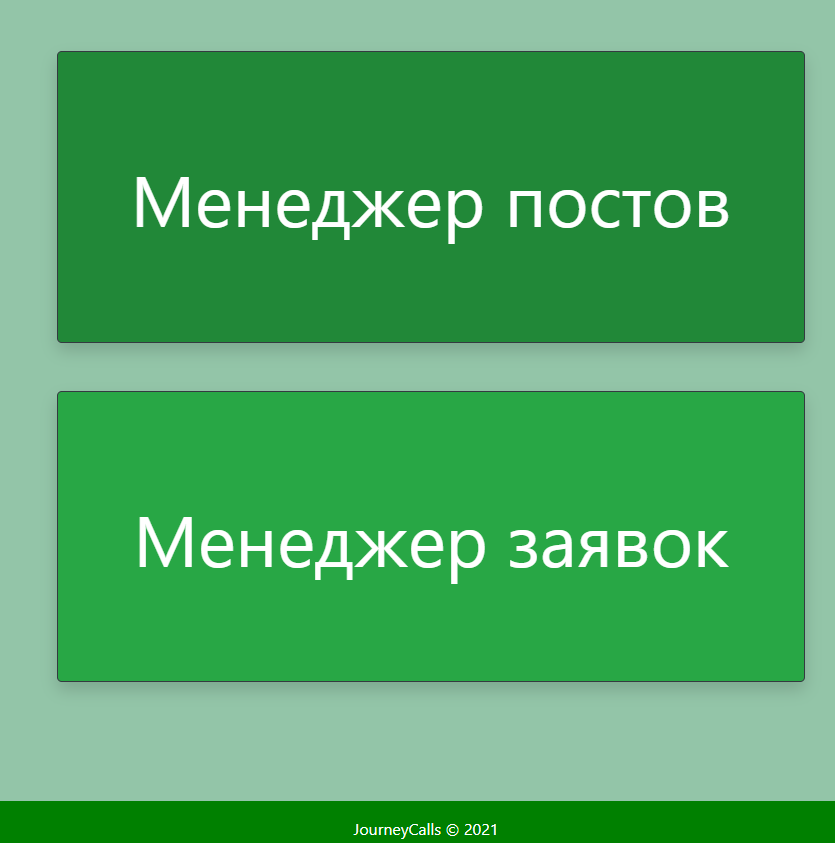


Рис.12 – Страница выбора менеджера

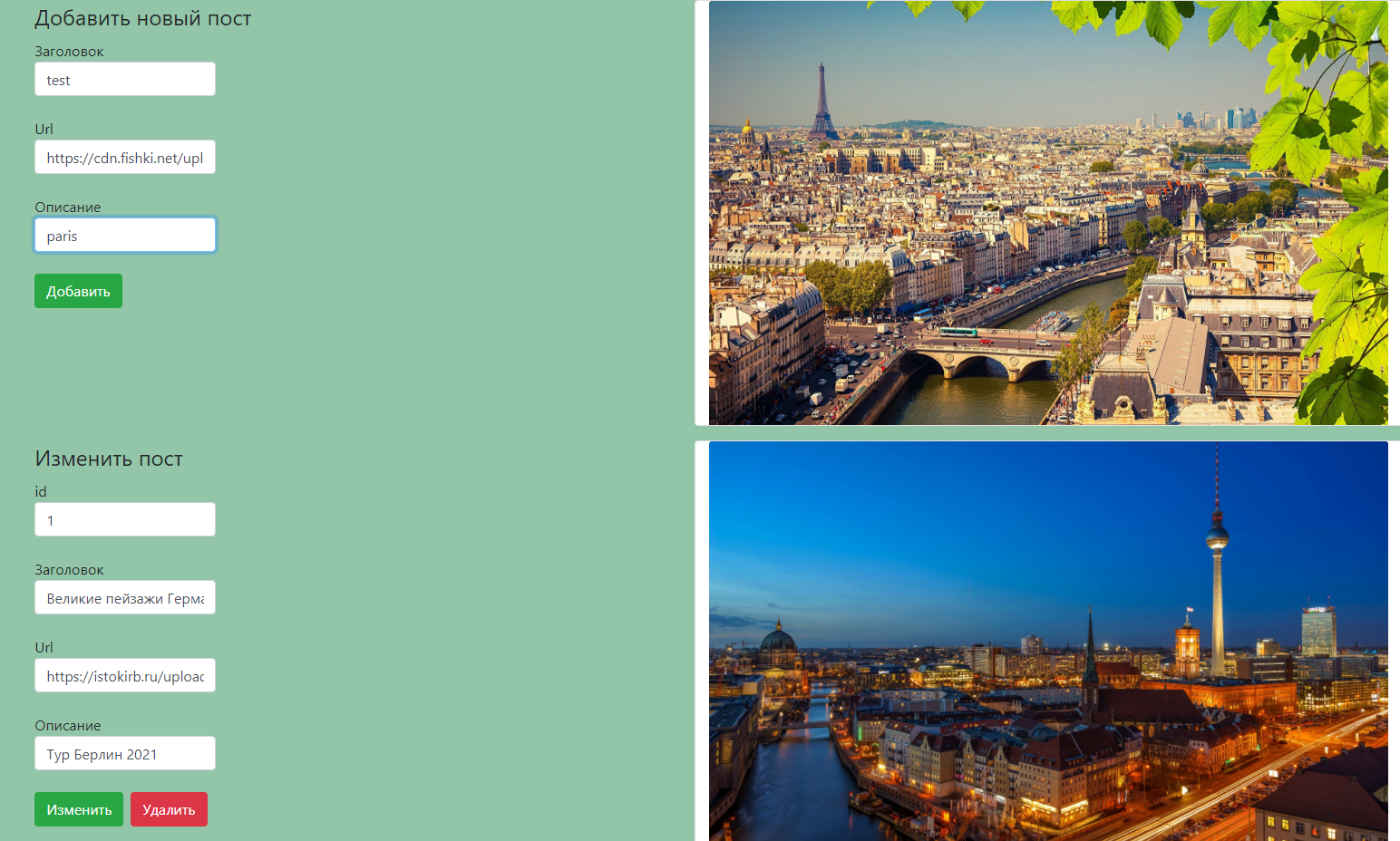


Рис.13 – Страница менеджера постов

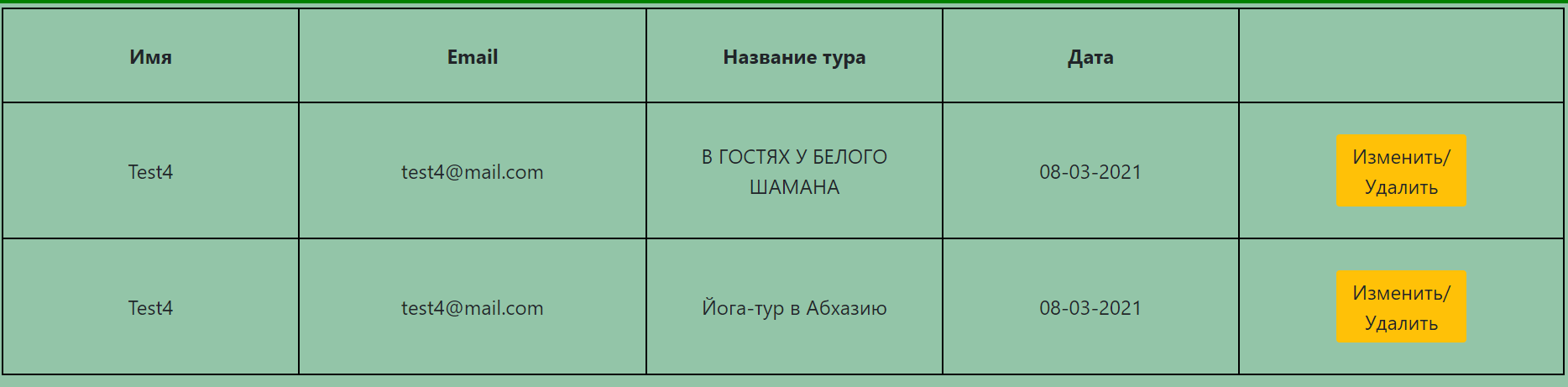


Рис.14 – Страница менеджера заявок

# 5. Руководство программиста.

### 5.1. Организация ввода данных в программу и вывода результатов

Данные в программу поступают от пользователя путем ввода информации на страницах сайта в предназначенные для этого формы.

Вывод информации происходит на страницах сайта в предназначенные для этого блоки.

### 5.2. Структура сайта

Проект состоит из следующих основных директорий – e2e, node\_modules, src.

*е2е* – в ней находятся определённые файлы, отвечающие за тестирование приложения

*node\_modules* – содержит все пакеты и библиотеки, необходимые для работы приложения

*src* - содержит весь исходный код приложения:

• app – содержит директории с модулями и компонентами сайта, а также главные app модули проекта

• assets – в этой папке хранятся всевозможные статические файлы (стили, шрифты, изображения и т.д)

• environments – хранит переменные окружения

*• auth:* В данной директории хранятся компоненты регистрации и авторизации, а также auth.service.

*• shared:* В данной директории хранятся компоненты использующиеся в нескольких компонентах и по несколько раз.

• models: В данной директории хранятся модели системы

• services: В данной директории хранятся сервисы системы

*• system:* В данной директории хранятся основные компоненты сайта:

Страница авторизации(login.component.ts):

На странице авторизации присутствует форма для введения email и пароля от аккаунта пользователя, а также кнопка «Войти» отправляющая данные. Производится обработка формы: проверка на валидацию полей, наличие зарегистрированного пользователя.

Страница регистрации(register.component.ts):

На странице регистрации присутствует форма с несколькими полями для введения данных(имя, почта, пароль) пользователя, а также кнопка «Отправить» отправляющая данные.

Производится обработка формы: проверка на валидацию полей, совпадение пароля с подтверждающим паролем, наличие уже зарегистрированного пользователя(по почте), и занесение в базу нового пользователя.

Главная страница (home.component.ts):

Страница со всеми турами. Пользователи могут записаться на них нажав по кнопке «Записаться». Создастся заявка.

Страница чата(chat.component.ts):

Для пользователя: страница чата с админом.

Для админа: страница чата с пользователями, необходимо выбрать с каким из них открыть чат. Имеется сортировка этих пользователей по количеству сообщений.

Страница оповещений(notifications.component.ts):

Страница с уведомлениями.

Страница выбора менеджера(manage-choose.component.ts):

Страница с выбором одного из менеджеров.

Страница менеджера постов(post-manager.component.ts):

Страница с редактором постов, добавление, изменение и удаление их.

Страница менеджера заявок(request-manager.component.ts):

Страница с редактором заявок, изменение и удаление их. Можно изменить тур в заявке.

Шапка(header.component.ts)

Шапка сайта. Cсылки на другие страницы.

Подвал(footer.component.ts)

Подвал сайта. Копирайт.

Пост(post.component.ts)

Экземпляр поста(тура).

Сообщение(message.component.ts)

Экземпляр сообщения.

### 5.3. Описание компонентов и модулей системы

Для страниц регистрации и авторизации создан отдельный модуль – auth.module. Он будет контейнером для компонентов авторизации и регистрации:

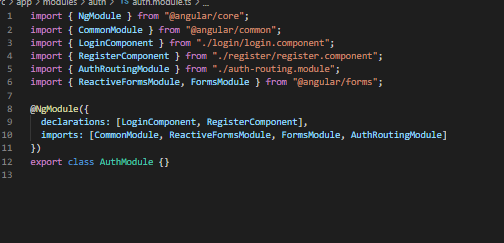


Рис.15 – auth.module.ts

Для перехода между страницами авторизации и регистрации были созданы отдельные роуты в auth-routing.module.ts :



Рис.16 – auth-routing.module.ts

Для компонентов авторизации и регистрации были созданы формы FormGroup с параметрами валидации:

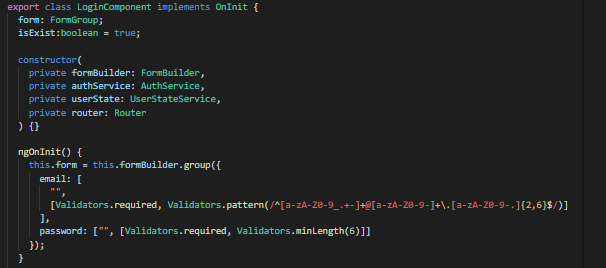


Рис.17 – login.component.ts

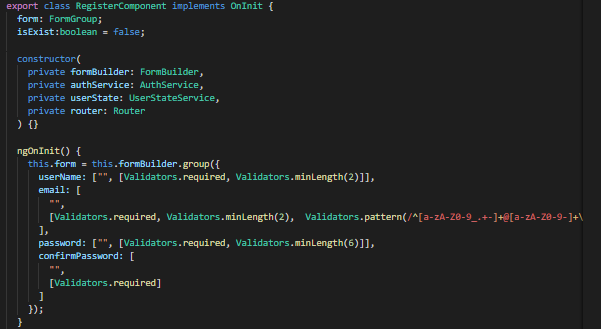


Рис.18 – registration.component.ts

Для работы с данными пользователя из базы данных db.json были созданы отдельная модель user.model.ts и сервис user-state.service.ts:

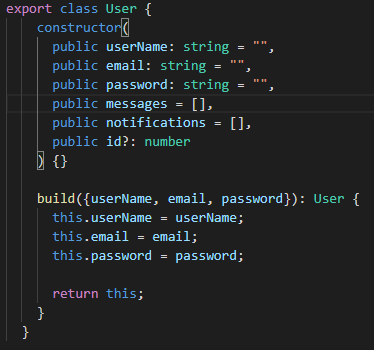


Рис.19 – user.model.ts

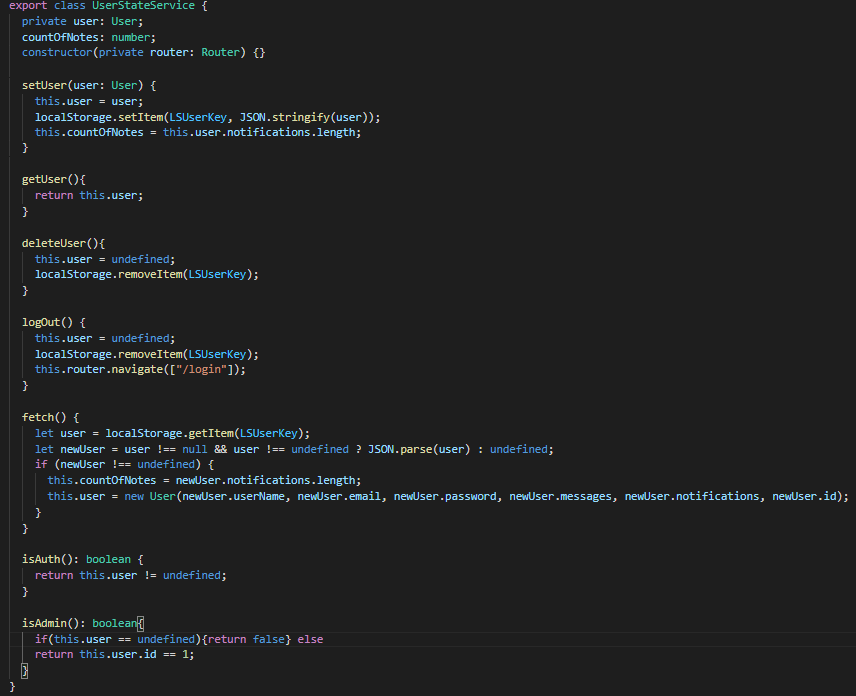


Рис.20 – user-state.service.ts

Для модуля аутентификации был создан отдельный сервис:



Рис.21 – auth.service.ts

Реализация авторизации и регистрации(рисунок 14-15):

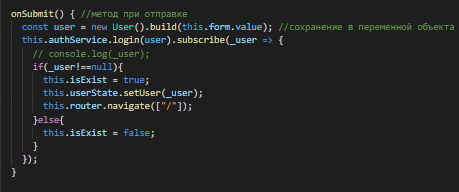


Рис.22 – Реализация в компоненте авторизации



Рис.23 – Реализация в компоненте регистрации

Для работы с сообщениями была создана отдельная модель message и создан отдельный сервис – message.service.ts:



Рис.24 – message.model.ts

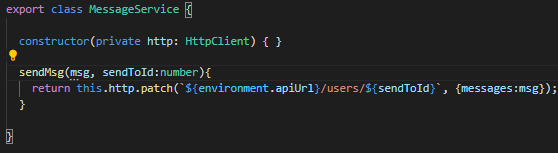


Рис.25 – message.service.ts

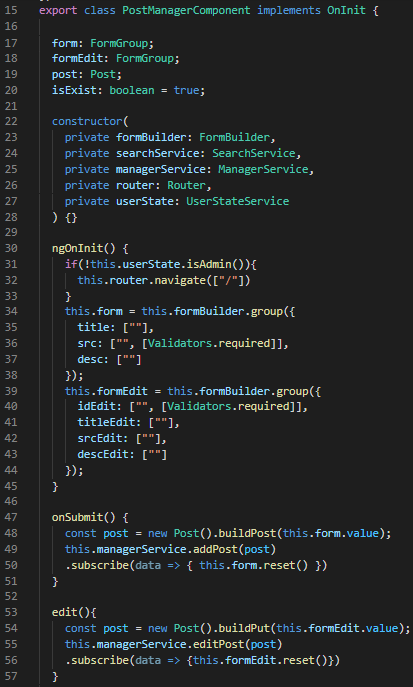
Для изменения постов создана страница менеджера постов(рисунок 18-19)



Рис. 26 – post-manager.component.ts

Для изменения заявок создана страница менеджера заявок

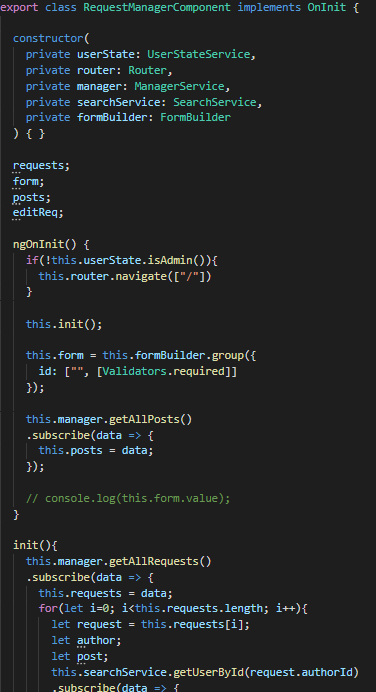
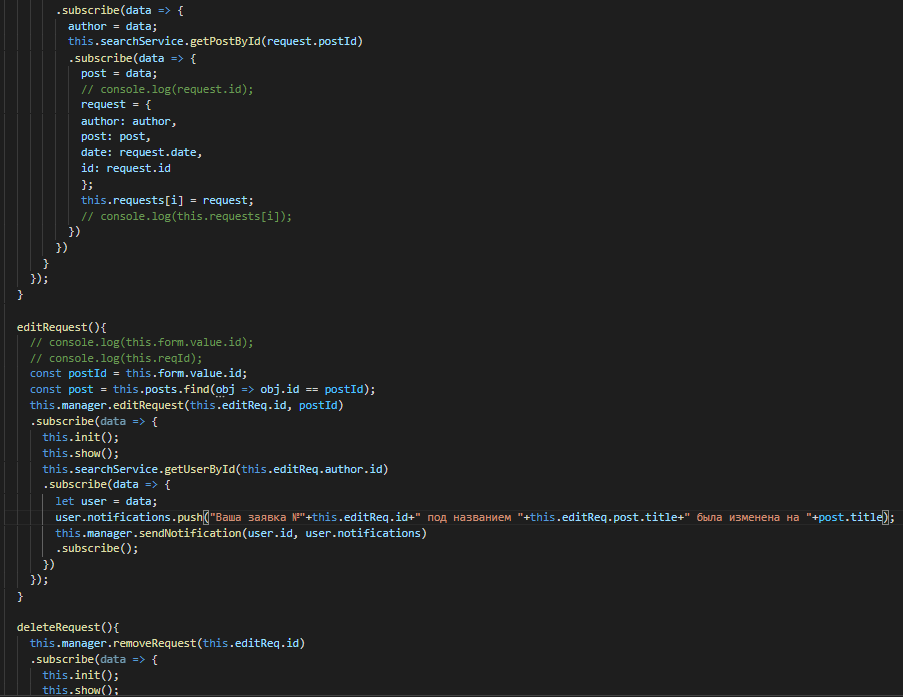
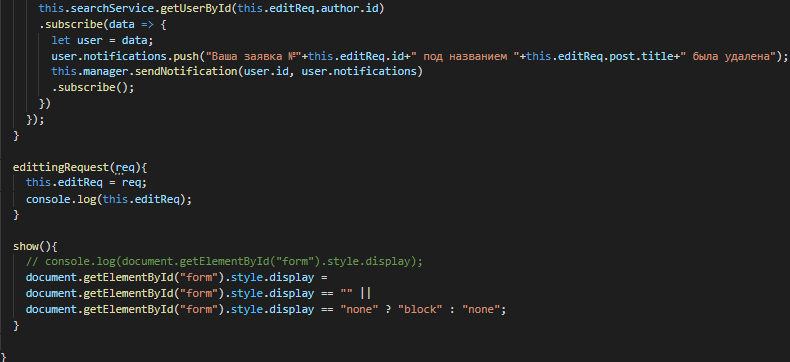
  

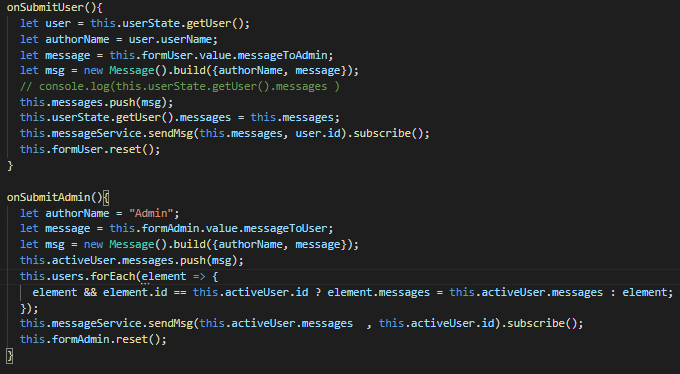
Рис.27 – request-manager.component.ts

Для менеджеров создан общий сервис



Рис. 28 – settings.service.ts

Возможность переписки осуществляется в компоненте чата



Для сортировки данных был реализован метод пузырьковой сортировки по алфавиту:

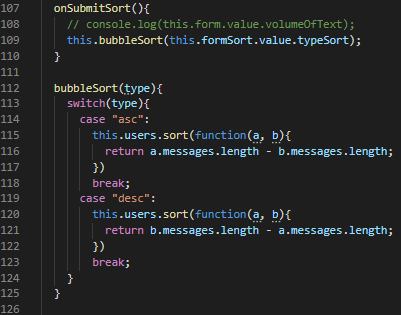


Рис.29 – Пузырьковая сортировка в home.component.ts

Для различного поиска по БД создан сервис - search.service.ts (рисунок 22):



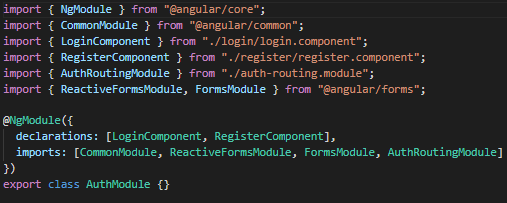
Рис.30 – search.service.ts

# Приложение

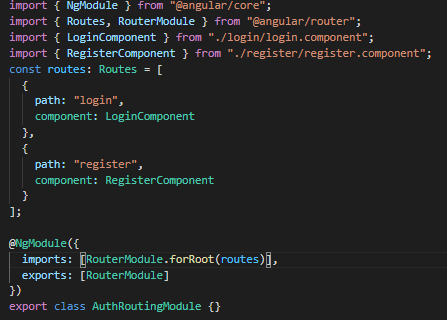
**app.module.ts**



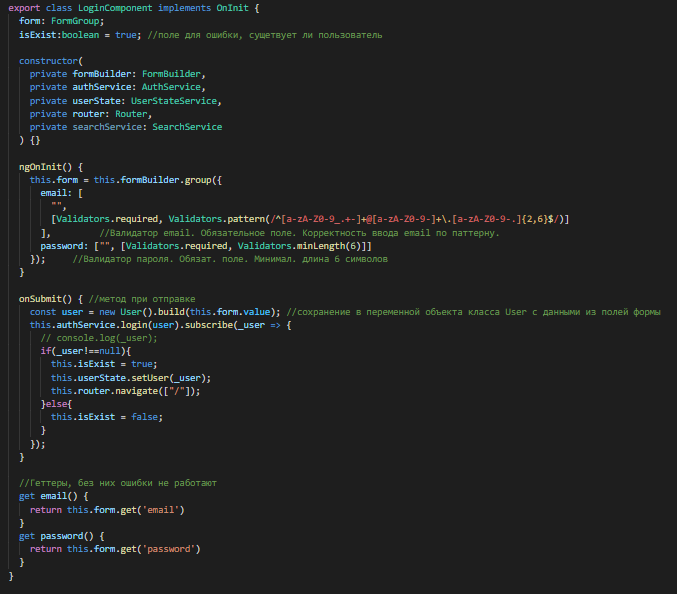
**auth.module.ts**



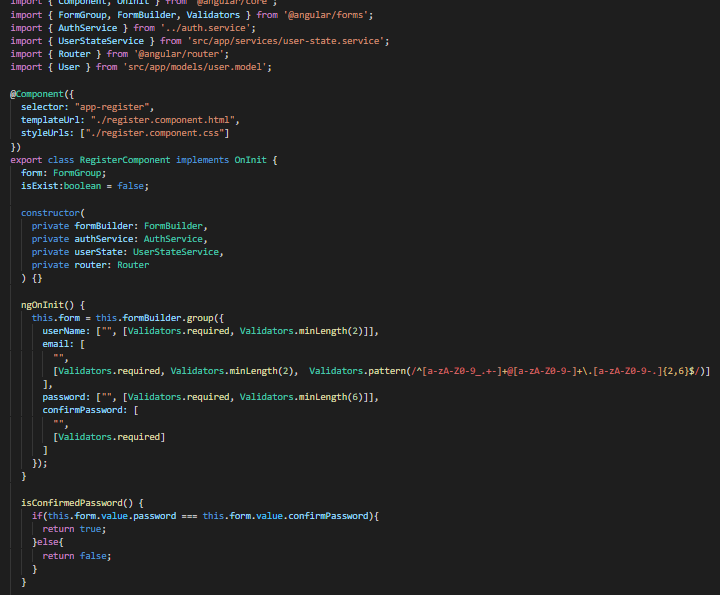
**auth-routing.module.ts**



**login.component.ts**



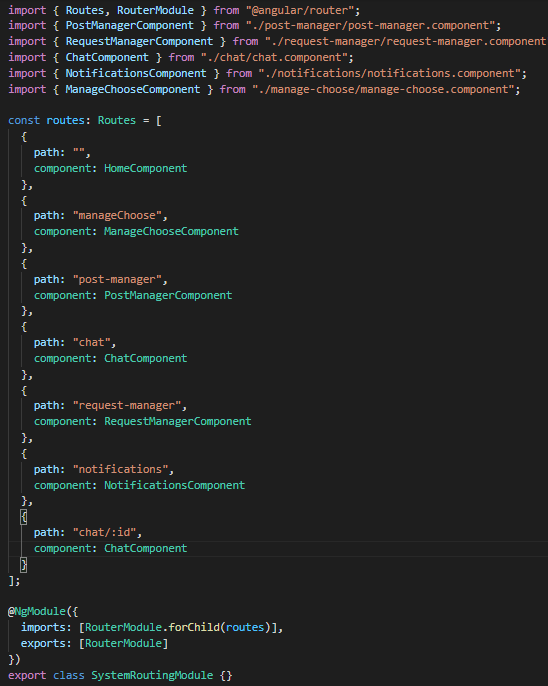
**registration.component.ts**

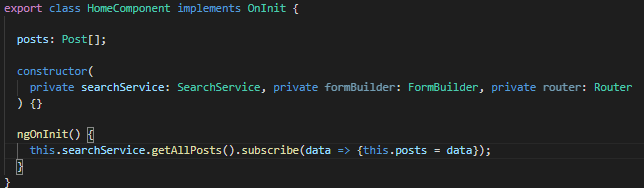
**system.module.ts**



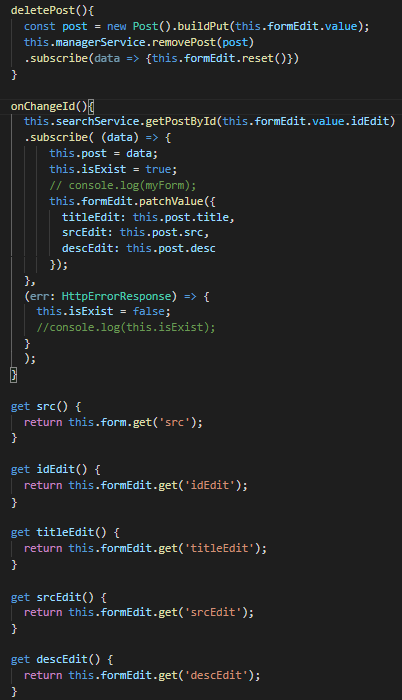
**System-routing.module.ts**



**home.component.ts**



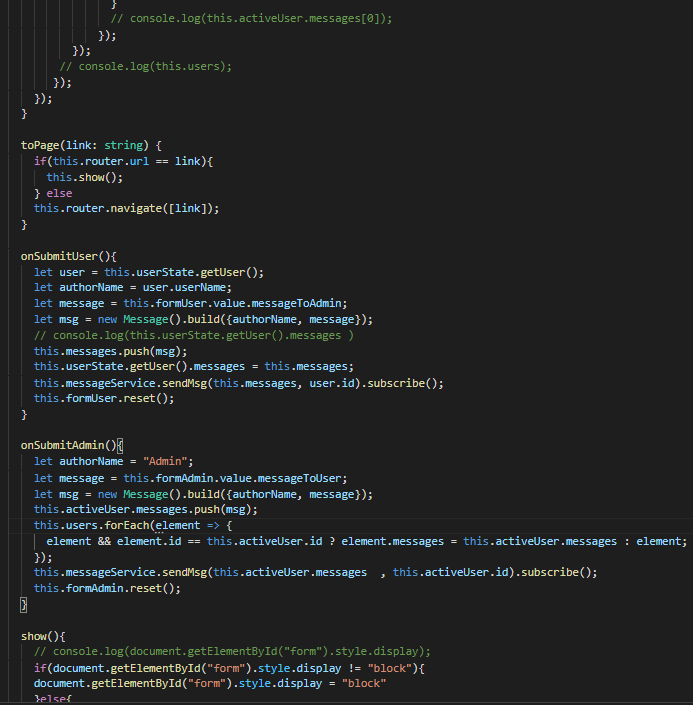
**Post-manager.component.ts**



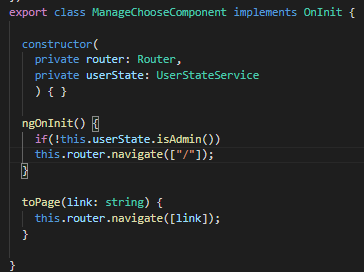
**Request-manager.component.ts** 

**Post.component.ts**

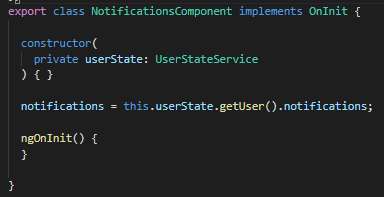


**Messege.component.ts**  

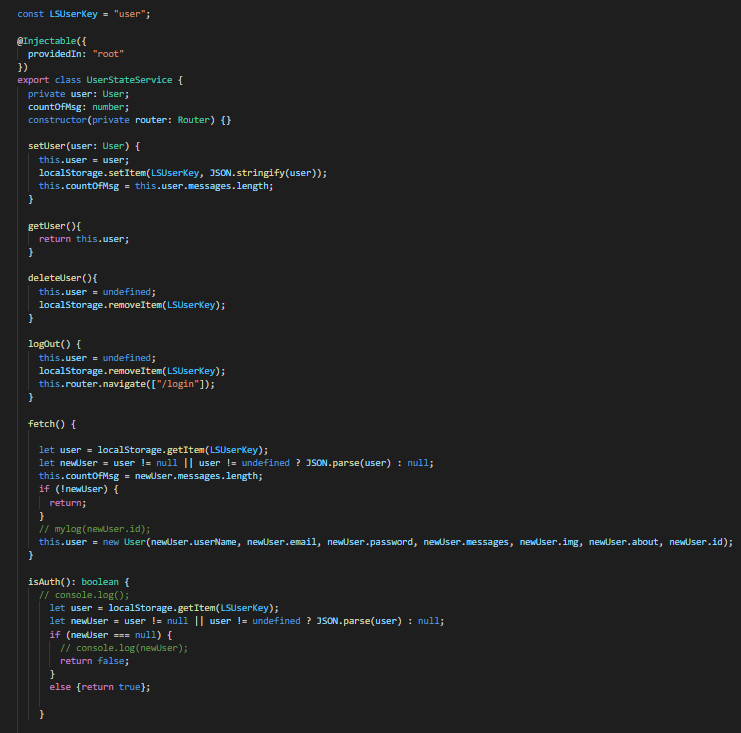
**Manage-choose.component.ts**



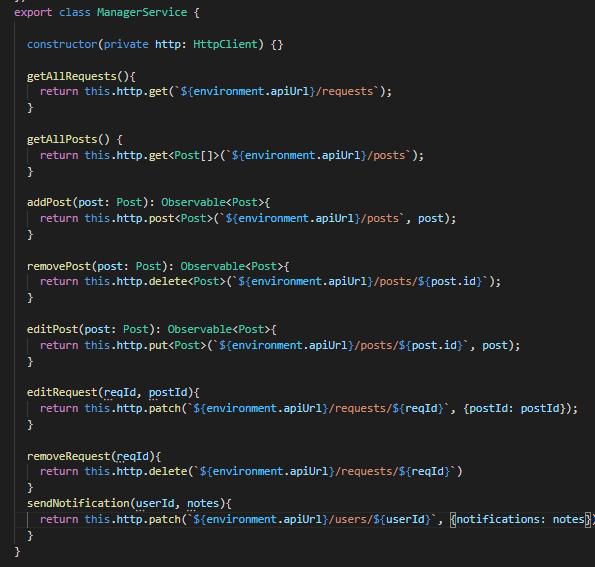
**notifications.component.ts**



**User-state.service.ts**



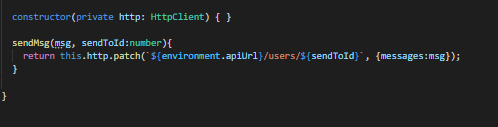
**manager.service.ts**



**Search.service.ts**



**Massage.service.ts**



**Request.service.ts**

