Санкт–Петербургский государственный университет
Математическое обеспечение и администрирование информационных систем

Ким Юния Александровна

Front-end для проекта Цифровая карта офиса

Отчёт по учебной практике

Научный руководитель:
к.т.н., доцент Литвинов Ю. В.

Консультант:
руководитель группы, ООО «Ланит-Терком»Исаков В. В.

Санкт-Петербург
2020 г.

Содержание

Введение

«Ланит-Терком»[1] – российская компания в сфере IT-технологий, предлагающая клиентам различные услуги, начиная от разработки программно-аппаратного обеспечения и реинжиниринга и заканчивая IT-консалтингом.

На данный момент в компании существует проблема того, что сотрудники не могут найти нужное им помещение или определённого человека, когда приезжают в офис, в котором они не работают. Сейчас «Ланит-Терком»имеет три офиса в Санкт-Петербурге и два филиала за его пределами, поэтому данная проблема является актуальной в компании. Новый проект «Цифровая карта офиса»смог бы решить эту задачу. Цифровую карту планировалось интегрировать в качестве модуля в существующую в компании внутреннюю информационную систему учёта рабочего времени.

У некоторых крупных компаний имеются системы для внутреннего пользования, благодаря которым работники могут совершать следующие действия: выводить карты различных офисов компании, получать информацию о сотрудниках, осуществлять поиск различных данных. Такие программы заметно упрощают работу в больших компаниях.

Данная работа является частью студенческого группового проекта компании «Ланит-Терком». Автору отчёта предлагалось поучаствовать в разработке front-end части проекта.

Постановка задачи

Целью данной работы является front-end разработка цифровой карты офиса.

Для достижения данной цели были поставлены следующие задачи.

- Изучение фреймворка Angular [2].
- Разработка архитектуры front-end части.
- Настройка роутинга.
- Вёрстка страницы авторизации и основной страницы.

У оставшейся части команды разработчиков front-end части были следующие задачи.

- Создание шаблона приложения (макеты).
- Реализация прототипа карты.

1. Обзор задач front-end части

В данном разделе приводится обзор задач front-end части, которые выполнялись другими разработчиками команды.

1.1 Описание функциональных требований

Изначально цифровая карта офиса задумывалась как одностраничное приложение. Для доступа к функционалу карты необходимо авторизоваться в системе, которая предусматривает два типа пользователей: сотрудник и администратор.

Все пользователи могут совершать следующие действия.

- Авторизация в системе.
- Просмотр списка офисов и их элементов (помещений), редактирование.
- Просмотр карты с возможностью бронирования помещений и вывода информации о сотруднике.
- Поиск помещения или сотрудника.

Администраторы, помимо уже перечисленных возможностей, имеют дополнительную функциональность.

- Редактирование списка сотрудников.
- Прикрепление сотрудника к рабочему месту.
- Создание карты.

1.2 Создание шаблона приложения

Перед началом разработки содержательной части веб-приложения одним из членов команды front-end части был разработан примерный шаблон будущей цифровой карты офиса. Работа была проделана с помощью онлайн-сервиса для дизайнеров интерфейсов и веб-разработчиков под названием Figma [3]. В созданные макеты были включены страница авторизации и основная страница.

Страница авторизации состоит из формыдля ввода логина и пароля пользователя, а также из кнопки для входа в систему. Макет данной страницы можно увидеть на рис. 1.

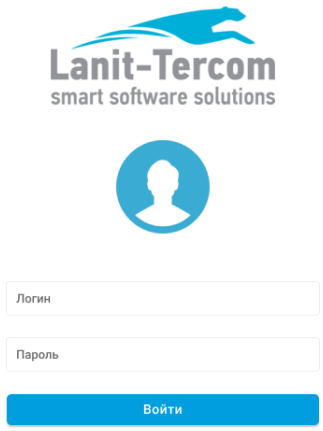

Рис. 1: Макет страницы авторизации

Что касается основной страницы, то она подразделена на три части.

В верхней части страницы располагается header, в котором есть кнопки перехода в настройки и выхода из учётной записи. Чуть ниже слева отображается название выбранной для вывода карты, а справа имеется строка поиска.

В левой части страницы находится sidebar, содержащий информацию о пользователе, а также его аватар. Под этим размещается список имеющихся офисов, каждый из которых можно раскрыть. При раскрытии какого-либо из офисов выпадает лист доступных для отображения цифровых карт.

Оставшуюся и бóльшую часть экрана занимает сама цифровая карта.

Один из макетов данной страницы можно увидеть на рис. 2.

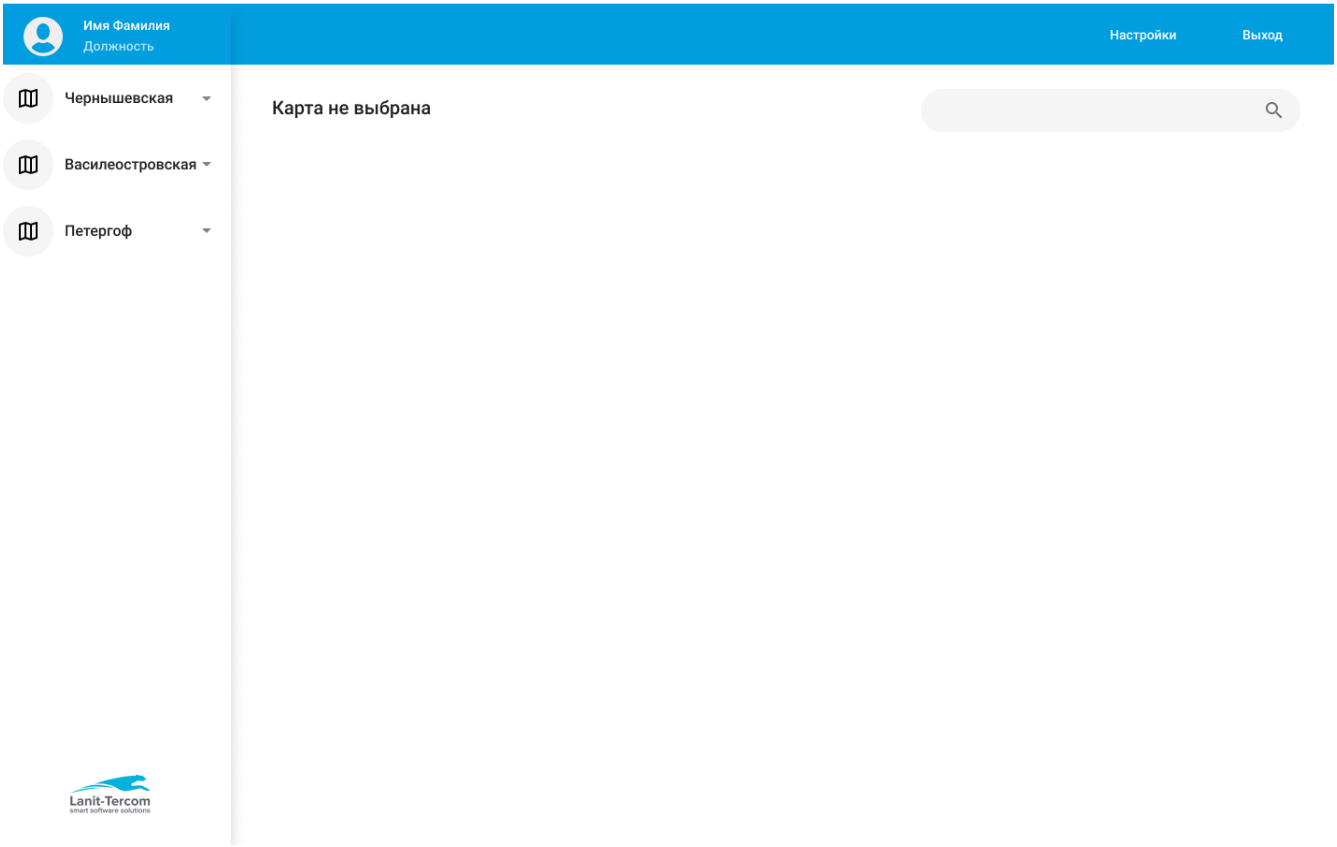

Рис. 2: Макет основной страницы без выбранной карты и с нераскрытым левым меню

1.3 Реализация прототипа карты

Для работы с картами был использоваться формат SVG, позволяющий добавлять в веб-приложение векторную графику. Перед отрисовкой карты на экране приложение получает данные о нужных помещениях от back-end части. На основе принятой информации в шаблон страницы добавляется SVG-элемент, внутри которого расположены теги, отвечающие за отображение каждого из помещений. Макет отрисованной карты на основной странице можно увидеть на рис. 3.

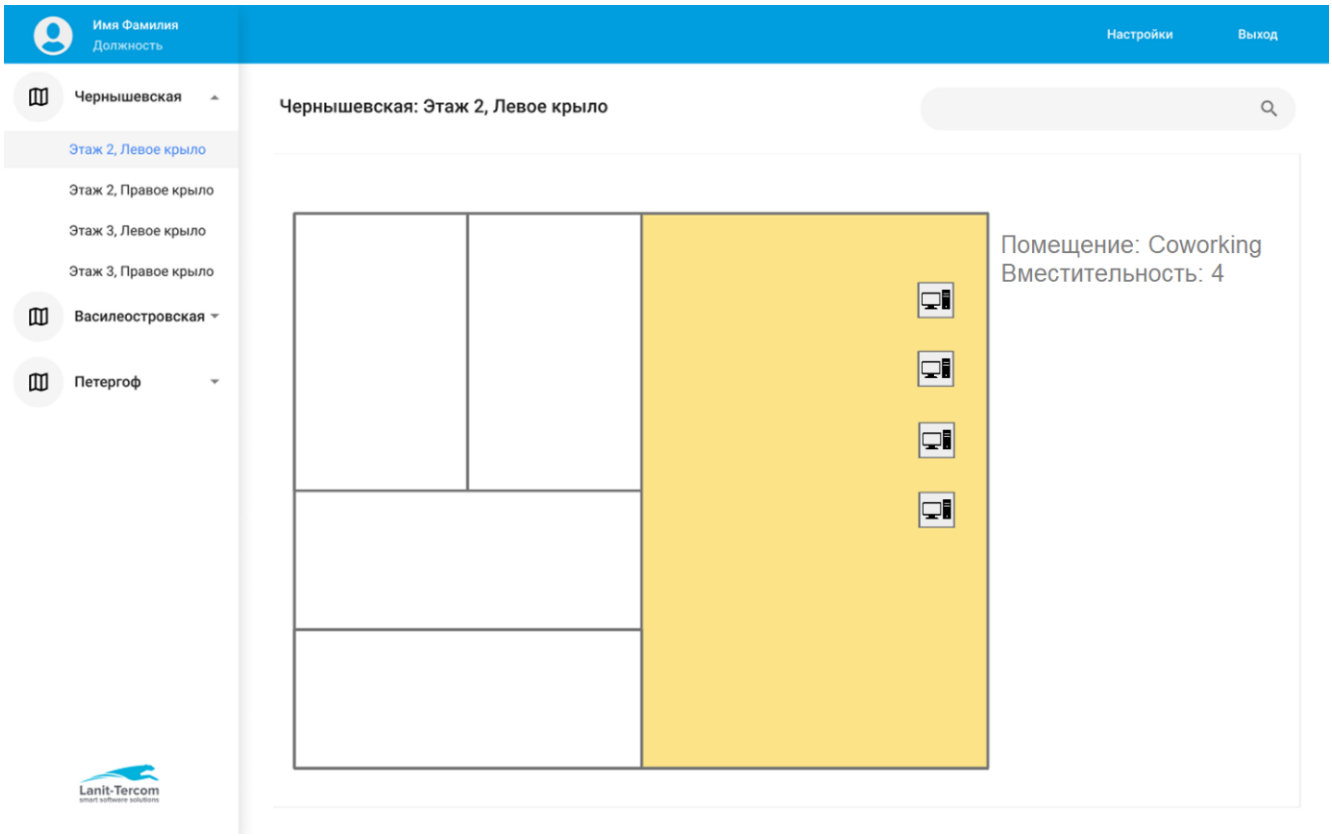

Рис. 3: Макет отрисованной карты на основной странице с раскрытым левым меню

2. Описание решения

Для разработки веб-приложения использовался широко распространённый фреймворк Angular, с которым работали кураторы проекта.

2.1 Роутинг и разработка архитектуры front-end части

Одной из первых задач при front-end разработке стала настройка роутинга. Роутинг – переходы по ссылкам и всё что с ними связано, т.е. маршрутизация внутри приложения. Для реализации роутинга использовался популярный паттерн под названием ленивая загрузка, реализация которого в Angular доступна из коробки.

При большом размере веб-приложения его загрузка может занимать достаточно много времени, что не очень хорошо и удобно для пользователя, который будет вынужден дождаться окончания этой операции для продолжения работы. Ленивая загрузка позволяет решить эту проблему. Она загружает не сразу всё приложение, а только то, что пользователь ожидает увидеть в данный момент, остальное подгружается по мере необходимости.

Для того чтобы реализовать данный паттерн, приложение должно быть разделено на несколько частей, называемых модулями, которые и будут загружаться в нужный момент. Модуль – составная часть приложения на Angular. В каждом приложении обязан быть как минимум один корневой модуль. Модули сами по себе обычно состоят из компонентов – элементов или совокупностей элементов веб-страницы, которые отображаются на экран.

На рис. 4 представлена архитектура front-end части приложения. При старте пользователю загружается корневой модуль App, из которого можно перейти либо в модуль авторизации Auth, либо на основную страницу, представленную модулем Home, включающим нескольких других модулей. Ещё раз отметим, что модули подгружаются лишь тогда, когда они необходимы, поэтому при запуске App все остальные модули (Auth, Home, Navigation, Content и т.д.) загружены не будут. Они подгрузятся только в том случае, если пользователь захочет перейти на соответствующие страницы.

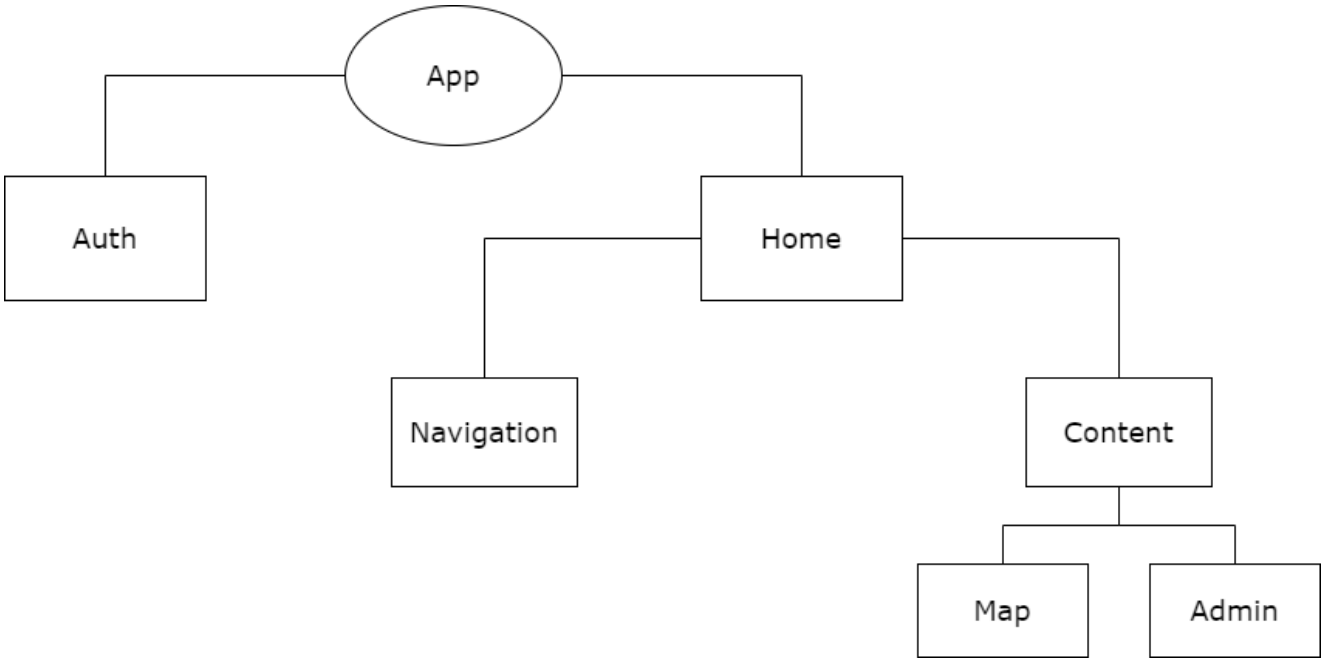

Рис. 4: Архитектура приложения (схема модулей)

2.2 Вёрстка

В процессе разработки страниц были задействованы библиотека Angular Material [4], которая содержит компоненты, используемые для вёрстки, и HTTP-клиент для тестирования веб-сайтов под названием Postman [5]. Также активно использовалась методология БЭМ [6], которая была создана для разработки веб-сайтов. Она использует компонентный подход к веб-разработке, который предполагает разделение на независимые блоки, что позволяет создавать переиспользуемые компоненты и делает разработку более удобной и эффективной. БЭМ расшифровывается как «блок-элемент-модификатор». Элемент – это составная часть блока, а сущность, в свою очередь, определяет внешний вид, состояние или поведение какого-либо объекта.

При вёрстке веб-страниц данная методология применялась только для наименования классов объектов в HTML-коде. Методология имеет определённые правила формирования имён. Например, класс скрытого меню мы можем назвать «menu menu_hidden», а классу элемента меню, который в данный момент отображается, можно дать имя «menu__item menu__item_visible».

Страница авторизации сама по себе предполагает наличие формы с полями для ввода необходимой для авторизации информации о пользователе,

например, логина и пароля.

В Angular существует 2 подхода к созданию форм:

- Шаблонные (template-driven) формы.
- Реактивные (reactive) формы.

Реактивные формы имеют следующие преимущества над шаблонными формами.

- Создаются программно, а не через шаблон HTML.
- Больше компонентного кода, меньше HTML разметки.
- Более гибкие, умеют работать со сложными сценариями.
- Проще тестировать.

По вышеперечисленным причинам при разработке страницы авторизации были использованы именно реактивные формы. Саму реализацию можно увидеть на рис. 5.

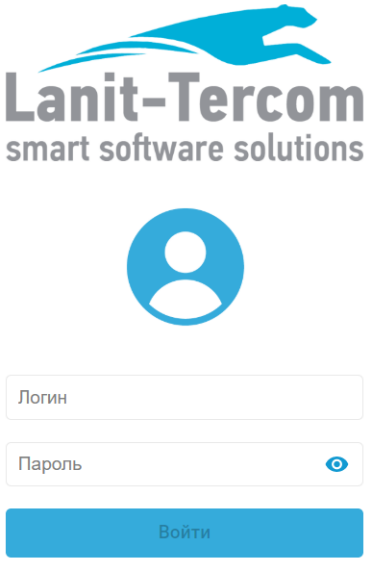

Рис. 5: Свёрстанная страница авторизации

Теперь рассмотрим рис. 6, на котором можно увидеть схему модулей и компонентов внутри Home, отвечающего за основную страницу. Модуль Home подразделён на два других: Navigation и Content. Последний из них отвечает за вывод карты, а также за некоторую функциональность администратора. Что касается модуля навигации, то он состоит из двух компонентов: top-menu (header) и side-menu (sidebar).

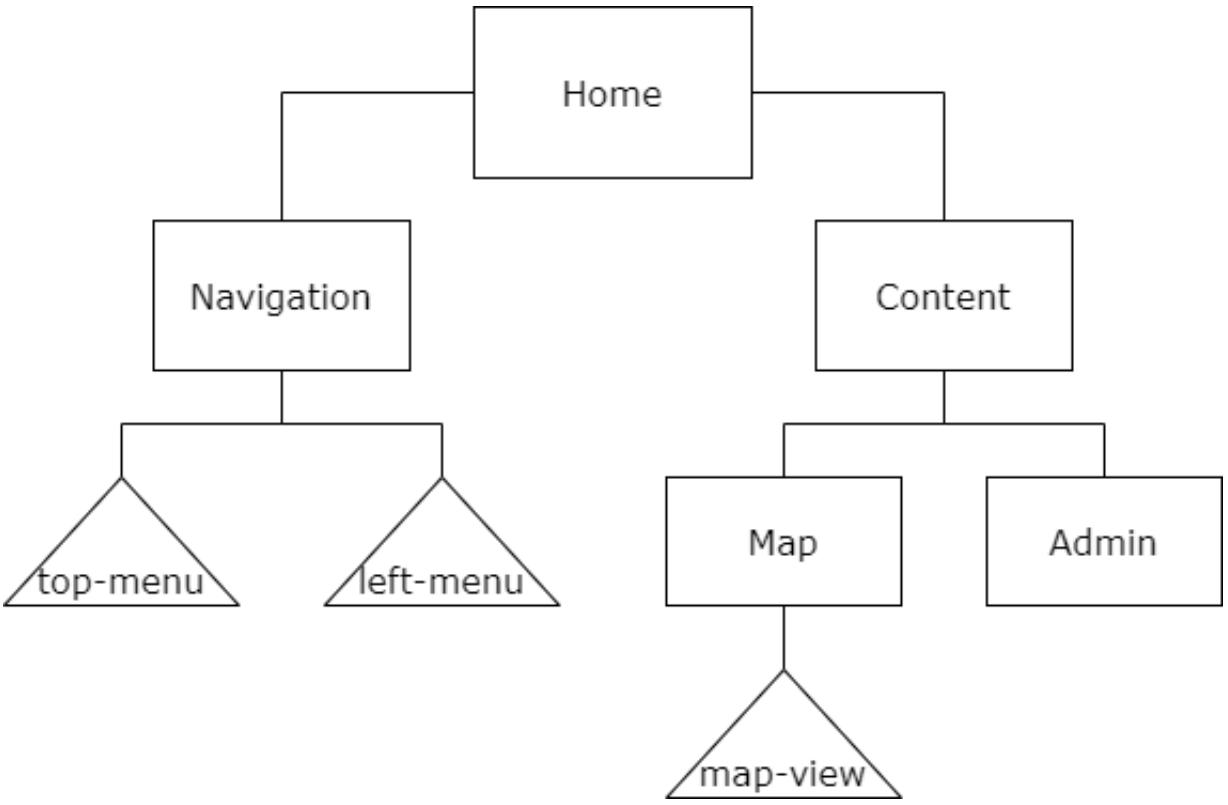

Рис. 6: Схема модулей и компонентов Home

При вёрстке основной страницы выяснилось, что необходимо получать данные, в частности, список доступных для отображения карт, с back-end части, а также передавать какую-то информацию между компонентами внутри модуля. Для этого были использованы так называемые сервисы – классы, которые выполняют некоторые специфические задачи.

В нашем случае было создано два сервиса: Offices – для получения данных об офисах и MapName – для вывода названия карты в header после выбора из sidebar. Offices получает информацию о картах с серверной части приложения в виде JSON, после этого информация обрабатывается – извлекаются необходимые данные, которые передаются в компонент left-menu, который выводится на экран. MapName, в свою очередь, сначала принимает информацию, а именно название выбранной для отображения карты, из компонента left-menu, а после передаёт её в другой компонент, top-menu, для вывода пользователю.

Готовую основную страницу авторизации можно увидеть на рис. 7.

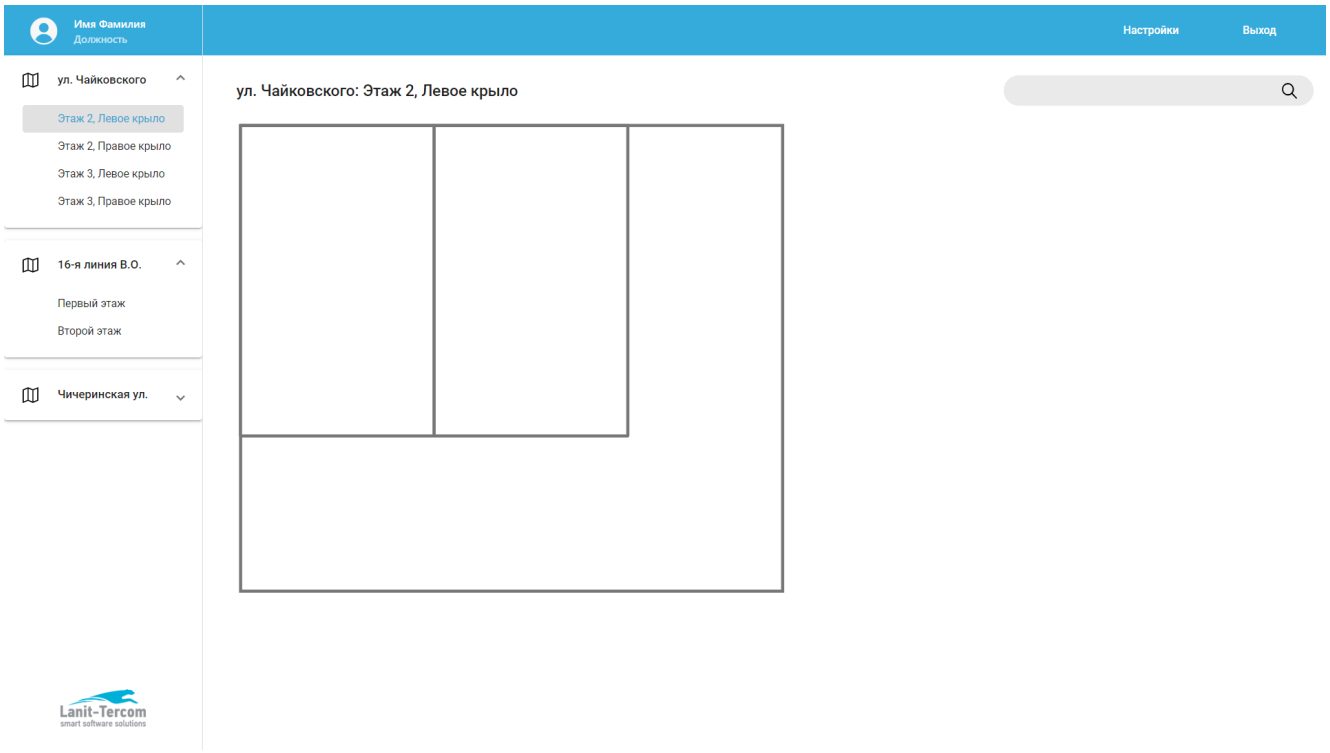

Рис. 7: Свёрстанная основная страница

Заключение

В рамках данной учебной практики были достигнуты следующие результаты.

- Изучен фреймворк Angular.
- Разработана архитектура front-end части.
- Настроен роутинг.
- Свёрстаны страница авторизации и основная страница.

Текущие результаты представлены в репозитории на GitHub [7].

Список литературы

[1] Веб-сайт компании «Ланит-Терком».

URL: https://www.lanit-tercom.ru (дата обращения: 09.06.2020).

[2] Веб-сайт фреймворка Angular.

URL: https://angular.io (дата обращения: 09.06.2020).

[3] Веб-сайт онлайн-сервиса Figma.

URL: https://www.figma.com (дата обращения: 09.06.2020).

[4] Веб-сайт библиотеки Angular Material.

URL: https://material.angular.io (дата обращения: 09.06.2020).

[5] Веб-сайт HTTP-клиента Postman.

URL: https://www.postman.com (дата обращения: 09.06.2020).

[6] Веб-сайт методологии БЭМ.

URL: https://ru.bem.info (дата обращения: 09.06.2020).

[7] Репозиторий проекта «Цифровая карта офиса».

URL: https://github.com/lanit-office-map/OfficeMap (дата обращения: 09.06.2020).