

Управление образования г. Пензы

МБОУ «Лицей современных технологий управления № 2» г. Пензы

Конструктор интерактивных планов помещений в зданиях

Выполнили:

Коваленко Иван Сергеевич,
Искоркин Роман Александрович,
Лисин Григорий Сергеевич
обучающиеся МБОУ ЛСТУ №2 г. Пензы

Руководители:

Адамский Сергей Сергеевич,
учитель информатики,
педагог-методист МБОУ ЛСТУ №2 г. Пензы

Пенза, 2026 г

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. АНАЛИЗ И ПРОЕКТИРОВАНИЕ	5
1.1. Обзор существующих решений для интерактивных планов зданий.....	5
1.2. Формирование базы данных JSON для помещений и связей между ними	6
2. РАЗРАБОТКА ВЕБ-ИНТЕРФЕЙСА ПЛАНА	7
2.1. Архитектура веб-приложения и выбор технологического стека (HTML, CSS, JavaScript, SVG).....	7
2.2. Визуализация плана помещений с использованием веб-технологий.....	8
2.3. Реализация функций поиска помещений и отображения информации	9
3. РЕАЛИЗАЦИЯ АЛГОРИТМА НАВИГАЦИИ	10
3.1. Представление плана здания в виде графа для алгоритмов поиска пути	10
3.2. Разработка алгоритма построения оптимального маршрута.....	10
3.3. Реализация построения пути для эвакуации	11
3.4. Тестирование веб-приложения и алгоритма навигации	12
3.5. Анализ результатов тестирования и рекомендации по дальнейшему развитию.....	13
ЗАКЛЮЧЕНИЕ	13
СПИСОК ЛИТЕРАТУРЫ.....	14

ВВЕДЕНИЕ

Современные многофункциональные здания, такие как административные центры, учебные заведения или бизнес-комплексы, часто отличаются сложной планировкой с множеством этажей, коридоров и специализированных помещений. Эта архитектурная сложность создаёт существенные трудности для посетителей и сотрудников при поиске нужных кабинетов, особенно при первичном посещении объекта. В условиях массовых мероприятий, например во время конференций, проблема навигации по зданию усугубляется, приводя к значительным временным потерям и психологическому дискомфорту. Традиционные средства навигации, включая статические указатели и бумажные схемы, не обеспечивают достаточной гибкости и оперативности при изменении маршрутов или временном закрытии помещений. Особенно критична эта проблема в экстренных ситуациях, когда требуется быстро найти альтернативные пути эвакуации. Несовершенство существующих решений негативно сказывается на общей эффективности использования пространства и безопасности людей, что подтверждает необходимость разработки современных цифровых инструментов навигации.

В качестве технологического решения предлагается одностраничное веб-приложение с интерактивной схемой этажей: для каждого этажа используется фоновое изображение плана (PNG/JPG), а маршруты и граф навигации отображаются поверх в отдельном SVG-слое. Приложение включает поиск помещений по названию и построение оптимального маршрута между точками, а также отдельный режим эвакуации. Использование веб-стандартов (HTML, CSS, JavaScript) обеспечивает кроссплатформенность и работу без установки дополнительного программного обеспечения.

Целью данной работы является создание функционирующего прототипа интерактивного плана на базе веб-технологий, реализующего алгоритмы автоматизированной навигации по графу помещений. Практическая значимость

проекта заключается в предоставлении готового решения, адаптируемого под конкретные архитектурные особенности зданий.

Актуальность разработки обусловлена тенденциями урбанизации и ростом числа сложных многофункциональных пространств, где эффективная навигация становится критически важным элементом инфраструктуры. Цифровизация ориентации в помещениях соответствует современным требованиям к доступности общественных зданий и комфорту пользовательского опыта. Особую значимость проект приобретает в контексте создания безбарьерной среды для маломобильных групп населения, нуждающихся в детализированной информации о путях перемещения. Развитие веб-технологий и алгоритмов поиска пути открывает новые возможности для создания интеллектуальных систем навигации, превосходящих по функциональности традиционные аналоги. Внедрение таких решений способствует оптимизации потоков посетителей, снижению эксплуатационных затрат и повышению уровня безопасности в зданиях. Данный проект вносит вклад в это направление, предлагая практическую реализацию с открытым исходным кодом для дальнейшего развития.

1. АНАЛИЗ И ПРОЕКТИРОВАНИЕ

1.1. Обзор существующих решений для интерактивных планов зданий

Современные платформы для визуализации архитектурных объектов классифицируются по используемой графической технологии: WebGL обеспечивает аппаратно-ускоренную трехмерную отрисовку, SVG представляет векторное DOM-ориентированное визуальное представление, а Canvas предоставляет растровый контекст для непосредственного рисования пикселей. WebGL подходит для сложных 3D-сцен и моделирования освещения, однако требует оптимизации для работы на мобильных устройствах. SVG обеспечивает высокую четкость масштабирования и тесную интеграцию с HTML и CSS, что упрощает обработку событий и стилизацию элементов планов. Canvas даёт высокую гибкость при рендеринге произвольной графики и анимации, но управление отдельными объектами и их интерактивность реализуются сложнее, так как отсутствует встроенная модель DOM. Выбор технологии определяется требованиями к интерактивности, производительности и сложности представления.

Существующие решения:

- 1) <https://harleensachdev.github.io/interactiveschoolmap/> - открытый код, но функции не позволяют строить графы поверх схемы. Навигация лишь текстовыми подсказками.
- 2) <https://www.rustore.ru/catalog/app/com.shenid.mobileApp> - на момент написания данного текста проект недоступен, но есть приложение (которое не работает) в магазине rustore, проект интегрирует в планы помещений расписание уроков, но свои схемы создавать нельзя.
- 3) https://interactive_map.tilda.ws/ - сайт содержит просто планы помещений школы без интерактивных элементов.

- 4) <https://www.mappedin.com/industries/colleges-and-universities/> - мощный коммерческий инструмент для создания планов помещений с возможностью бесплатной работы, но недоступный в РФ.

Для моделирования внутренних пространств было выбрано образовательные учреждения с репрезентативной планировкой, которая включает последовательность кабинетных и коридорных структур. Школы обычно характеризуются сочетанием одно- и многоэтажных блоков, разнообразием размеров помещений и наличием зон общего пользования. Такая структурная разнообразность позволяет отработать алгоритмы маршрутизации и учесть факторы, влияющие на поток людей и распределение функциональных зон. Школа - подходящий объект для моделирования, поскольку здесь требуются разные режимы навигации и предоставления контекстной информации.

1.2. Формирование базы данных JSON для помещений и связей между ними

В разрабатываемом проекте модель здания хранится в одном файле `building.json` и используется как единый источник данных для визуализации и навигации. На верхнем уровне JSON содержит метаданные (`schemaVersion`, `title`), перечень ассетов (папка/пути к изображениям) и основные разделы: `floors` (этажи), `connectors` (межэтажные связи, например лестницы) и `evacuation` (настройки эвакуации).

Каждый объект `floors` описывает один этаж и включает: индекс/идентификатор, подпись для кнопки выбора этажа, путь к фоновой картинке `backgroundImage`, список кабинетов `rooms`, список узлов графа `nodes` и сам граф `graph` в виде списка смежности (для каждого `nodeId` - список соседей). Кабинеты `rooms` содержат `id`, `label`, `info`, `image`, набор CSS-классов и `style` (позиционирование `top/left` в `px` или `%`, чтобы совпадать с изображением плана). Узлы `nodes` хранят `id` и `style` (координаты), а также могут иметь подпись для

отладки и редактирования. Отдельно на этаже задаётся список `exitNodes` (узлы, ведущие к выходам) и стили элементов интерфейса (например `showAllPathsButtonStyle`).

Межэтажные переходы описываются в `connectors`: каждая лестница задаётся как группа узлов по этажам (`nodesByFloor`), что позволяет корректно связывать именно «свою» лестницу на каждом уровне (например LWF -> LWS -> LWT). Такой подход исключает ошибочные маршруты, когда алгоритм выбирает ближайшую, но топологически неверную лестницу. Для удобства адаптации под другие здания в проект встроен визуальный редактор: можно включить показ графа, перемещать узлы и кабинеты с привязкой к невидимой сетке, создавать рёбра и экспортировать обновлённый `building.json` без ручного правки кода.

2. РАЗРАБОТКА ВЕБ-ИНТЕРФЕЙСА ПЛАНА

2.1. Архитектура веб-приложения и выбор технологического стека (HTML, CSS, JavaScript, SVG)

Проект реализован как статическое клиентское веб-приложение и состоит из четырёх основных файлов: `index.html` (разметка), `styles.css` (стили), `script.js` (логика) и `building.json` (данные здания). При открытии страницы скрипт загружает `building.json` через `fetch`, формирует DOM-структуру этажей и запускает обработчики взаимодействия. Серверная часть не требуется: приложение рассчитано на локальный запуск и работу в офлайн-среде.

Отображение этажа строится по слоям: фоновое изображение плана (`backgroundImage`), слой интерактивных маркеров кабинетов (кнопки с абсолютным позиционированием), SVG-слой маршрута (линии пути и подсветка шагов) и дополнительный SVG-слой графа (для режима «Показать граф» и редакторов). Управляющие элементы интерфейса включают переключение этажей, поиск, установку точки «Я здесь», режим эвакуации и кнопку QR для переноса текущего состояния (URL) на другое устройство.

Состояние навигации хранится на клиенте и синхронизируется с адресной строкой. Параметр `roomId` задаёт текущую позицию пользователя, параметр `to` - целевое помещение, а `evacuate=true` включает эвакуационный режим. Это позволяет открывать страницу сразу с готовым маршрутом или сценарием эвакуации, а также делиться ссылкой через QR-код. Редактор данных (граф/кабинеты/этажи/лестницы) использует те же структуры JSON и экспортирует изменения обратно в файл `building.json`.

2.2. Визуализация плана помещений с использованием веб-технологий

Визуальная часть плана построена на фоновых растровых изображениях этажей. Такой подход упрощает перенос существующих бумажных/цифровых планов: достаточно подготовить изображение (PNG/JPG) и задать для объектов координаты в процентах или пикселях. Координатная система едина для всех объектов данного этажа: кабинеты и узлы графа позиционируются относительно контейнера изображения, что обеспечивает совпадение с планом без сложной векторизации.

Панорамирование и масштабирование реализованы на уровне контейнера этажа с помощью CSS-transform. Масштабирование не изменяет исходные координаты объектов: меняется только визуальное представление, поэтому расчёты маршрута выполняются в исходной системе координат, а линия пути корректно накладывается поверх плана. Обработаны сценарии мыши и сенсорных устройств, чтобы приложение оставалось удобным на телефонах и планшетах.

Маршрут визуализируется поверх плана в отдельном SVG-слое: для каждого этажа рисуется последовательность отрезков по узлам графа. При переходе между этажами маршрут разбивается на сегменты, и интерфейс подсвечивает актуальный этаж и текущий шаг. Дополнительно предусмотрен режим отображения графа (узлы и рёбра) для контроля топологии и отладки расположения точек.

В режиме редактирования SVG-слой графа становится интерактивным: узлы и кабинеты можно перемещать с привязкой к невидимой сетке, рёбра создавать в режиме соединения (перетягивание от узла к узлу с временной пунктирной линией) и удалять. Окна редакторов открываются только для активного режима и могут перемещаться по экрану за шапку, что облегчает точную настройку схемы.

2.3. Реализация функций поиска помещений и отображения информации

Поиск помещений реализован на стороне клиента: при загрузке `building.json` формируется индекс по полям `id` и `label` (а также по дополнительным метаданным, если они заданы). Пользователь вводит запрос, получает список совпадений и может сразу выбрать действие: перейти к описанию помещения, установить «Я здесь» или построить маршрут.

Карточка помещения отображается во всплывающем окне и содержит название, описание и при необходимости изображение (например фото кабинета). Из карточки доступны основные действия навигации: старт маршрута к выбранному кабинету и фиксация текущего местоположения. При выборе объекта карта автоматически переключается на нужный этаж и фокусируется на области, где расположен кабинет.

Для переноса текущего состояния между устройствами добавлена кнопка QR: она открывает всплывающее окно с QR-кодом текущего адреса страницы, включая все GET-параметры (`roomId`, `to`, `evacuate`). Генерация QR выполняется полностью офлайн средствами JavaScript, без обращения к внешним сервисам, поэтому функция работает даже при отсутствии интернета в здании.

Редактор кабинетов позволяет добавлять и изменять помещения прямо на схеме: задавать `id`, `label`, `info`, `image`, классы и координаты. При необходимости кабинет можно привязать к конкретному узлу графа (`nodeId`), чтобы навигация начиналась/заканчивалась в строго заданной точке (например, у двери

кабинета). После правок данные экспортируются в JSON, что делает перенос проекта на другие здания быстрым и воспроизводимым.

3. РЕАЛИЗАЦИЯ АЛГОРИТМА НАВИГАЦИИ

3.1. Представление плана здания в виде графа для алгоритмов поиска пути

Для построения маршрутов план здания представлен как взвешенный граф. Узлы графа (nodes) соответствуют ключевым точкам перемещения: пересечения коридоров, входам в блоки, лестничным площадкам и другим опорным точкам. Рёбра (graph) описывают доступные проходы между узлами на одном этаже. Каждый узел имеет координаты на плане, поэтому длина ребра может вычисляться как геометрическое расстояние между точками в пикселях исходного изображения.

Связи между этажами реализованы отдельным уровнем данных - connectors. Для каждой лестницы задаётся набор узлов по этажам (nodesByFloor), и при построении глобального графа добавляются рёбра между соседними этажами только внутри одной и той же лестницы. Если на каком-либо этаже у лестницы нет прохода к нужной зоне, алгоритм не должен «перепрыгивать» на другую лестницу. Начальная и конечная точки маршрута выбираются по кабинету: либо по явно заданному nodeId, либо по ближайшему узлу к координате кабинета.

3.2. Разработка алгоритма построения оптимального маршрута

Оптимальный маршрут строится алгоритмом Дейкстры по объединённому графу. Для уникальности вершин используется составной ключ «этаж:узел». Внутри этажа вес ребра равен евклидову расстоянию между центрами узлов. Межэтажные рёбра формируются из connectors и соединяют узлы лестницы на соседних этажах.

Переход по лестнице дополнительно «штрафуется» параметром stairPenaltyPx: к весу межэтажного ребра прибавляется фиксированное

значение. Алгоритм не будет «гулять» по этажам ради небольшого выигрыша в несколько пикселей расстояния на плане. Значение `stairPenaltyPx` можно задать в `building.json` в разделе `navigation`, подбирая его под масштаб конкретных изображений и архитектуру здания.

Результат поиска пути преобразуется в пошаговую инструкцию: маршрут группируется по этажам, интерфейс показывает последовательность переходов и автоматически переключает этаж при необходимости. При каждом построении маршрута адресная строка обновляется (`roomId` - откуда, `to` - куда), что позволяет сохранять и пересылать ссылку на маршрут. При изменении исходной позиции или цели маршрут пересчитывается, и визуализация обновляется без перезагрузки страницы.

3.3. Реализация построения пути для эвакуации

Эвакуационный режим использует тот же граф и алгоритм поиска пути, но целевой набор определяется настройками `building.json` (раздел `evacuation`). В проекте поддерживаются два варианта: явный список конечных выходов `finalExitRoomIds`. Это позволяет задавать несколько выходов и автоматически выбирать ближайший безопасный вариант.

После запуска эвакуации система строит маршрут до ближайшего выхода и отображает его на плане. Маршрут сохраняется в URL через параметр `evacuate=true`, чтобы сценарий можно было открыть повторно или отправить по ссылке/QR.

Для повышения наглядности предусмотрена кнопка «Показать все пути к выходам» на соответствующем этаже. В этом режиме строятся и рисуются дополнительные направления от текущего положения к узлам, отмеченным как `exitNodes`, что помогает быстро сориентироваться даже при ограниченной видимости схемы или при необходимости выбрать альтернативу.

3.4. Тестирование веб-приложения и алгоритма навигации

Проведены функциональные тесты веб-интерфейса с целью проверки корректности отображения плана на разных разрешениях и в разных браузерах. Проверялись: переключение этажей, работа панорамирования и масштабирования, отображение кабинетов и всплывающих карточек.

Проверка алгоритма маршрутизации выполнялась на типовых сценариях перемещения между кабинетами на разных этажах, включая случаи с несколькими лестницами. Также тестировались эвакуационные сценарии с несколькими выходами и отображение альтернативных направлений на этаже. Выявленные ошибки связности графа устранялись через редактор и повторное тестирование подтверждало корректность маршрутов.

В повседневном сценарии пользователь находит нужный кабинет через поиск или кликом на схеме, получает карточку с описанием и строит маршрут от текущей позиции. Приложение ведёт по шагам, автоматически переключая этаж и подсвечивая сегменты пути, что снижает время на ориентацию в незнакомом здании.

Сценарий «поделиться маршрутом» реализован через параметры URL и QR-код: после построения маршрута в адресной строке появляются параметры `roomId` и `to`, а QR-код содержит эту ссылку целиком. Пользователь может открыть маршрут на телефоне или переслать ссылку в мессенджере, получив тот же результат при загрузке страницы.

Для администрирования и переноса системы на другие объекты предусмотрены встроенные редакторы. Можно добавлять этажи, менять фоновые изображения, расставлять кабинеты и узлы с привязкой к сетке, настраивать лестницы (межэтажные связи) и экспортировать обновлённый `building.json`. Это уменьшает зависимость от разработчика и ускоряет обновление схем.

3.5. Анализ результатов тестирования и рекомендации по дальнейшему развитию

В ходе тестирования были выявлены основные факторы, влияющие на качество навигации: точность координат объектов на плане и корректность топологии графа, особенно в местах межэтажных переходов. При ошибках в данных (например отсутствующий узел или неверная лестничная связь) маршрут может приводить к тупику, поэтому важны проверки целостности и удобные средства редактирования.

Дальнейшее развитие проекта целесообразно направить на поддержку ограничений доступности (безбарьерные маршруты, лифты), временных перекрытий и расширение информационных карточек помещений. В качестве направления улучшения офлайн-использования можно оформить приложение как PWA (установка на устройство и кэширование ассетов), сохранив при этом текущую простую архитектуру без обязательной серверной части.

ЗАКЛЮЧЕНИЕ

Разработанная концепция интерактивного онлайн-плана реализована в виде веб-приложения, предназначенного для навигации по многоэтажному зданию. В основе решения лежит единая JSON-модель `building.json`, описывающая этажи, кабинеты, узлы графа и связи между ними, а также фоновые изображения планов. Это обеспечивает воспроизводимость данных и упрощает адаптацию проекта под другие объекты.

Приложение использует стандартные веб-технологии HTML, CSS и JavaScript и не требует серверной части. Маршруты и элементы графа отображаются поверх планов в SVG-слое, что даёт точное наложение линий и хорошую читаемость на разных разрешениях. Проверка на разных устройствах подтвердила работоспособность интерфейса и основных функций.

Навигационный модуль строит оптимальные маршруты по взвешенному графу с учётом межэтажных переходов. Реализованы два ключевых режима

применения: повседневная навигация между кабинетами и эвакуация к ближайшему выходу. Состояние маршрута синхронизируется с URL, что позволяет открывать и передавать готовые маршруты. Для удобства переноса на мобильные устройства добавлена офлайн-генерация QR-кода текущей ссылки.

Практическая ценность проекта повышена за счёт встроенных визуальных редакторов: можно редактировать граф, кабинеты, этажи и лестницы непосредственно на схеме, выравнивая элементы по сетке и сразу проверяя маршрутизацию. Экспорт JSON после правок делает обновление данных быстрым и независимым от ручного изменения кода.

Перспективы развития включают поддержку более сложных ограничений (лифты, доступность, временные перекрытия), автоматическую валидацию целостности данных при экспорте и оформление приложения как автономного офлайн-инструмента (PWA). Эти направления сохраняют заложенную в проекте модульность и расширяемость.

Первая версия и перспективные наброски конструктора планов здания расположены по ссылке: <https://github.com/Luintopagent/site>, рабочий пример располагается по ссылке <https://ДМИП.рф/fplan/>

СПИСОК ЛИТЕРАТУРЫ

1. Аникеева О.С. Публикация карт в сети интернет: эволюция картографии // Наука. Инновации. Технологии. — 2015. — №2. — С. 79–82.
2. Барадкулькина А.С., Федюкович Т.В., Ефремов А.А. Тестирование программного обеспечения в процессе разработки прикладных программ // 60-я юбилейная научная конференция аспирантов, магистрантов и студентов бгуир. — Минск, 2024. — С. 310–311.
3. Вагин Д.В., Петров Р.В. Современные технологии разработки веб-приложений. — Новосибирск: Изд-во НГТУ, 2019. — 52 с.

4. Сатинов А.К., Смагулова А.С., Белозубов А.В. Аспекты разработки современного web-приложения // Университет еңбектері – труды университета. — 2021. — №4. — С. 317–321.
5. Сысолетин Е.Г., Ростунцев С.Д. Проектирование интернет-приложений. — Екатеринбург: Издательство Уральского университета, 2015. — 92 с.
6. Фофанов О.Б. Алгоритмы и структуры данных. — Томск: Издательство Томского политехнического университета, 2014. — 126 с.

Приложение 1. Скриншоты проекта

Этаж 1
Этаж 2
Этаж 3

Тех. центр
Л
В
К
С
11
10
9
Ж
М
Л
Библ
Мед
СП
7
Л
К
Ф
5
6
Г

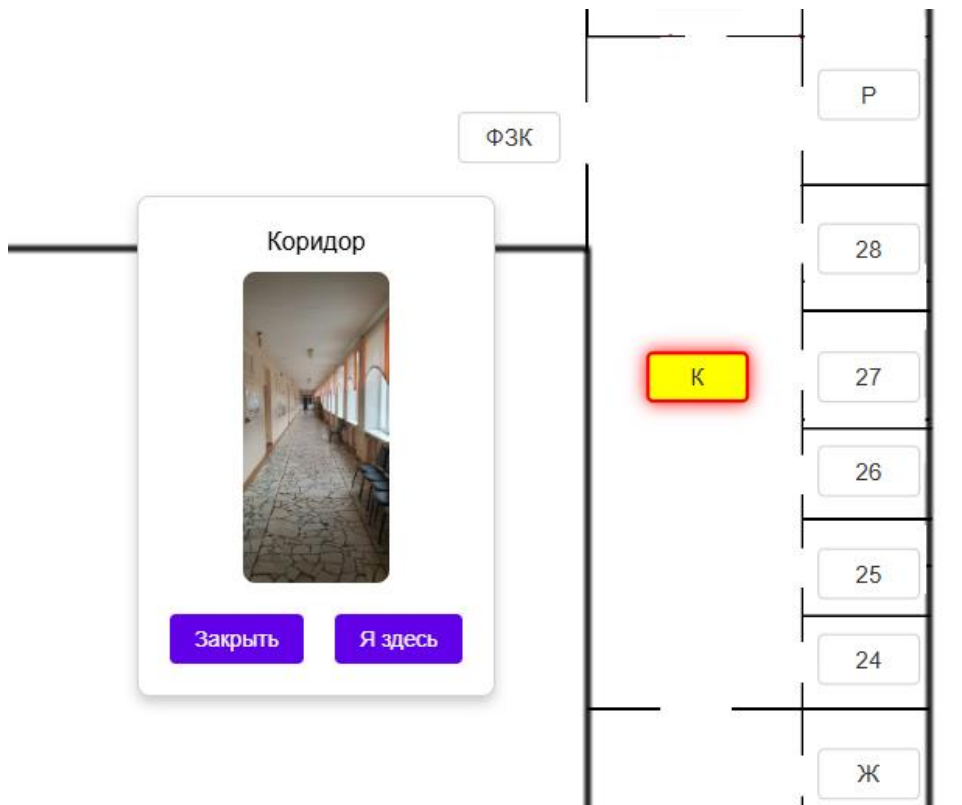
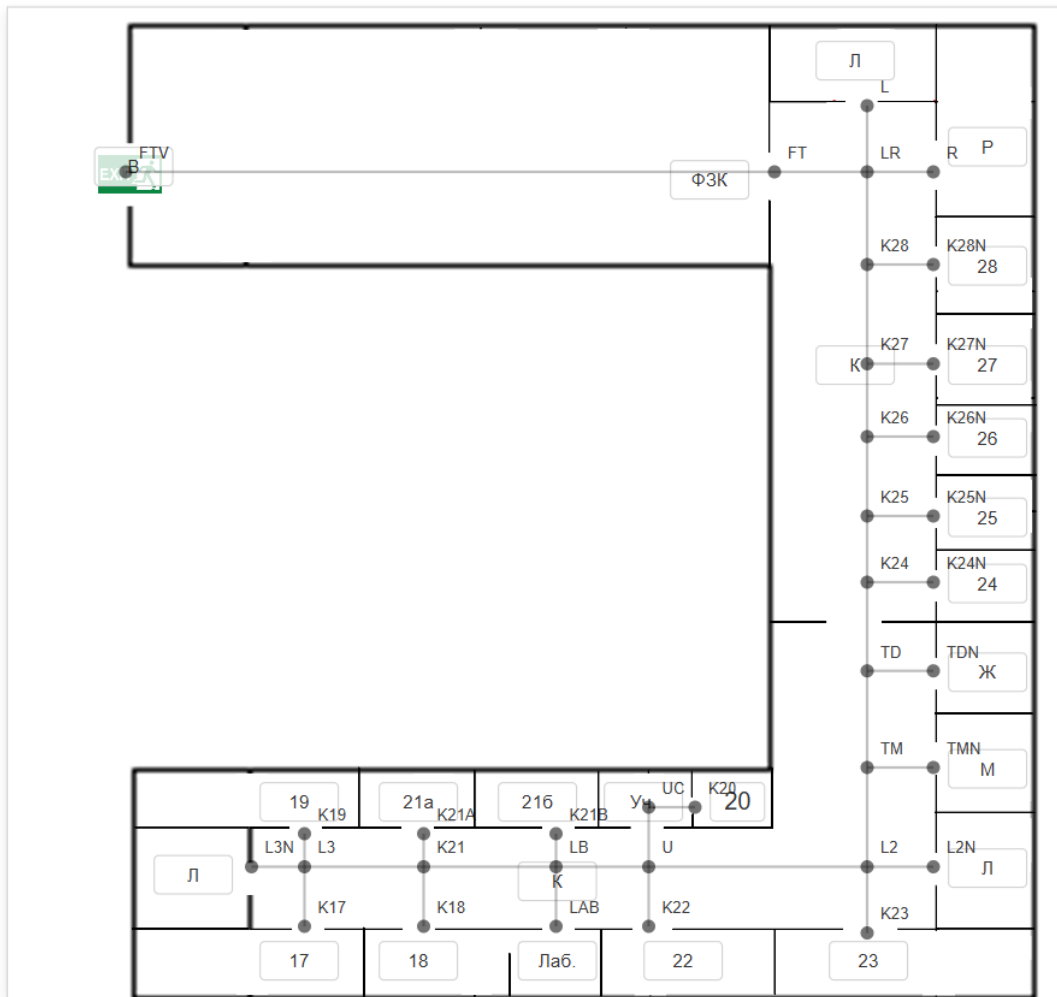
Показать граф редактор графа редактор кабинетов редактор этажа редактор связи лестниц

Этаж 1. Дойдите до лестницы, затем поднимитесь на Этаж 2.
Завершить Далее

К 42
41
40
39
СП
СП
Л
38
А
36

Этаж 3. Следуйте по маршруту до: Кабинет русского языка.
Завершить

The image displays a software interface for a floor plan navigation application. At the top left, there are three vertical buttons labeled 'Этаж 1' (Floor 1), 'Этаж 2' (Floor 2), and 'Этаж 3' (Floor 3). The main area shows a detailed floor plan of the first floor with various rooms labeled: 'Тех. центр' (Tech center), 'Л' (Lobby), 'В' (Office), 'К' (Office), 'С' (Office), '11', '10', '9', 'Ж' (Office), 'М' (Office), 'Л' (Lobby), 'Библ' (Library), 'Мед' (Medical), 'СП' (Storage), '7', 'Л' (Lobby), 'К' (Office), 'Ф' (Office), '5', '6', and 'Г' (Office). A green arrow indicates a path starting from room 'Ф' and moving towards room 'Л'. Below the floor plan, there is a control bar with several checkboxes: 'Показать граф', 'редактор графа', 'редактор кабинетов', 'редактор этажа', and 'редактор связи лестниц', followed by a purple button 'Экспорт JSON'. A pop-up dialog box is overlaid on the floor plan, containing the text 'Этаж 1. Дойдите до лестницы, затем поднимитесь на Этаж 2.' and two buttons: 'Завершить' and 'Далее'. To the right, a vertical strip of the floor plan shows rooms 'К', '42', '41', '40', '39', 'СП', 'СП', 'Л', '38', 'А', and '36', with a green arrow indicating a path upwards. Another pop-up dialog box is overlaid on this strip, containing the text 'Этаж 3. Следуйте по маршруту до: Кабинет русского языка.' and a purple button 'Завершить'. The interface also features several floating icons: a globe, a QR code, a flag, and a blue circle with a white 'K'.



Рецензия научного руководителя на проектную работу

Представленная работа посвящена разработке прототипа одностраничного веб-приложения для интерактивной навигации по многоэтажному зданию (на примере школы): визуализация планов этажей, поиск помещений, построение оптимального маршрута и отдельный эвакуационный режим. Тематика является актуальной, поскольку задачи навигации в сложных общественных зданиях напрямую связаны с удобством пользователей и вопросами безопасности, а использование веб-технологий обеспечивает кроссплатформенность и простоту внедрения.

Авторами выполнен полный цикл проектирования: проведён обзор существующих решений, предложена структура данных здания в формате JSON, описана архитектура приложения (HTML/CSS/JavaScript + SVG-слой), реализована маршрутизация по модели графа с применением алгоритма Дейкстры и введением «штрафа» за межэтажные переходы, что повышает реалистичность маршрутов. Отдельно следует отметить практическую ценность встроенных визуальных редакторов (кабинеты/узлы/рёбра/лестницы) с последующим экспортом данных в JSON — это существенно упрощает адаптацию решения под другие объекты без ручной правки кода.

К сильным сторонам работы отношу:

- логичную структуру и последовательность изложения (от анализа к проектированию, реализации и тестированию);
- обоснованный выбор стека и технологических решений (SVG-слой для маршрутов и графа поверх растрового плана);
- корректное применение графов;
- ориентацию на реальное использование: синхронизация состояния с URL и генерация QR-кода для переноса маршрута на другое устройство;

В целом работа производит положительное впечатление: проект доведён до работоспособного прототипа, решения обоснованы, функциональность соответствует заявленным целям, а результаты обладают практической значимостью и потенциалом масштабирования.

Заключение: работа соответствует требованиям к проектным/исследовательским работам данного уровня, может быть рекомендована к защите

Научный руководитель: Адамский С. С.

Дата: «12» января 2026 г.

