

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«___» _____ 2023 г.

Разработка интерактивного веб-интерфейса для карты аудиторного фонда университета

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ – 090301.2023.065 ПЗ ВКР

Руководитель работы,
к.п.н., доцент каф. ЭВМ
_____ Ю.Г. Плаксина
«___» _____ 2023 г.

Автор работы,
студент группы КЭ-406
_____ М.Э. Черемушкин
«___» _____ 2023 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ С.В. Сяськов
«___» _____ 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«___» _____ 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-406

Черемушкину Михаилу Эдуардовичу,

обучающемуся по направлению

09.03.01 «Информатика и вычислительная техника»

1. Тема работы: «Разработка интерактивного веб-интерфейса для карты аудиторного фонда университета» утверждена приказом по университету от 25 апреля 2023 г. № 753-13/12

2. Срок сдачи студентом законченной работы: 01 июня 2023 г.

3. Исходные данные к работе:

Требования к функционалу разрабатываемого приложения:

- Интерактивный веб-интерфейс трехмерной карты вуза
- Серверная операционная система Ubuntu Server 22.04
- Система управления базами данных PostgreSQL
- Поддержка кроссплатформенности
- Механизм поиска аудиторий по названию (номеру)
- Веб-приложение должно поддерживать GET-запросы с параметрами поиска, чтобы сразу по URL отобразить выделенную аудиторию

4. Перечень подлежащих разработке вопросов:

Выпускная квалификационная работа должна содержать разбор следующих вопросов:

1. Анализ предметной области
2. Определение требований
3. Проектирование системы
4. Реализация клиентской и серверной частей
5. Тестирование

5. Дата выдачи задания: ____ декабря 2022 г.

Руководитель работы _____ / Ю.Г. Плаксина /

Студент _____ / М.Э. Черемушкин /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Анализ предметной области	01.01.2023	
Определение требований	15.01.2023	
Проектирование системы	01.02.2023	
Реализация клиентской и серверной частей	01.03.2023	
Тестирование	01.05.2023	
Компоновка текста работы и сдача на нормоконтроль	15.05.2023	
Подготовка презентации и доклада	24.05.2023	

Руководитель работы _____ / Ю.Г. Плаксина /

Студент _____ / М.Э. Черемушкин /

АННОТАЦИЯ

М.Э. Черемушкин Разработка интерактивного веб-интерфейса для карты аудиторного фонда университета. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 43 с., 24 ил., библиогр. список – 19 наим.

Работа посвящена разработке интерактивного веб-интерфейса, содержащего трёхмерную карту аудиторного фонда университета. Она предназначена для облегчения ориентации внутри кампуса НИУ «ЮУрГУ».

Данная работа состоит из введения, пяти глав, заключения, библиографического списка.

В главе "Анализ предметной области" проводится обзор систем и проблем, связанных с интерактивными картами. В главе "Определение требований" формулируются функциональные и нефункциональные требования. В главе "Проектирование" определяется архитектура и выбираются технологии. В главе "Реализация" разрабатывается интерфейс. В главе "Тестирование" проверяется работоспособность интерфейса. В главе "Заключение" подводятся итоги и делаются рекомендации.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	9
1.1 Обзор литературы.....	9
1.2 Обзор аналогов	11
1.3 Анализ основных технологических решений	16
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.....	23
2.1 Функциональные требования.....	23
2.2 Нефункциональные требования	23
3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ.....	24
3.1 Архитектура предлагаемого решения.....	24
3.2 Описание данных	25
4. РЕАЛИЗАЦИЯ КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТЕЙ	28
5 ТЕСТИРОВАНИЕ.....	36
5.1 Проведение процедуры тестирования.....	36
ЗАКЛЮЧЕНИЕ.....	41
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	42

ВВЕДЕНИЕ

В последнее время на смену бумажным картам и атласам пришли электронные, или интерактивные, карты, позволяющие находить любые объекты в считанные секунды: достаточно ввести название объекта, щелкнуть на соответствующей кнопке, и нужный фрагмент карты окажется перед вами. Но дело не только в скорости — в электронных картах, в отличие от обычных, реализован интерактивный просмотр, то есть они реагируют на действия пользователя. Также особенностью интерактивных карт является наибольшая информативность по сравнению с бумажным вариантом. Степень интерактивности карт может быть разной. В простейшем случае интерактивность заключается, например, в отображении какой-то информации по выбранному объекту.

Наибольшее распространение интерактивные-карты получили в отображении организации городской структуры: улицы, муниципальные здания, магазины, степень загруженности на дорогах и прочие объекты, пользующиеся спросом у граждан.

Также немаловажной областью использования интерактивных карт стали предприятия и офисные здания с большим количеством пристроек, помещений и этажей, где сложно ориентироваться новым сотрудникам, а иногда затрудняет поиск даже знающим работникам.

Но в случае с сотрудниками им нужно знать только своё рабочее место, столовую и кабинет начальника, иначе же дело обстоит с первокурсниками в учебных заведениях, которым нужно каждый раз искать новую аудиторию. В случае, когда у вуза только один корпус ситуация проще, но когда у вуза 3 и более корпусов с различной структурой зданий ориентироваться, особенно новым студентам, становится в разы сложнее.

Кампус университета – комплекс сооружений, состоящий из обособленной территории, коммуникаций, зданий, дорог и дорожного покрытия и т.д. [1]. Именно этот факт позволяет рассматривать «кампус», как отдельный проект для карты.

Карты кампусов часто размещаются на сайтах университетов. Есть ряд примеров в российской практике, когда карта включается в мобильное приложение. Однако она не является ядром этих проектов и представляет собой аналог печатных версий (т.е. не интерактивна). Эти приложения включают в себя такие разделы, как расписание занятий, информация о предстоящих мероприятиях, новости университета, т.е. они дублируют информацию, размещенную на сайте.

Особенностью данного проекта является использование 3D-моделей. Так как карта рассчитана на ограниченную и относительно небольшую территорию, эстетическому аспекту уделяется значительное внимание.

Целью выпускной квалификационной работы является разработка трёхмерной интерактивной карты аудиторного фонда Южно-Уральского государственного университета. Для достижения цели были поставлены следующие задачи:

- Анализ предметной области

- Определение требований
- Проектирование системы
- Реализация клиентской и серверной частей
- Тестирование

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор литературы

Интерактивная карта – электронная карта, работающая в режиме двустороннего диалогового взаимодействия человека (пользователя) и компьютера и представляющая собой визуальную информационную систему [2]. Она позволяет расширить понятие информативности благодаря скрытым данным; выполнив определенные действия (наведение курсора), пользователь может получить дополнительные сведения.

Тема интерактивной геолокации стала актуальной в начале 2009 года, когда для навигаторов и смартфонов стало возникать большое число приложений с GPS привязкой.

Цифровые карты направлены на решение следующих задач: навигация, поиск точки на карте, построение оптимального маршрута, фильтрация по заданному параметру.

В основе цифровой карты лежат базы данных, которые имеют географическую привязку (геоданные), то есть в которых объекты привязаны к системе координат, что позволяет связывать их между собой. Например, здание N имеет определенные координаты, которые совпадают с координатами заведения M; из этого мы можем сделать вывод, что заведение M располагается в здании N. Далее в базы данных добавляются и другие слои информации.

Интерактивная карта состоит из слоев, которые различаются по типу географического объекта: полигон (объект, в основе которого лежит многоугольник – парк, территория страны, озеро, море и т.д.), точка (ресторан, памятник, и т.д.), линия (улица, граница страны, границы города, река и т.д.).

Интерактивные карты являются более перспективным способом визуализации, в отличие от печатных аналогов. Они получили ряд свойств, которые позволяют воспринимать большие потоки информации за короткий период времени. К возможностям интерактивной карты можно отнести: масштабирование, панорамирование, демонстрацию хронологии, приближение (с более подробной детализацией), возможность проследить последствия каких-либо событий («до» и «после»). Важным плюсом также является возможность редактирования карты в реальном времени, что невозможно в печатных аналогах.

Карта может быть не только способом хранения информации, но и инструментом для сбора данных. Примером может быть инструмент, выпущенный CNN в 2008 году во время президентских выборов, где пользователи могли оставлять свои голоса, которые в режиме реального времени отображались на карте. Таким образом, можно было сразу увидеть ситуацию по выборам в разных регионах США [3].

Можно рассматривать карту как новую форму коммуникации. Представление данных в связи с географической привязкой позволяет осуществить интеграцию существующего знания с новым. Пространственный, временной и культурный контекст помогает создать сеть ассоциаций, благодаря которой вновь изученный материал встраивается в то, что пользователь уже знает [4]. Кроме привычного

понимания карты как инструмента для разметки территории в географических целях, всё большую популярность набирают проекты, где карта представляет собой инструмент графического интерфейса. Они ориентированы, в том числе, на гуманитарные науки. Например, создатель исторического проекта Revillna (сайт, посвященный Вильнюсскому гетто) называет свой проект интерактивной картой. Он объясняет необходимость использования карты тем, что гетто четко определено в пространстве и времени, оно было буквально ограничено заборами территориально и строгими хронологическими рамками. Карта на сегодняшний день является наиболее интуитивно понятным отображением этого [4].

Интерактивные карты являются популярным инструментарием для гуманитарных наук (чаще для исторических дисциплин), в том числе и благодаря возможности нелинейного анализа. Рассмотрим проект, посвященный истории города Кливленд [5]. На стартовой странице представлена карта, на которой отмечены здания, события, люди, связанные с историей этого города. При выборе одной из категорий осуществляется переход на страницу с историей, дополненной изображениями. Из этого пункта есть возможность совершить переход на страницу со схожими тегами. И наконец, предоставлена возможность выбора случайной истории. Так проявляется нелинейность анализа.

Отдельным видом интерактивной карты можно назвать ГИС (Геоинформационная система). Геоинформационные системы – автоматизированные информационные системы, предназначенные для обработки пространственно-временных данных, основой интеграции которых служит географическая информация [6].

ГИС подразделяются по функциональным возможностям (на полнофункциональные общего назначения, специализированные ГИС, ориентированные на решение конкретной задачи в какой-либо предметной области, информационно-справочные системы для домашнего и информационно-справочного пользования); по пространственному признаку (глобальные, общенациональные, региональные, локальные); по проблематематической ориентации (общегеографические, экологические, природопользовательские, отраслевые); по способу организации географических данных (векторные, растровые, векторно-растровые).

ГИС включает в себя 5 компонентов:

1. Система ввода – программное обеспечение для получения данных;
2. Графическая и тематическая базы данных;
3. Система визуализации – программное обеспечение для вывода на монитор имеющейся информации в виде карт, таблиц, схем и т.д.;
4. Система управления и обработки – программное обеспечение для управления данными, при помощи которой происходит их поиск, сортировка, удаление, добавление, исправление и анализ;
5. Система вывода – программное обеспечение для представления результатов работы в виде, удобном пользователю [7].

Именно функции ГИС разводят понятия «ГИС» и «интерактивная карта». Первая функция – визуализация. ГИС – мощное средство представления данных. Вторая – организация, она позволяет управлять информацией наиболее

комфортным способом. И наконец, обработка и анализ – функции ГИС, превращающие её из инструмента по работе с готовыми данными (визуализатора) в инструмент по созданию новых данных на их основе моделирования и прогнозирования [8]. Можно сделать вывод, что любая ГИС включает в себя интерактивную карту, но не любая интерактивная карта позволяет создавать новые данные на основе базы данных.

1.2 Обзор аналогов

На сегодняшний день на рынке существует множество аналогов интерактивных трехмерных карт, которые могут быть использованы в контексте создания карты аудиторного фонда вуза.

Критериями для сравнения аналогов были выбраны: кроссплатформенность, поиск, трёхмерность, стоимость реализации.

Рассмотрим несколько из них:

SmartMap

Является продуктом студентов СПбГЭТУ «ЛЭТИ». SmartMap [9] – это интерактивная карта с дополненной реальностью, которая помогает найти нужную им аудиторию и проложить кратчайший путь к ней.

Особенности:

- Приложение имеет реализацию как на Android, так и на IOS.
- В ходе разработки был создан алгоритм на основе компьютерного зрения для оцифровки физических карт.
- В приложении есть AR-модуль для построения маршрутов и возможность определять местоположения с помощью Beacon-метки.

Интерфейс и визуальный вид самой карты представлен на рисунке 1.

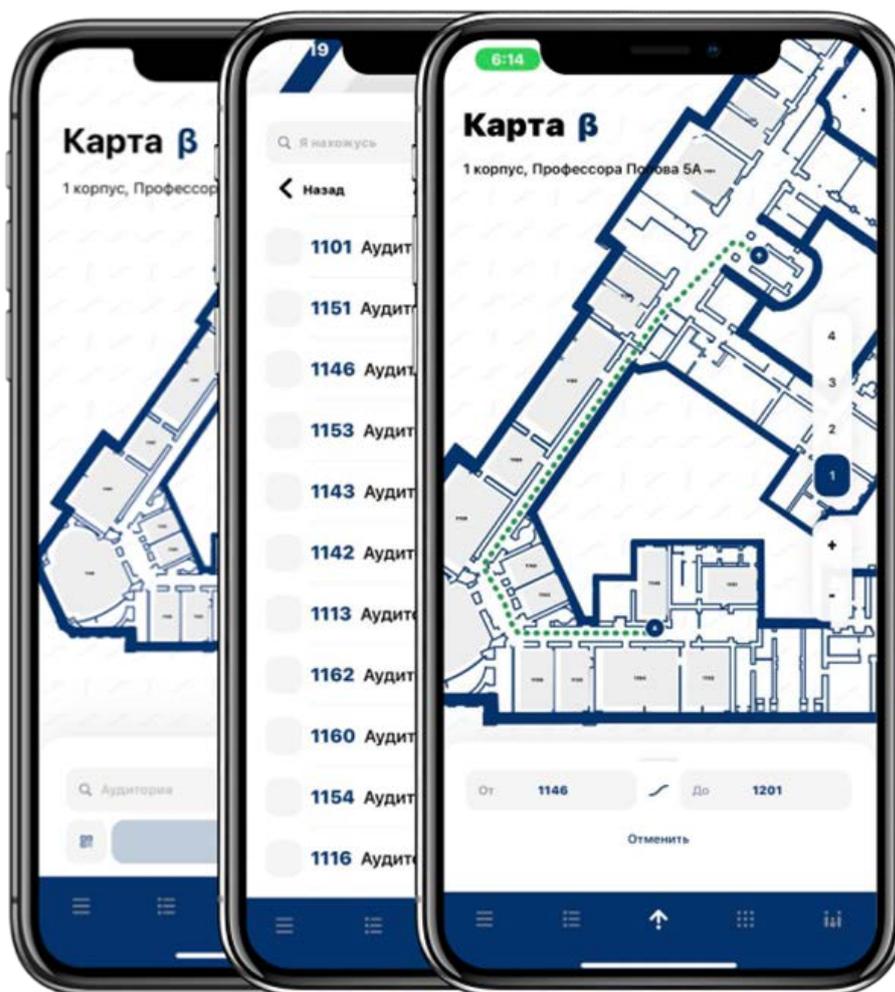
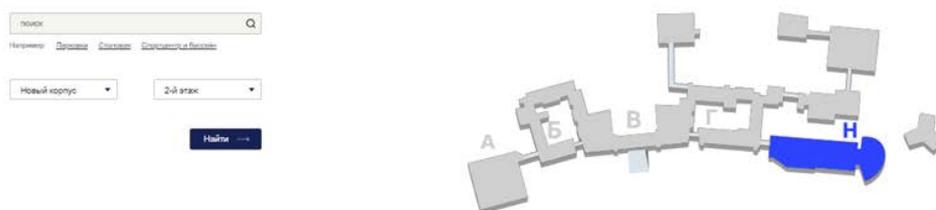


Рисунок 1 – SmartMap

Карта Университета МГИМО

Карта встроена на веб-странице университета МГИМО [10]. Имеет интерактивный поиск по корпусам и этажам. Каждый этаж представлен в виде двухмерной проекции с возможностью клика на аудиторию, где показана информация о ней. Интерфейс и визуальный вид самой карты представлен на рисунке 2.

Карта Университета



Новый корпус, 2-й этаж

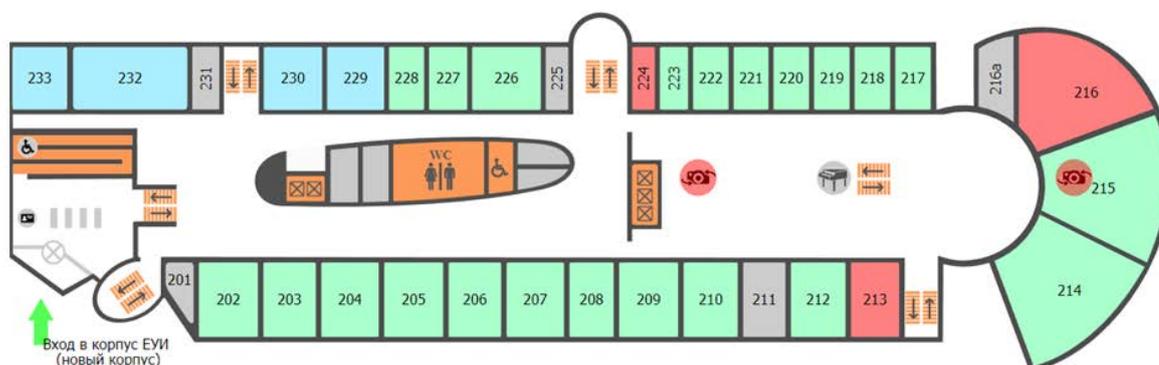


Рисунок 2 – Карта Университета МГИМО

План МИИГАиК

Интерфейс и визуальный вид самой карты представлен на рисунке 3.

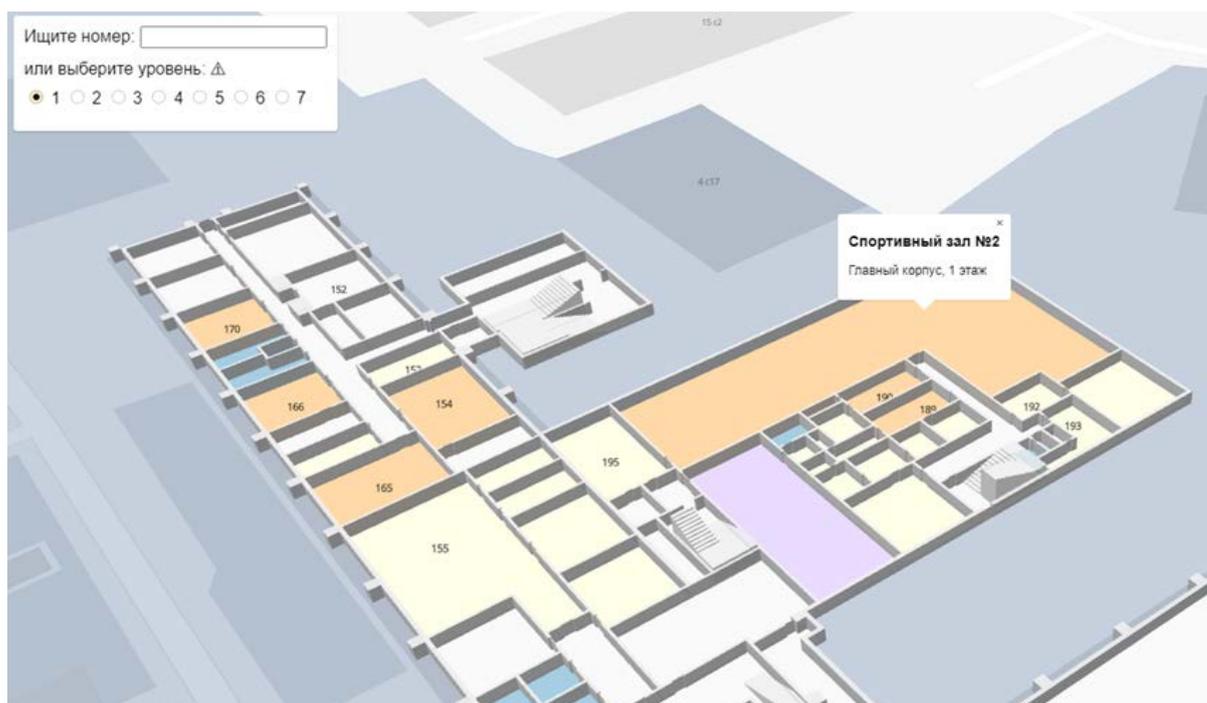


Рисунок 3 – План МИИГАиК

Интерактивная карта данного вуза [11] построена на основе библиотеки MapVox GL JS [12]. MapVox – американский поставщик пользовательских онлайн-карт для веб-сайтов и приложений.

Особенностью реализации данного приложения можно назвать использование готовых решений, то есть данную интерактивную карту можно смоделировать за кратчайшие сроки, но только предоставленный разработчиками функционал может быть использован.

2ГИС Этажи

Этажи [13] — это особенность, позволяющая посмотреть схему этажей здания и найти на ней нужную организацию.

Этажи для веба сделаны на технологии WebGL [14]: они полностью трёхмерные, их можно вращать и приближать. Интерфейс и визуальный вид самой карты представлен на рисунке 4.

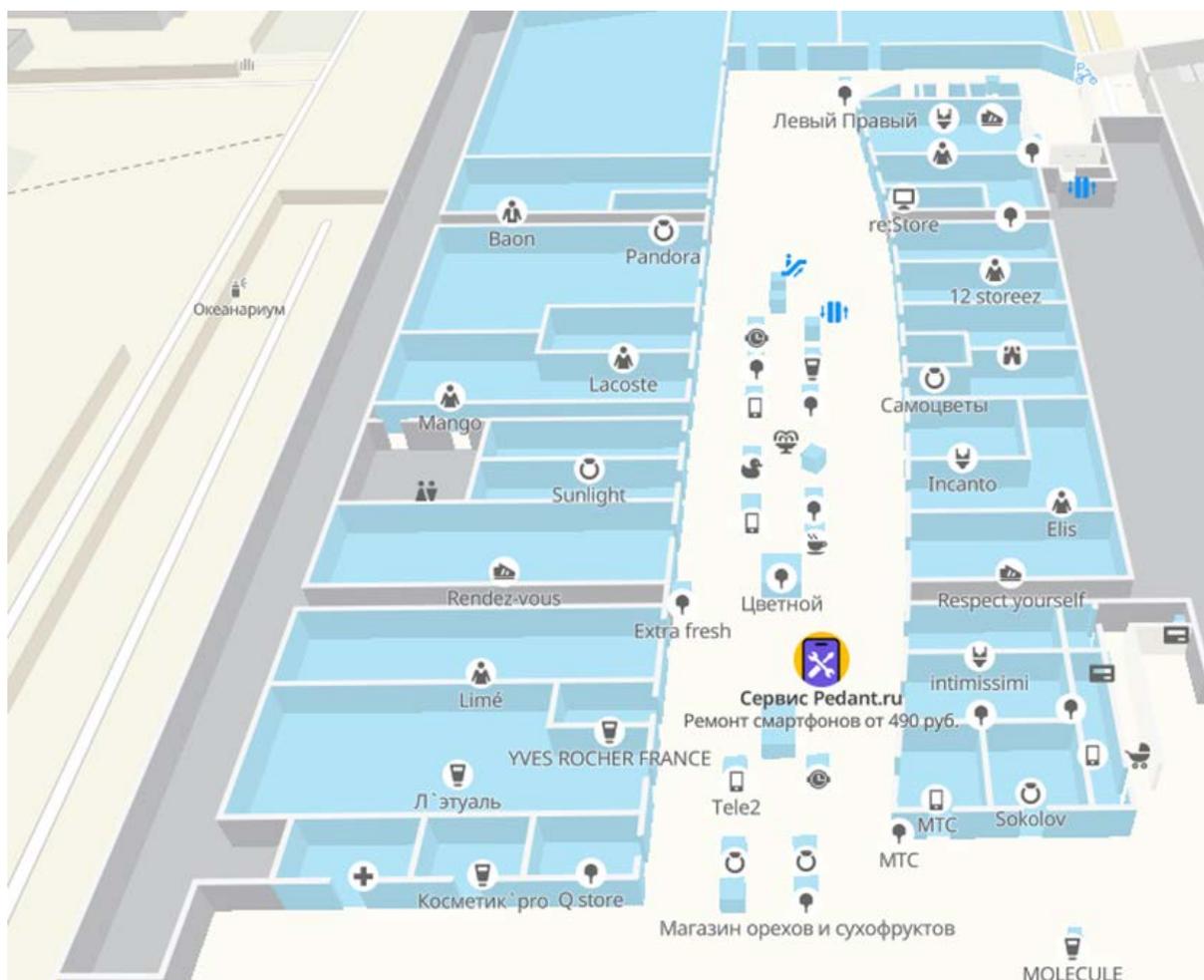


Рисунок 4 – 2ГИС Этажи

ЯНДЕКС.КАРТЫ

Яндекс.Карты [15] — это картографическая служба компании Яндекс. Представлена обществу в 2004 году и является самой часто используемой в СНГ. Из основного функционала есть: поиск по карте, загруженность дорог, отслеживание городского транспорта, построение маршрутов и панорамы улиц крупных городов.

Приложение так же, как и 2ГИС Этажи имеет возможность визуализации внутренней структуры торговых комплексов, но в отличие от своего аналога использует технологию псевдо трёхмерности. На двумерное изображение заранее

накладываются тени, придающие объём объектам. Но эти «запечённые тени» имеют один недостаток: при вращении карты наклон стен направлен всегда в одну сторону, что придаёт эффект эскизности или недовершенности.

На схему нанесены все заведения на этаже: магазины, кафе, банкоматы и так далее. Отдельными значками отмечены объекты, без которых сложно обойтись: парковки, гардеробы, туалеты и точки, где можно зарядить смартфон. Есть пиктограммы с лестницами, лифтами и эскалаторами. Интерфейс и визуальный вид самой карты представлен на рисунке 5.

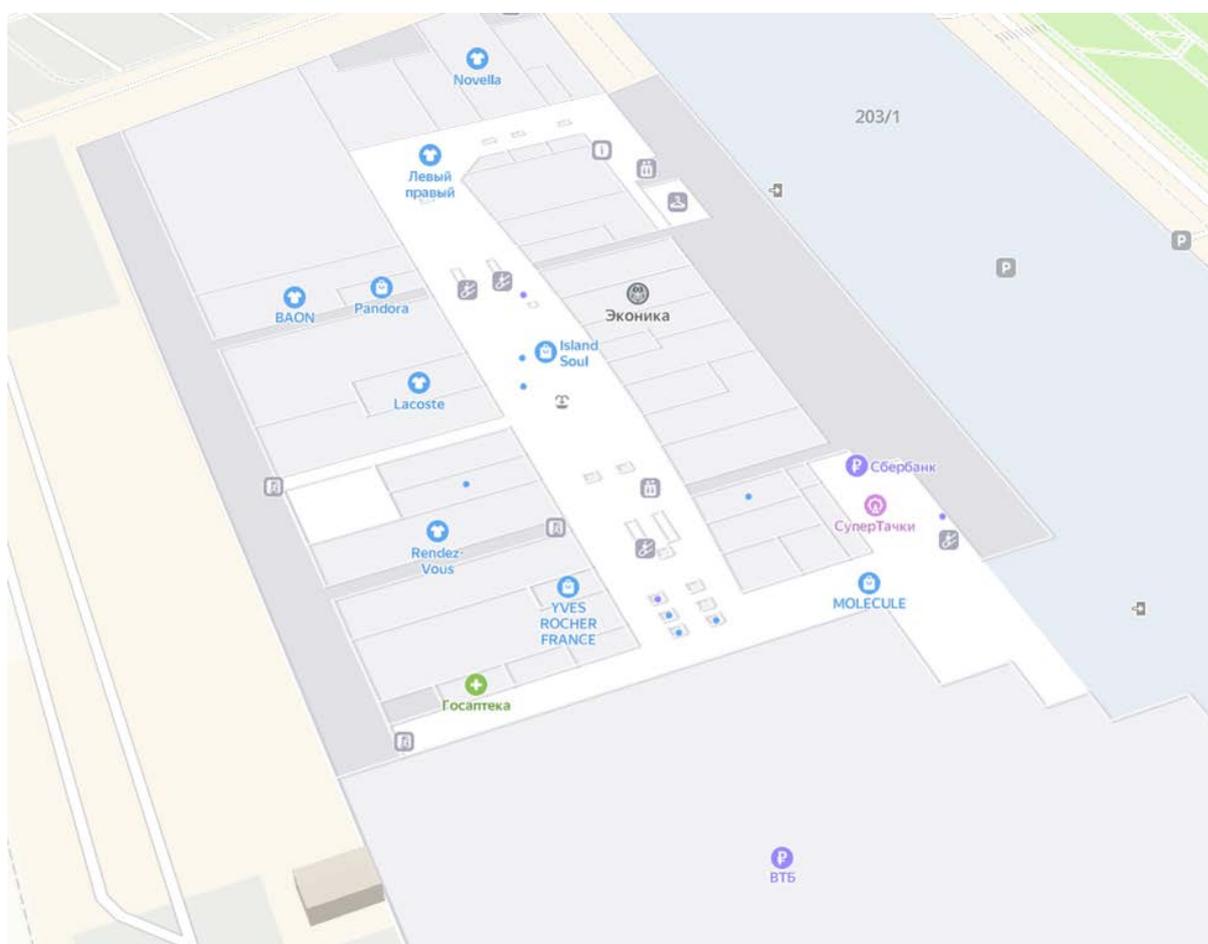


Рисунок 5 – ЯНДЕКС.КАРТЫ

В таблице 1 приведено сравнение вышеперечисленных аналогов.

Таблица 1 – Сравнение аналогов

	SmartMap	карта МГИМО	карта МИИГАиК	2ГИС Этажи	Яндекс Карты
кроссплатформенность	±	+	+	+	+
поиск	+	+	+	+	+
трёхмерность	–	–	+	+	±

стоимость реализации	средняя	низкая	средняя	высокая	высокая
----------------------	---------	--------	---------	---------	---------

Таким образом, приложение 2ГИС Этажи является бесспорным лидером интерактивных трёхмерных карт, но использование их продукта не бесплатно, поэтому не подходит для разработки. Следовательно, создание приложения, свободного от выявленных недостатков, и учитывая немаловажные преимущества является актуальной задачей.

1.3 Анализ основных технологических решений

Далее представлен обзор существующих технологий и инструментов, которые могут быть использованы для реализации проекта. Это позволяет оценить преимущества и недостатки каждой технологии, а также выбрать оптимальный набор инструментов для достижения поставленных целей.

В рамках данного пункта проводится сравнительный анализ существующих решений, а также оценка их применимости и соответствия задачам, поставленным в работе. Этот анализ помогает определить технологии и инструменты, которые лучше всего подходят для решения конкретных задач.

Выбор языка программирования

Выбор языка программирования имеет большое значение. Правильный выбор языка программирования помогает упростить и ускорить процесс разработки, а также обеспечить эффективность и масштабируемость созданного приложения.

Так как разрабатывается веб-приложение, то выбор языка программирования должен основываться на его возможностях в области веб-разработки.

При выборе языка программирования нужно также учитывать его популярность и поддержку в сообществе разработчиков. Язык программирования, который имеет большое количество разработчиков и активную поддержку, может быть более стабильным и безопасным выбором, поскольку он может быстро исправлять ошибки и устранять уязвимости.

В целом, выбор языка программирования должен основываться на требованиях к создаваемому приложению, опыте и знаниях разработчика, а также на поддержке и популярности языка в сообществе разработчиков.

Далее рассмотрим основные языки для разработки клиентской части приложения. Предпочтения выбора будем строить по следующим критериям: производительность, простота разработки, разнообразие инструментов, большое сообщество, безопасность.

JavaScript

JavaScript (JS) является одним из самых популярных языков программирования для разработки фронтенда. Рассмотрим его плюсы и минусы [16].

Плюсы:

- Широкое распространение: JS является стандартом веб-разработки и широко используется для создания интерактивных и динамических веб-страниц.

- Простота: JS прост в изучении и использовании, поскольку язык использует простые и понятные конструкции.
- Быстрота разработки: JS позволяет быстро создавать прототипы и простые приложения благодаря своей простоте.
- Огромное сообщество: JS имеет огромное сообщество разработчиков и множество библиотек и фреймворков для разработки.
- Огромное количество библиотек и фреймворков для разработки, таких как React, Angular, Vue, и т.д.
- Возможность использования JavaScript как для фронтенда, так и для бекенда (Node.js).

Минусы:

- Ошибки и безопасность: JS подвержен ошибкам и уязвимостям безопасности, особенно если код написан неаккуратно.
- Производительность: JS медленнее, чем некоторые другие языки, такие как C++, что может стать проблемой при разработке сложных приложений.
- Сложность масштабирования: JS может быть сложным в масштабировании приложений, особенно при использовании большого количества библиотек и фреймворков.
- Различия браузеров: JS может иметь различное поведение в различных браузерах, что может создавать проблемы при разработке кросс-браузерных приложений.

В целом, JS — это отличный выбор для разработки фронтенда благодаря своей простоте, широкой распространенности и огромному сообществу. Однако, при разработке сложных приложений или при неаккуратной работе с языком, могут возникнуть проблемы с безопасностью и производительностью.

Ruby

Ruby является динамическим языком программирования, который часто используется для веб-разработки и может использоваться для написания фронтенд-кода. Давайте рассмотрим некоторые из плюсов [17] и минусов [18] использования Ruby для фронтенда.

Плюсы:

- Простота: Ruby имеет простой и понятный синтаксис, который легко читать и писать. Это может ускорить процесс разработки и уменьшить количество ошибок.
- Мощный фреймворк: Ruby on Rails – один из самых популярных и мощных фреймворков веб-разработки. Он предоставляет разработчикам множество инструментов и функций, которые позволяют создавать эффективные и красивые веб-приложения.
- Динамическая типизация: Ruby является динамически типизированным языком программирования, что означает, что не нужно задавать тип переменной при ее объявлении. Это упрощает процесс написания кода и сокращает время, необходимое для написания программы.
- Разнообразие инструментов: Ruby имеет множество библиотек и инструментов, которые могут помочь в разработке фронтенд-кода. Например,

Sass позволяет использовать переменные и функции в CSS, а CoffeeScript – преобразовывает код на языке Ruby в JavaScript.

Минусы:

- **Производительность:** Ruby может быть медленным в сравнении с некоторыми другими языками программирования, такими как C++ или Java. Это может стать проблемой при работе с большими объемами данных или приложениями с высокой нагрузкой.

- **Сложность настройки:** Настройка Ruby может быть сложной, особенно для новичков в этом языке. Это может затруднить начало работы с фреймворком Ruby on Rails.

- **Большое количество файлов:** Ruby on Rails создает много файлов, что может усложнить процесс отслеживания их всех и их взаимосвязей.

В целом, Ruby не является наилучшим выбором для разработки фронтенд-приложений, поскольку имеются более подходящие языки программирования, такие как JavaScript, для работы с браузерами и фронтенд-разработки в целом.

PHP

PHP является языком программирования, который часто используется на сервере для разработки динамических веб-сайтов и веб-приложений. Хотя PHP не является настолько популярным языком для разработки фронтенда, он может использоваться и в этой области.

Рассмотрим плюсы и минусы PHP для разработки фронтенда [19].

Плюсы:

- **Простота использования:** PHP легко изучить и использовать, даже для новичков в программировании. Синтаксис PHP прост и интуитивно понятен.

- **Широкое распространение:** PHP является одним из наиболее популярных языков программирования в мире. Существует множество библиотек и фреймворков, которые делают его использование в фронтенде более удобным и эффективным.

- **Большое сообщество:** существует большое сообщество разработчиков, которые работают с PHP, что означает, что вы можете найти много полезной информации и поддержки в онлайн-сообществах.

Минусы:

- **Производительность:** PHP не является самым быстрым языком, и он может иметь проблемы с производительностью, особенно при работе с большими объемами данных.

- **Безопасность:** PHP может иметь проблемы с безопасностью, особенно при работе с пользовательскими данными. Некоторые уязвимости могут привести к хакерским атакам на ваш сайт.

- **Сложность поддержки:** PHP может быть трудным для поддержки из-за того, что он может иметь несколько версий и требовать различных настроек сервера.

- **Сложность масштабирования:** PHP может быть сложным для масштабирования, особенно если вы разрабатываете большой и сложный проект. Это может привести к проблемам с производительностью и доступностью сайта.

В целом, использование PHP для фронтенда может привести к сложностям в разработке, обслуживании и безопасности приложения. Хотя PHP может быть

полезен для некоторых конкретных задач на стороне клиента, для общих целей разработки фронтенда он не является лучшим выбором.

Вывод по выбору языка программирования

В таблице 2 приведено сравнение языков программирования.

Таблица 2 – Сравнение языков программирования

	JavaScript	Ruby	PHP
Производительность	+	+	–
Простота разработки	+	+	+
Разнообразие инструментов	+	–	+
Большое сообщество	+	–	+
Безопасность	+	+	–

Учитывая все критерии, лучшим выбором для разработки приложения будет JavaScript.

Выбор фреймворка

Существует 3 актуальных фреймворка для фронтенда: React, Angular и Vue. Рассмотрим каждый из них по следующим критериям: производительность, размер библиотеки, обширное сообщество, простота разработки, обширный функционал.

React

Один из самых популярных фреймворков для разработки фронтенда на языке JavaScript. Вот некоторые его плюсы и минусы:

Плюсы:

- Эффективная виртуализация DOM: React использует виртуализацию DOM для обновления только необходимых элементов на странице, вместо полного перестроения страницы при каждом изменении. Это повышает производительность и улучшает пользовательский опыт.

- Компонентный подход: React построен на компонентном подходе, что делает код более организованным, легко переиспользуемым и удобным для тестирования.

- Модульность: React основан на компонентах, что позволяет разбивать большие проекты на более мелкие и управляемые компоненты.

- Отличная документация: у React есть официальная и обширная документация, которая помогает новичкам быстро освоиться с фреймворком.

- Широкое сообщество и поддержка: React имеет одно из самых больших сообществ в веб-разработке, что обеспечивает огромную базу знаний и готовых решений для разработчиков.

- Удобный отладочный инструментарий: React имеет множество инструментов для отладки, включая расширение React Developer Tools, который помогает разработчикам быстро и легко находить ошибки в коде.

- React Native: React Native позволяет создавать приложения для мобильных устройств с использованием React. Это значит, что разработчики могут использовать один и тот же язык программирования (JavaScript) и повторно использовать многие компоненты, что ускоряет разработку мобильных приложений.

Минусы:

- Много способов решения одной задачи: React предоставляет множество способов решения одной и той же задачи, что может быть запутывающим для начинающих разработчиков.

Angular

Фреймворк для разработки веб-приложений на JavaScript и TypeScript. Вот некоторые из его преимуществ и недостатков:

Плюсы Angular:

- Компонентный подход: Angular основывается на компонентной архитектуре, что облегчает разработку, тестирование и поддержку кода.

- Декларативный язык разметки: Angular использует HTML как язык разметки, что упрощает процесс создания пользовательского интерфейса.

- TypeScript: Angular разрабатывается на TypeScript, который является надмножеством JavaScript и добавляет множество возможностей, таких как статическая типизация, интеллектуальное автодополнение и дополнительные функции для объектно-ориентированного программирования.

- RxJS: Angular поставляется с встроенной библиотекой RxJS, которая облегчает работу с асинхронными потоками данных.

- Инструменты для разработки: Angular имеет широкий выбор инструментов для разработки, таких как Angular CLI, который облегчает создание, тестирование и развертывание приложений.

Минусы Angular:

- Размер: Angular имеет довольно большой размер, что может замедлить загрузку страницы.

- Сложность: иногда компонентный подход Angular может привести к чрезмерной сложности кода, особенно в случае больших приложений.

- Зависимость от TypeScript: хотя TypeScript является преимуществом Angular, это также может стать недостатком, для начинающих разработчиков.

Vue

Прогрессивный JavaScript фреймворк для создания пользовательских интерфейсов. Ниже перечислены некоторые плюсы и минусы Vue.js:

Плюсы:

- Легкий и быстрый: Vue.js имеет очень маленький размер, что ускоряет процесс загрузки страницы и повышает производительность.

- Простота использования: Vue.js имеет легковесный API и документацию, которые позволяют быстро и легко начать работу.
- Реактивность: Vue.js позволяет автоматически обновлять пользовательский интерфейс при изменении данных в модели.
- Компонентная архитектура: Vue.js позволяет создавать компоненты, которые могут быть повторно использованы в разных местах приложения.
- Гибкость: Vue.js можно использовать как самостоятельную библиотеку, так и сочетать с другими библиотеками и фреймворками.

Минусы:

- Ограниченный функционал: Vue.js не имеет такой большой и разнообразной экосистемы, как у других фреймворков, таких как Angular или React.
- Не всегда подходит для больших проектов: Vue.js может достаточно хорошо масштабироваться для средних проектов, но может быть сложно использовать его для очень больших проектов с множеством разработчиков и сложной логикой.
- Ограниченная поддержка: Vue.js является более новым фреймворком, чем Angular или React, и может быть менее поддерживаемым и обновляемым в будущем.

Вывод по выбору фреймворка

React, Angular и Vue — это все мощные фреймворки/библиотеки для разработки фронтенд-приложений. Они имеют множество схожих и различных функций, а также плюсы и минусы. Однако, каждый из них может быть более подходящим для определенных проектов, в зависимости от конкретных потребностей.

В таблице 3 приведено сравнение языков программирования.

Таблица 3 – Сравнение языков программирования

	React	Angular	Vue
производительность	+	+	+
размер библиотеки	+	–	+
обширное сообщество	+	+	–
простота разработки	+	–	+
обширный функционал	+	+	–

Вывод по анализу технологических решений

Проведя анализ предметной области, можно сделать следующие заключения:

1. JavaScript — это один из наиболее удобных языков программирования в веб-разработке на сегодняшний день, что делает его хорошим выбором для веб-приложений.

2. React — это очень популярный фреймворк для фронтенд-разработки, который позволяет разработчикам создавать масштабируемые и высокопроизводительные веб-приложения.

3. Express — это быстрый и легкий веб-фреймворк для серверной разработки, написанный на JavaScript. Он позволяет разработчикам создавать API и веб-приложения на основе Node.js.

4. PostgreSQL — это мощная реляционная база данных, которая имеет широкий набор функций и возможностей, включая поддержку JSON, географических данных и многое другое. Она также является одним из самых надежных и масштабируемых решений для хранения данных.

В целом, выбранный стек технологий – JavaScript, React, Express и PostgreSQL – является очень мощным и эффективным для разработки современных веб-приложений. Он обеспечивает высокую скорость разработки, хорошую производительность и масштабируемость, а также поддерживается большим сообществом разработчиков.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Функциональные и нефункциональные требования используются в процессе разработки программного обеспечения для определения того, каким должно быть окончательное решение. Функциональные требования описывают, что должно делать программное обеспечение, то есть какие функции и возможности оно должно иметь. Нефункциональные требования описывают, как программное обеспечение должно работать, то есть какие должны быть характеристики качества.

2.1 Функциональные требования

- Клиентская часть должна быть реализована в виде веб-приложения, запускаемого в браузере, и представлена в виде трехмерной карты вуза
 - Реализовать механизм поиска аудиторий по названию (номеру) с указанием на карте
 - Реализовать административный интерфейс наполнения карты нужной информацией и ее актуализации операторами (добавление корпусов, аудиторий и т.д.)
 - Веб-приложение должно поддерживать GET-запросы с параметрами поиска, чтобы сразу по URL отобразить выделенную аудиторию
 - Взаимодействие клиентской и серверной сторон происходит посредством REST API запросов

2.2 Нефункциональные требования

- Клиентское веб-приложение должно работать во всех популярных на момент разработки браузерах: Mozilla Firefox (не ниже версии 58.0.2), Opera (не ниже версии 13), Google Chrome (не ниже версии 64.0.3282), Microsoft Edge (не ниже версии 25.10586), а также в их мобильных версиях.
 - Система управления базами данных PostgreSQL
 - Реализация проекта в виде веб-приложения, развернутого на сервере вычислительного центра УИ ЮУрГУ (Ubuntu Server 22.04)

3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

3.1 Архитектура предлагаемого решения

Приложение будет построено по следующей архитектуре: на клиентской стороне пользователь будет взаимодействовать с SPA (Single Page Application или одностраничное приложение), данные для которого будут получены при помощи REST API на сервере, графическое представление архитектуры на рисунке 6.

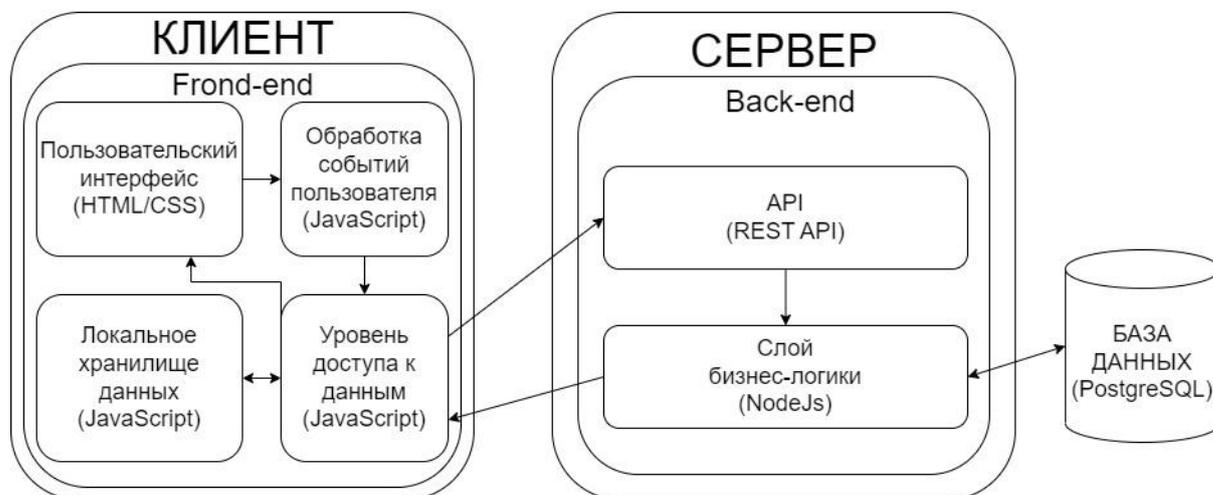


Рисунок 6 – Графическое представление архитектуры

REST API будет обеспечивать взаимодействие приложения и сервера при помощи протокола HTTP. При такой архитектуре взаимодействие с сервером осуществляется четырьмя операциям:

- получение данных с сервера
- добавление новых данных на сервер
- изменение данных
- удаление данных

Каждой из этих операций соответствует свой метод HTTP-запроса: GET, POST, PUT и DELETE соответственно.

SPA – веб-приложение, которое работает на одной HTML-странице. Все необходимые шаблоны страниц, JavaScript и CSS файлы подгружаются при первой загрузке, в дальнейшем взаимодействие приложения и сервера сводится к минимуму. Большая часть работы приложения производится на устройстве пользователя, а сервер только отправляет необходимые данные, обычно в формате JSON.

В одностраничных приложениях отображение полученных пользователем данных осуществляется динамически при помощи AJAX (Asynchronous JavaScript And XML), который представляет собой набор методов разработки и технологий и позволяет веб-приложению совершать свою работу асинхронно, то есть при обновлении конкретной части страницы не будет необходимости перезагружать её всю.

Основным преимуществом такой архитектуры является то, что одностраничные приложения создают у пользователя впечатление работы с

walls	JSON	координаты для построения аудитории
-------	------	-------------------------------------

floors – содержит информацию для хранения данных полов, столбцы приведены в таблице 5.

Таблица 5 – Таблица “floors”

Название столбца	Тип данных	Описание
id	INT	Идентификатор пола, первичный ключ
floor	INT	этаж пола
points	JSONB[]	координаты для построения пола

walls – содержит информацию для хранения данных стен, столбцы приведены в таблице 6.

Таблица 6 – Таблица “walls”

Название столбца	Тип данных	Описание
id	INT	Идентификатор стены, первичный ключ
floor	INT	этаж стены
points	JSONB[]	координаты для построения стены
isPillarBefore	BOOL	колонна перед началом стены
isPillarAfter	BOOL	колонна после окончания стены

users – содержит информацию для хранения данных аудиторий, столбцы приведены в таблице 7.

Таблица 7 – Таблица “users”

Название столбца	Тип данных	Описание
id	INT	Идентификатор пользователя, первичный ключ
login	VARCHAR	логин пользователя
password	VARCHAR	пароль пользователя в виде хэша
isAdmin	BOOL	свойство для режима администратора

sprites – содержит информацию для хранения данных изображения лестниц, лифтов, кафе и тд, столбцы приведены в таблице 8.

Таблица 8 – Таблица “sprites”

Название столбца	Тип данных	Описание
id	INT	Идентификатор изображения, первичный ключ
floor	INT	этаж изображения
position	INT[]	координаты расположения
spriteTypeId	INT	тип изображения

sprite_types – содержит информацию для хранения данных типов изображений, столбцы приведены в таблице 9.

Таблица 9 – Таблица “sprite_types”

Название столбца	Тип данных	Описание
id	INT	Идентификатор типа изображения, первичный ключ
name	VARCHAR	название типа изображения
img	VARCHAR	url изображения, хранящегося на сервере

4. РЕАЛИЗАЦИЯ КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТЕЙ

На рисунке 8 представлена главная страница приложения.



Рисунок 8 – Главная страница приложения

На главной странице приложения расположены трёхмерная карта, элемент выбора этажа и элемент с поиском аудитории.

Трёхмерная карта представляет собой трёхмерные представления корпусов с аудиториями, которые располагаются на картографическом изображении университетского кампуса.

Элемент выбора этажа расположен справа по центру основного окна приложения. Содержит секции выбора этажа, при клике на которые загружаются данные выбранного этажа.

Элемент поиска аудитории содержит строку поиска, в которую вводится искомая аудитория. Ниже расположены элементы “радиокнопки” с выбором нужного корпуса. Справа расположена кнопка, запускающая процесс поиска аудитории, и если искомая аудитория найдена, то камера пользователя перемещается к найденной выделенной аудитории, пример на рисунке 9.

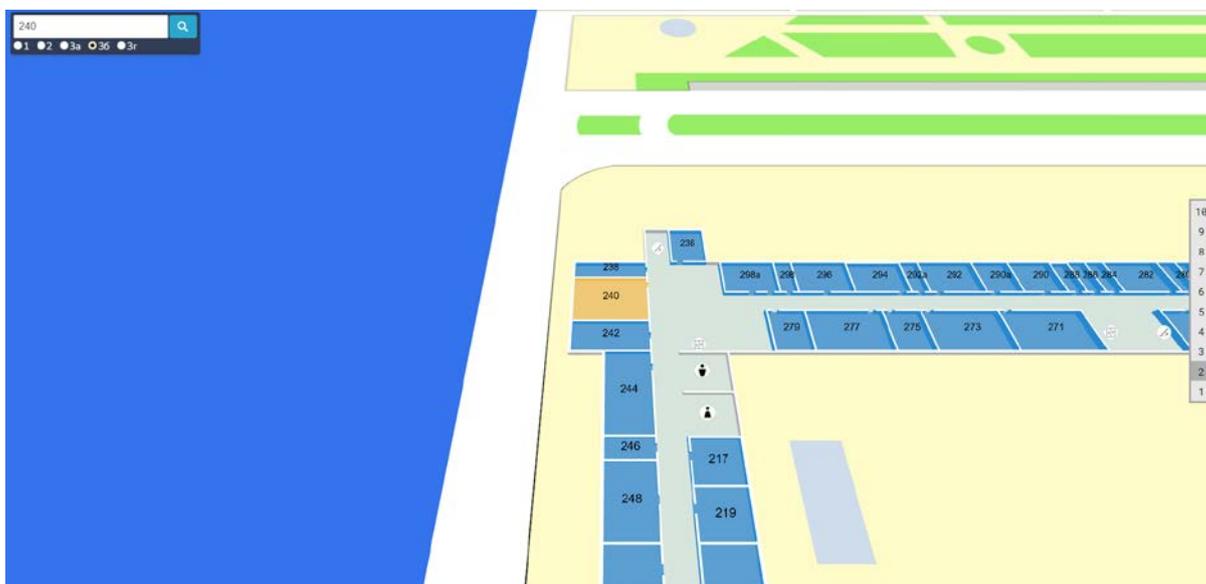


Рисунок 9 – Пример найденной аудитории 240 корпуса 3б

Также если приписать к основному url-адресу /admin, откроется страница авторизации, пример на рисунке 10.

Логин

Пароль

Войти

Рисунок 10 – Форма на странице авторизации

Далее рассмотрим функции администраторской части клиента.

При успешной авторизации на главную страницу добавляется элемент выпадающего списка с возможностью выбора редактируемых элементов, в числе которых находятся: аудитория, стена, пол, спрайт, тип спрайта, пример на рисунке 11.

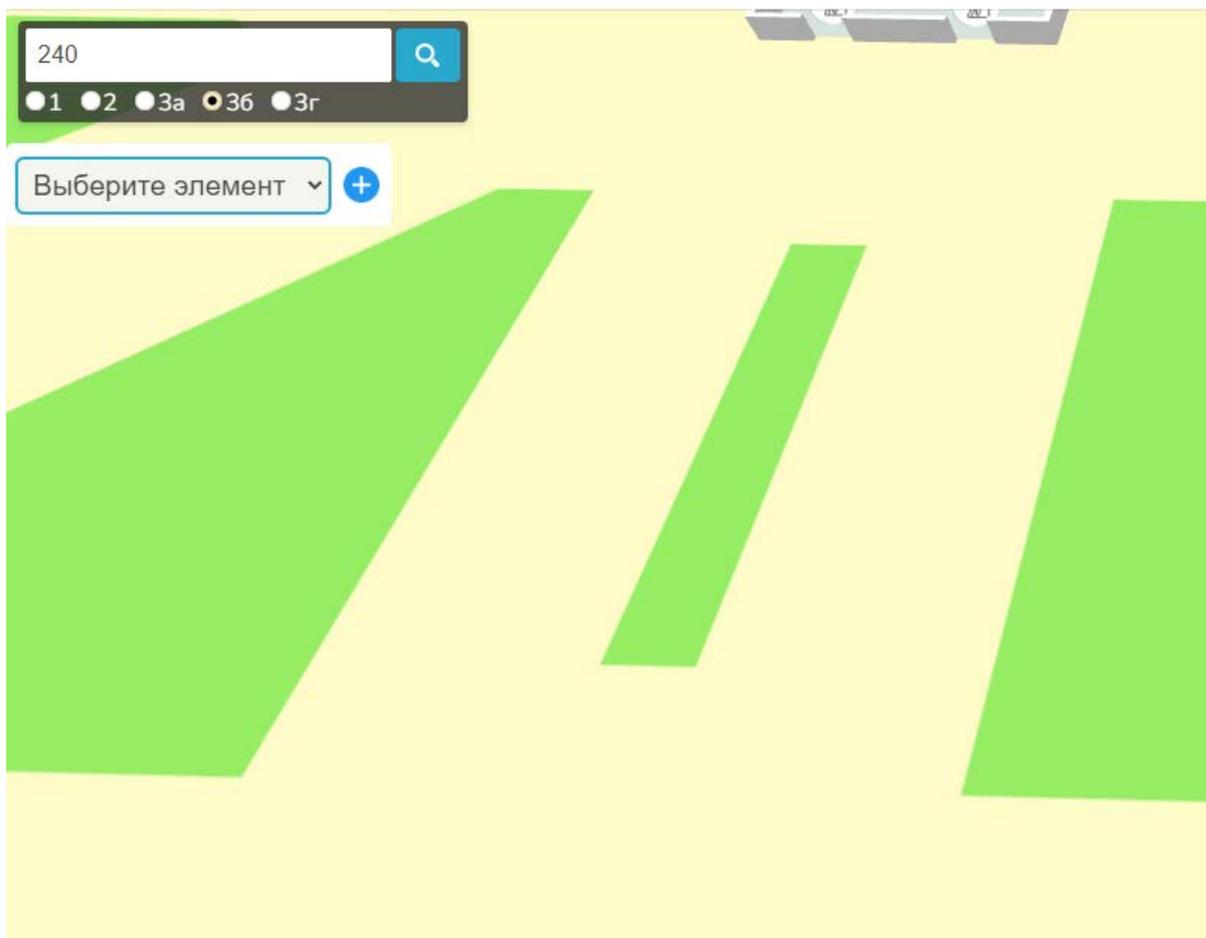


Рисунок 11 – Элемент выбора редактируемого элемента

Далее рассмотрим демонстрацию редактирования аудитории, пример на рисунке 12.

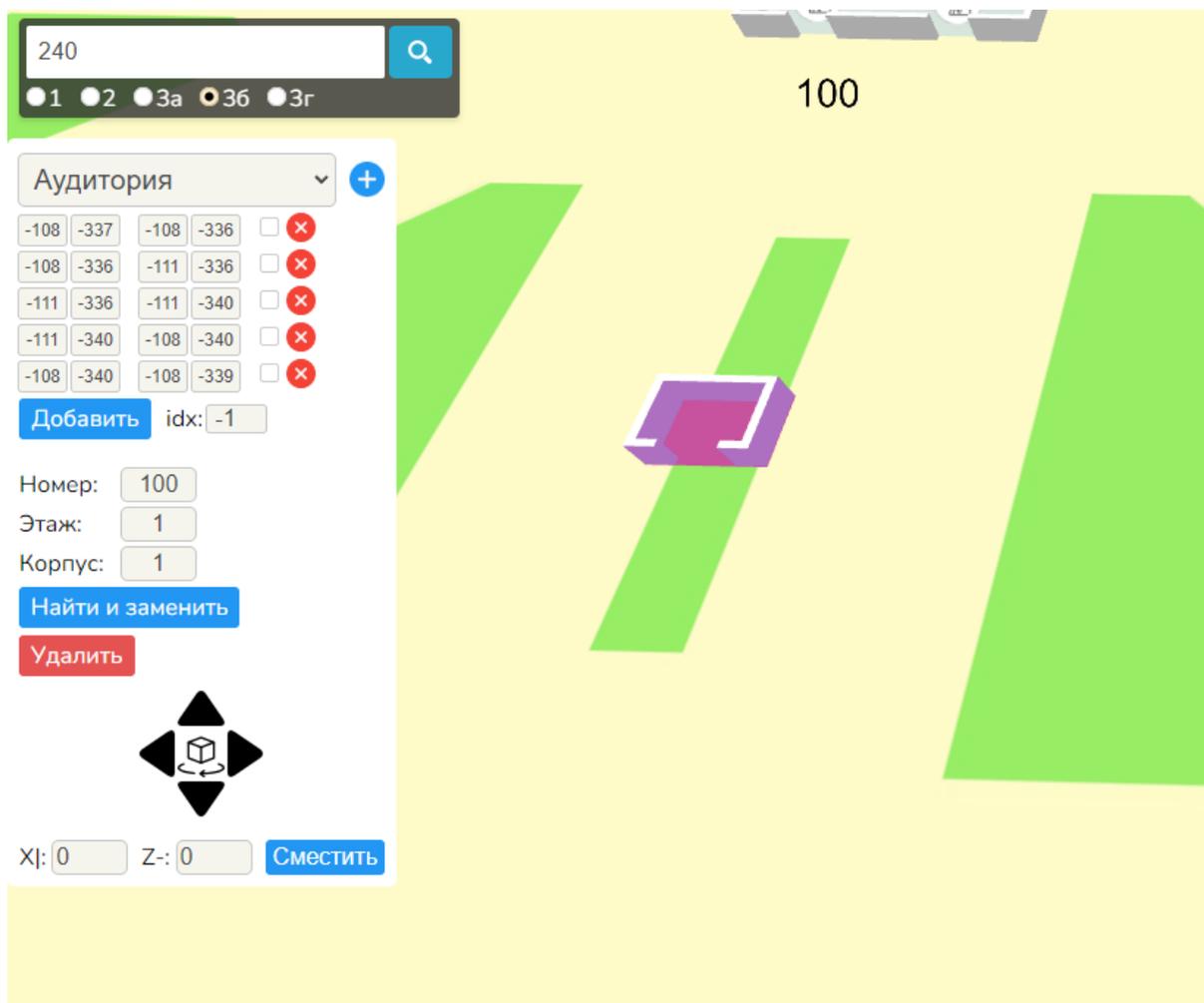


Рисунок 12 – Окно редактирования с выбранным объектом “Аудитория”

Первый элемент окна редактирования содержит координаты стен, при помощи которых строится аудитория, пример на рисунке 13.

-108	-337	-108	-336	<input type="checkbox"/>	✖
-108	-336	-111	-336	<input type="checkbox"/>	✖
-111	-336	-111	-340	<input type="checkbox"/>	✖
-111	-340	-108	-340	<input type="checkbox"/>	✖
-108	-340	-108	-339	<input type="checkbox"/>	✖
Добавить					idx: -1

Рисунок 13 – Редактирование координат аудитории

Каждая строка является отдельной стеной. Первая пара чисел – это начало стены, вторая пара – конец.

Правее второй пары чисел расположен флажок двух состояний, который регулирует является ли стена смежной с другой стеной. Данный параметр был введён для улучшения оптимизации.

Правее предыдущего элемента расположена кнопка, удаляющая соответствующую её строке стену.

Внизу расположены кнопка и поле ввода. Поле ввода содержит индекс строки, после которой при нажатии на кнопку добавится новая стена.

Следующая часть окна редактирования содержит информацию корпуса, этажа и номера аудитории, пример на рисунке 14.

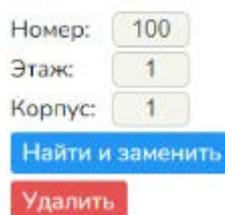


Рисунок 14 – Элемент редактирования номера, этажа и корпуса аудитории

Кнопка с надписью “Найти и заменить” находит в базе данных аудиторию по предоставленным параметрам, и обновляет строки координат на соответствующие найденному объекту.

Нижняя часть окна редактирования содержит элемент перемещения редактируемой аудитории, пример на рисунке 15.



Рисунок 15 – Элемент, отвечающий за перемещение редактируемого объекта

После окончания редактирования, чтобы добавить объект в базу данных, необходимо нажать на кнопку справа от выпадающего списка.

Большинство объектов имеют похожий функционал задания координат, параметров и перемещения объекта.

Следующий редактируемый объект – стена, пример окна редактирования на рисунке 16.

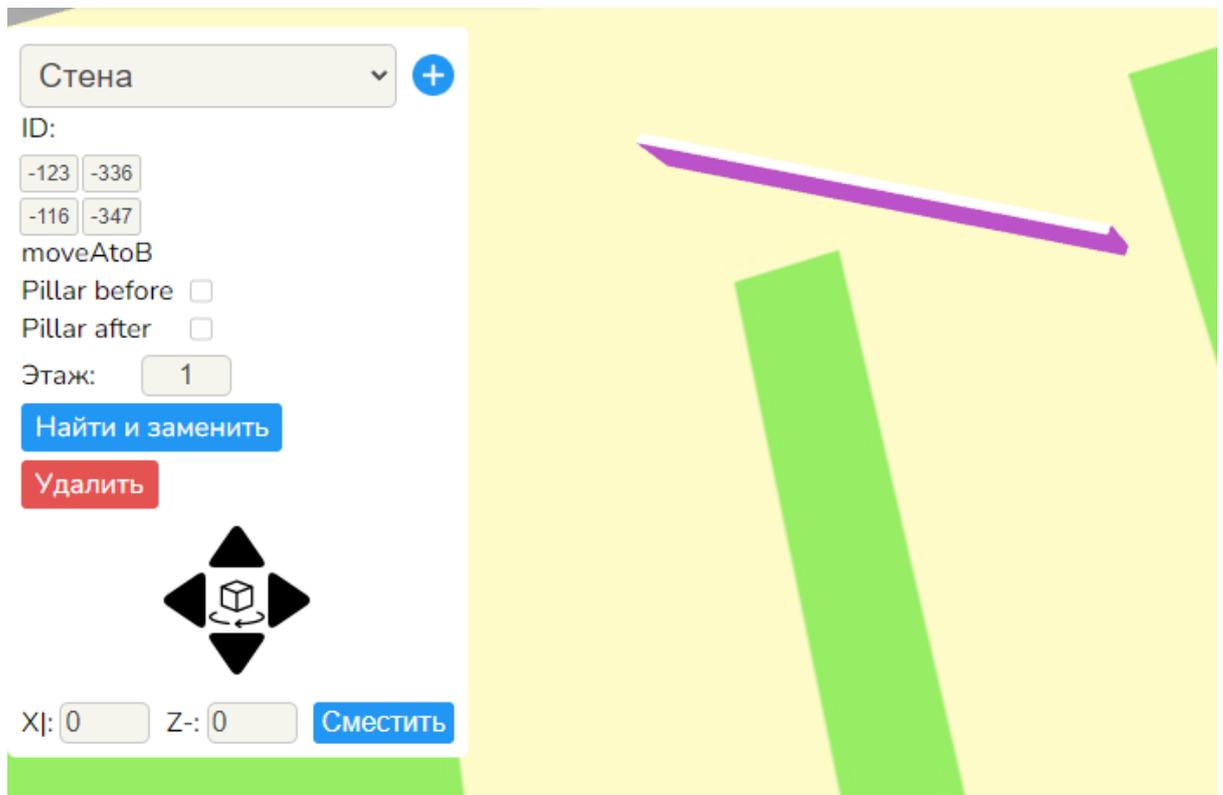


Рисунок 16 – Окно редактирования с выбранным объектом “Стена”

У данного объекта поле координат представлено измененным образом: верхняя строка и нижняя строка – это пары X Y координат начала и конца стены.

Из особенностей редактирования добавлены два поля “Pillar before” и “Pillar after”. Они необходимы для корректного отображения соединения при стыке с стен между собой.

Кнопка “moveAtoB” перемещает стену на её длину в сторону строки нижних координат.

Следующий редактируемый объект – пол, пример окна редактирования на рисунке 17.

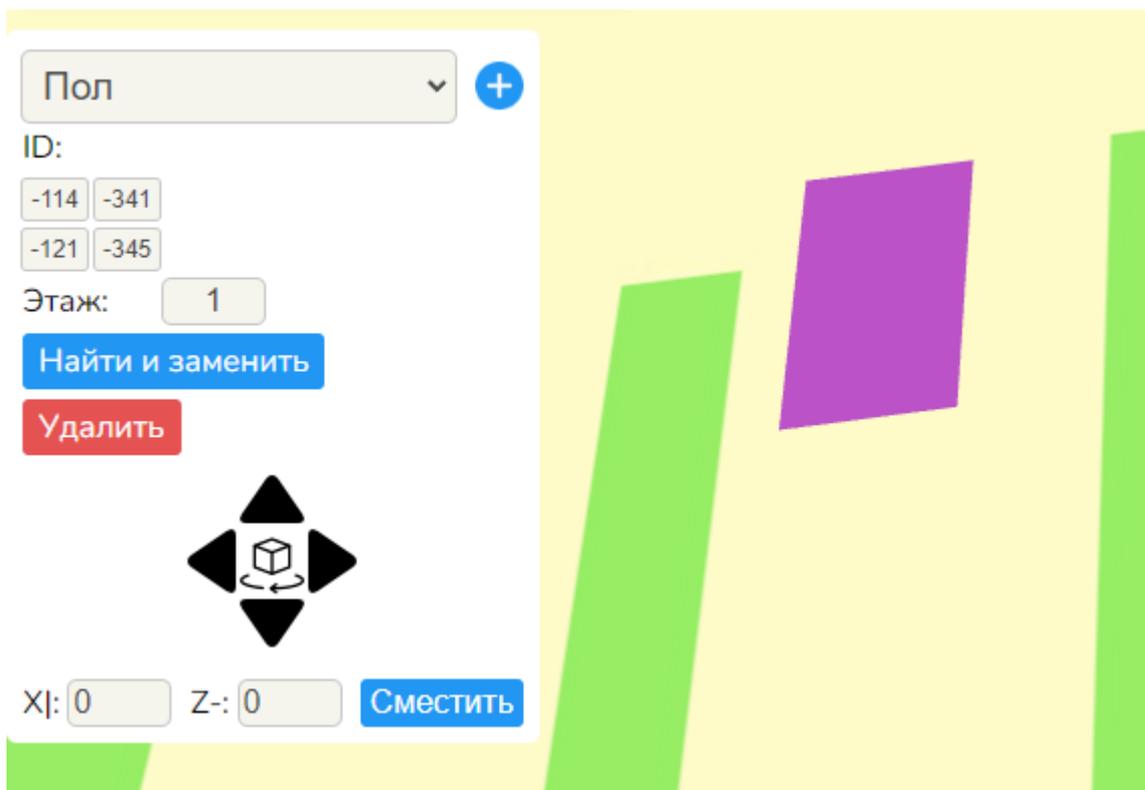


Рисунок 17 – Окно редактирования с выбранным объектом “Пол”

У данного объекта поле координат представлено измененным образом: верхняя строка и нижняя строка — это пары X Y координат противоположных углов четырёхугольника.

Следующий редактируемый объект – спрайт, пример окна редактирования на рисунке 18.

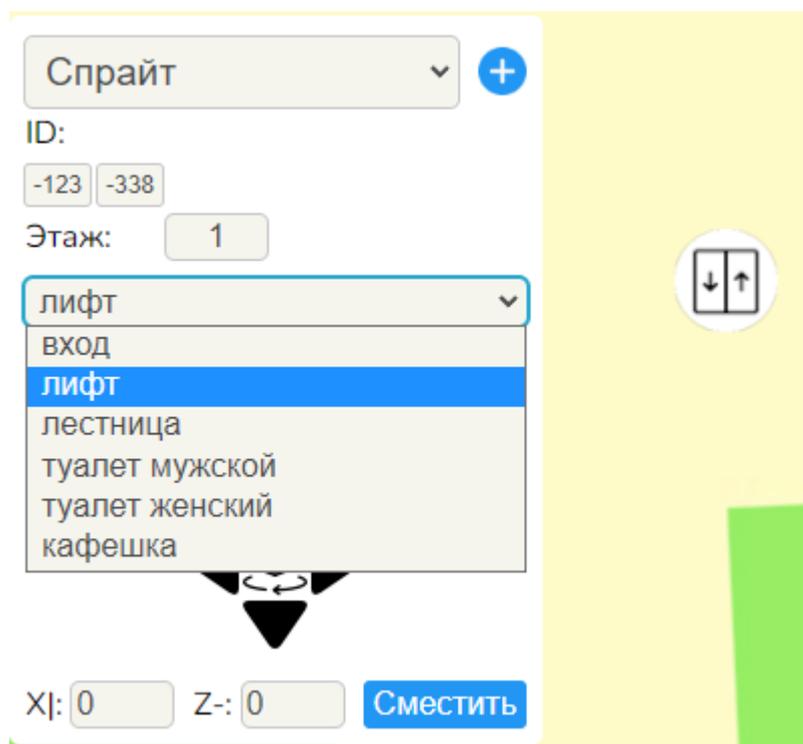


Рисунок 18 – Окно редактирования с выбранным объектом “Спрайт”

У данного объекта поле координат имеет одну строку.

Из особенностей редактирования добавлен выпадающий список существующих типов спрайтов.

Следующий редактируемый объект – тип спрайтов, пример окна редактирования на рисунке 19.

The screenshot shows a web interface for editing a 'Type of Sprites' object. At the top, there is a dropdown menu labeled 'Тип спрайтов' with a blue plus icon to its right. Below this is an 'ID:' label followed by a text input field and a 'Очистить' button. A list of six sprite types follows: 'вход', 'лифт', 'лестница', 'туалет мужской', 'туалет женский', and 'кафешка'. Each type has a blue 'Изменить' button and a red 'x' button to its right. Below the list is a 'Название:' label and a text input field. At the bottom, there is an 'Изображение:' label, a 'Выберите файл' button, and a 'Файл...бран' link.

Рисунок 19 – Окно редактирования с выбранным объектом “Тип спрайтов”

Окно редактирования данного объекта содержит список существующих спрайтов с возможностью редактирования. Ниже списка находится поле ввода для названия редактируемого спрайта и поле для загрузки изображения, которое будет отображаться на спрайте.

Кнопка “очистить” обнуляет поля “Название” и “ID”.

5 ТЕСТИРОВАНИЕ

5.1 Проведение процедуры тестирования

В ходе тестирования была проверена корректная работа основных наиболее важных элементов системы, а именно: поиск аудитории из окна браузера, поиск аудитории через url параметры и переключение этажей.

1. Тестирование функции поиска аудитории из окна браузера

Протестируем на примере поиска аудитории 240 корпуса 3б, введя соответствующие значения в поля ввода. Из рисунков 20 и 21 видно корректное состояние приложения при тестировании функции поиска.

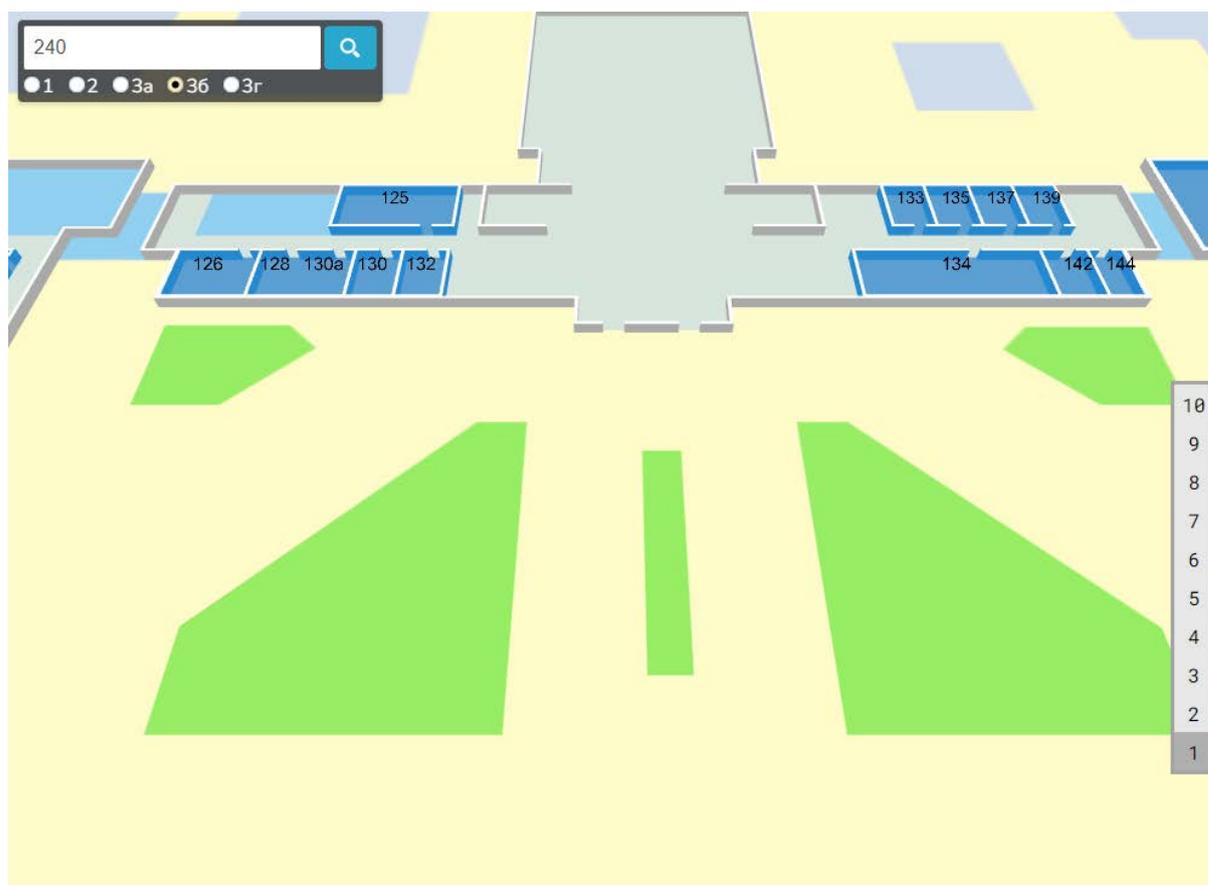


Рисунок 20 – Состояние окна браузера до нажатия кнопки поиска



Рисунок 21 – Состояние окна браузера после нажатия кнопки поиска

2. Тестирование функции поиска аудитории через url параметры

Для тестирования данной функции добавляем к ссылке адреса сайта значения параметров корпуса и номера аудитории: «<http://localhost:3001/?corpus=3б&number=227а>». После загрузки страницы окно приложения будет указывать на выделенную искомую аудиторию, пример на рисунке 22.

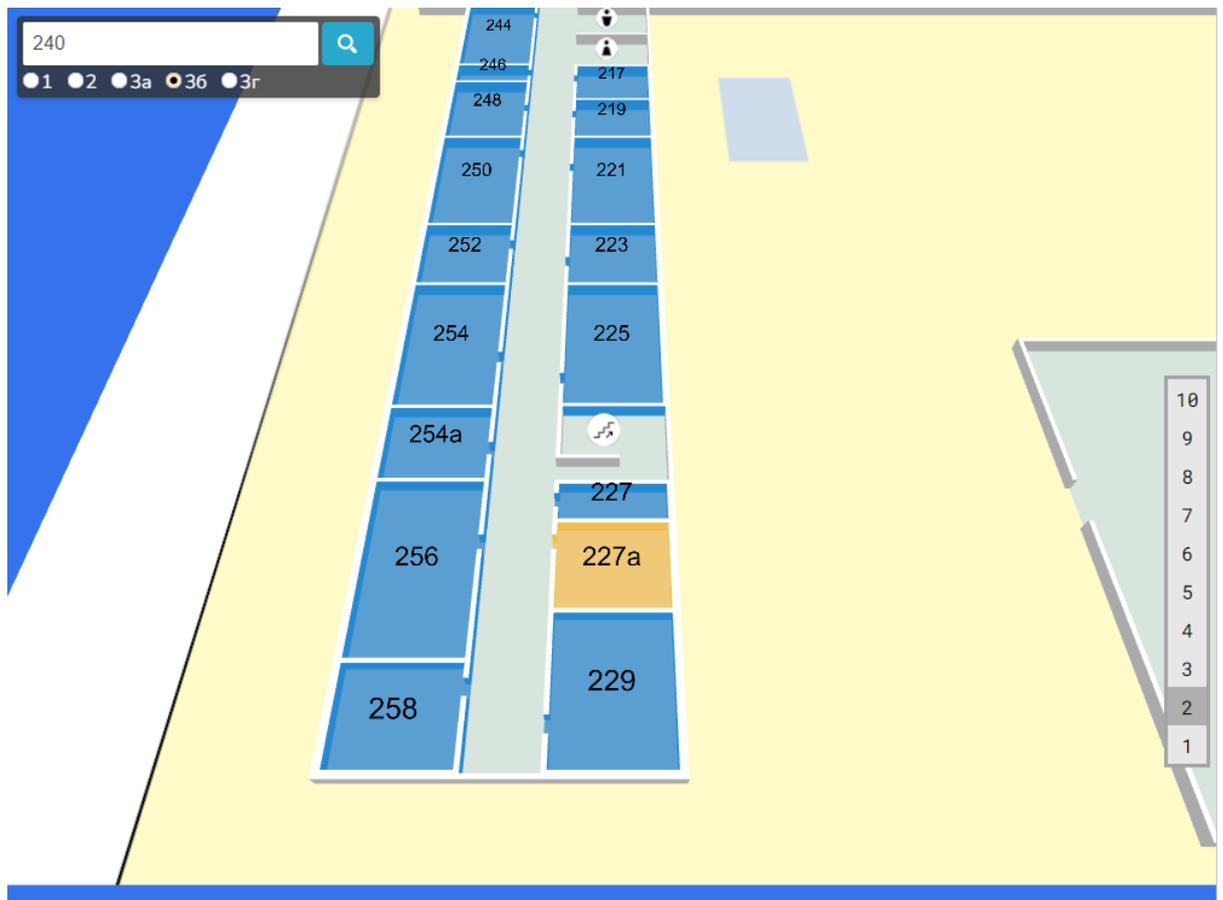


Рисунок 22 – Состояние окна браузера после загрузки страницы с параметрами поиска

3. Тестирование функции переключения этажей

Протестируем на примере переключения между первым и вторым этажами с видом на 3а, 3б и 3бв корпуса. Из рисунков 23 и 24 видно корректное состояние приложения при тестировании функции переключения этажей.

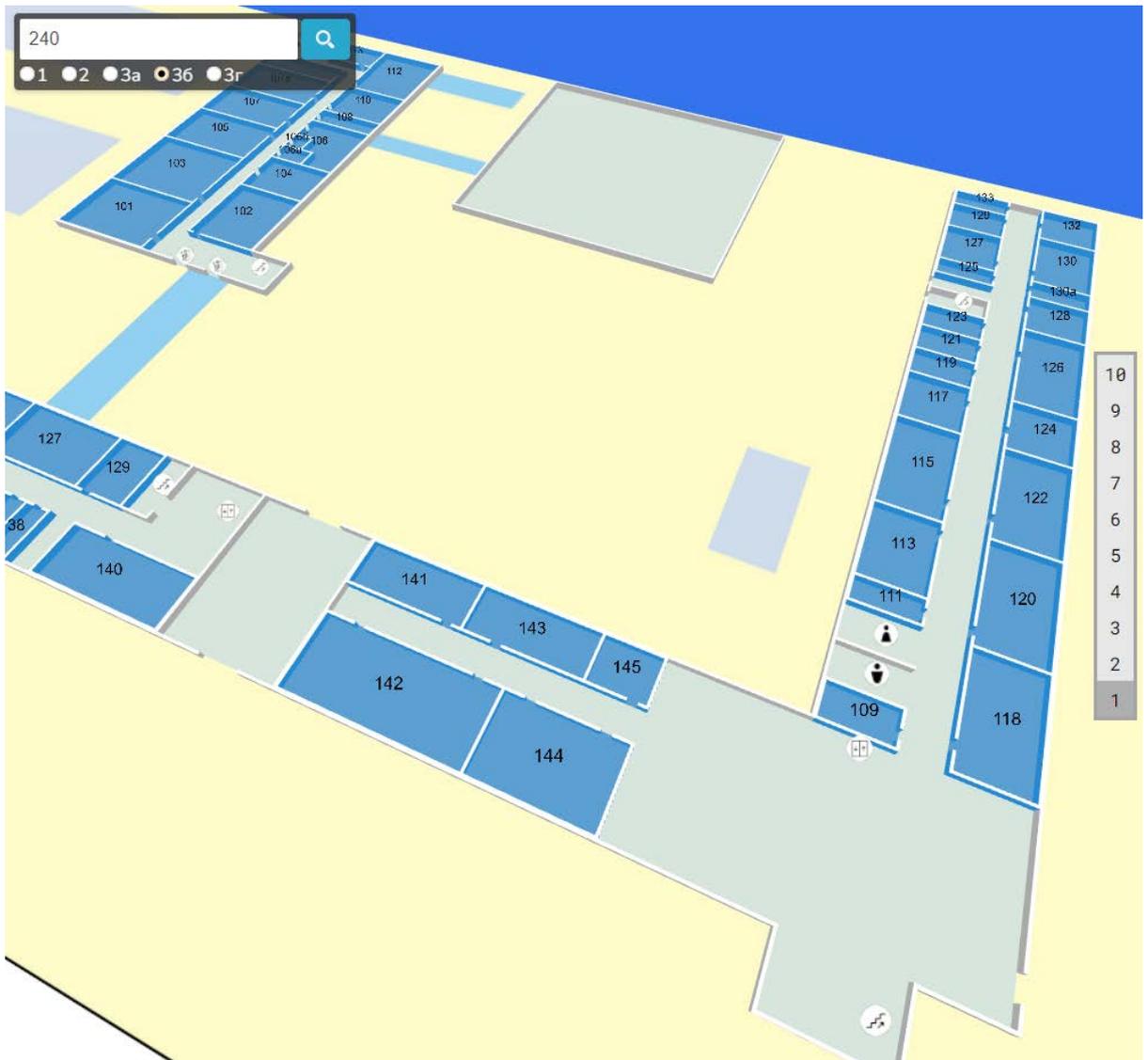


Рисунок 23 – Состояние окна браузера при значении “1” поля переключения этажей

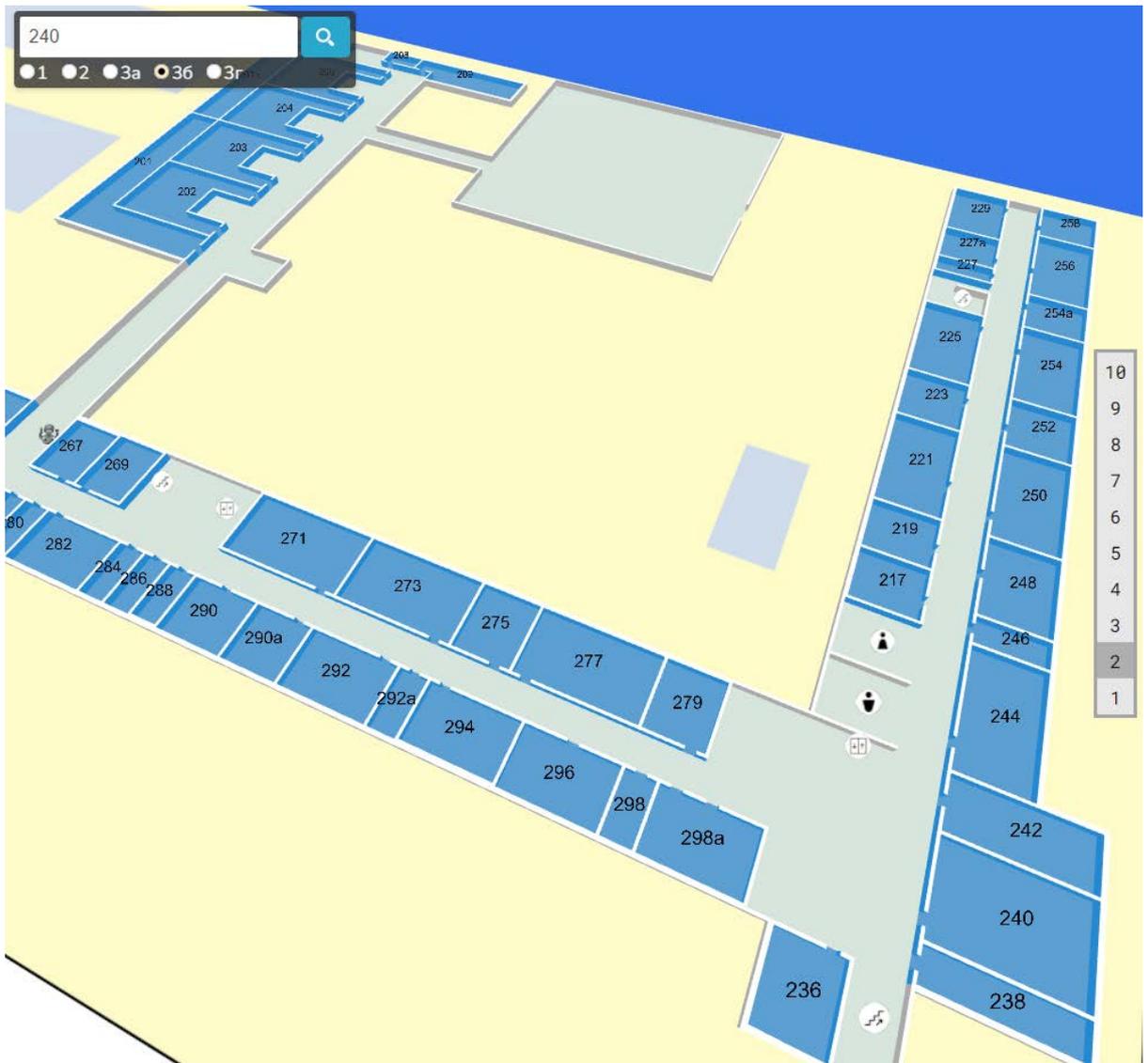


Рисунок 24 – Состояние окна браузера при значении “2” поля переключения этажей

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы было разработано веб-приложение интерактивной карты аудиторного фонда университета. Для достижения данной цели были решены следующие задачи:

1. Анализ предметной области
2. Определение требований
3. Проектирование системы
4. Реализация клиентской и серверной частей
5. Тестирование

Поставленные задачи выполнены успешно, цель достигнута. Удобный интерфейс веб-приложения позволяет с лёгкостью ориентироваться в кампусе НИУ «ЮУрГУ».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Кампус [Электронный ресурс]: Материал из Википедии – Режим доступа: <http://ru.wikipedia.org/?oldid=77047395>

2 Интерактивные карты [Электронный ресурс] // Словарь и энциклопедии на Академикe – Режим доступа: <http://dic.academic.ru/dic.nsf/ruwiki/641413>

3 Smashig journal [Электронный ресурс] // Интерактивные карты в среде современного веб-дизайна – Режим доступа: <http://smashingjournal.ru/?p=1565>

4 Kirsch A. [Электронный ресурс] // A New Interactive Map of the Vilna Ghetto Asks: What Good Is History if It Isn't Told? [Электронный ресурс] – Режим доступа: <http://www.tabletmag.com/jewish-life-and-religion/204434/daf-yomi-168>

5 История города Кливленд [Электронный ресурс] – Режим доступа: <http://clevelandhistorical.org/>

6 Геоинформационные системы [Электронный ресурс] // Тамбовский областной центр новых информационных технологий – Режим доступа: <http://gis.web.tstu.ru/chtogis2.htm>

7 Кычкин А.В. Геоинформационные системы: учеб. пособие. [Электронный ресурс] / Кычкин А.В. // Издательство Пермского национального исследовательского политехнического университета – Режим доступа: <http://edu.dvgups.ru/METDOC/ITS/GEOD/LEK/118/L2.html>

8 Составные части ГИС [Электронный ресурс] // Издательство Петрозаводского государственного университета – Режим доступа: http://loi.sccc.ru/gis/pgu_giskurs/gis15_a.htm

9 Веб-сайт «VibeLab». [Электронный ресурс]. – URL: <https://vibelab.etu.ru/projectSample.html?smart-map>. (Дата обращения: февраль 2023 года).

10 Веб-сайт «МГИМО». [Электронный ресурс]. – URL: <https://mgimo.ru/territory/map/>. (Дата обращения: февраль 2023 года).

11 Веб-сайт «МИИГАиК» [Электронный ресурс]. – URL: https://dev.kartfak.ru/miigaik_plan/#map=8.55/55.5032/37.5&l=2. (Дата обращения: февраль 2023 года).

12 Веб-сайт «MapBox GL JS». [Электронный ресурс]. – URL: <https://www.mapbox.com/mapbox-gl-js>. (Дата обращения: февраль 2023 года).

13 Веб-сайт «2ГИС Этажи». [Электронный ресурс]. – URL: <https://docs.2gis.com/ru/mapgl/examples/floors>. (Дата обращения: февраль 2023 года).

14 Веб-сайт «LOW-LEVEL 3D GRAPHICS API BASED ON OPENGL ES». [Электронный ресурс]. – URL: <https://www.khronos.org/webgl/>. (Дата обращения: февраль 2023 года).

15 Веб-сайт «Яндекс.Карты». [Электронный ресурс]. – URL: <https://yandex.ru/maps/>. (Дата обращения: февраль 2023 года).

16 Веб-сайт «1PS». [Электронный ресурс]. – URL: <https://1ps.ru/blog/sites/2021/osobennosti-javascript-zachem-uchit-etot-yazyik-programmirovaniya-i-gde-on-prigoditsya/>. (Дата обращения: май 2023 года).

17 Веб-сайт «LiquidHub». [Электронный ресурс]. – URL: <https://liquidhub.ru/blogs/blog/yazyk-programmirovaniya-ruby>. (Дата обращения: май 2023 года).

18 Веб-сайт «plusminusi». [Электронный ресурс]. – URL: <https://plusminusi.ru/yazyk-programmirovaniya-ruby-stoit-li-izuchat-plyusy-i-minusy/>. (Дата обращения: май 2023 года).

19 Веб-сайт «Skillbox media». [Электронный ресурс]. – URL: https://skillbox.ru/media/code/vybiraem_yazyk_programmirovaniya_chno_nuzhno_znat_o_php/. (Дата обращения: май 2023 года).