

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(Н И У « Б е л Г У »)**

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

РАЗРАБОТКА ИНТРАКТИВНОЙ КАРТЫ КОРПУСОВ НИУ БЕЛГУ

Выпускная квалификационная работа студента
обучающегося по направлению подготовки
09.03.02 Информационные системы и технологии
очной формы обучения, группы 07001409
Дворянина Дмитрий Михайлович

Научный руководитель
ст. пр. Удовенко И.В.

БЕЛГОРОД 2018

РЕФЕРАТ

Разработка интерактивной карты корпусов НИУ «БелГУ». – Дворянин Дмитрий Михайлович, выпускная квалификационная работа бакалавра. Белгород, ФГАОУВО Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 45, включая приложения 63, количество рисунков 19, количество использованных источников 30.

КЛЮЧЕВЫЕ СЛОВА: интерактивная карта, геоинформационная система, веб-сайт, браузер, маршрутизация, геолокация, корпуса, позиционирование.

ОБЪЕКТ ИССЛЕДОВАНИЯ: корпуса НИУ «БелГУ».

ПРЕДМЕТ ИССЛЕДОВАНИЯ: методы разработки интерактивных карт.

ЦЕЛЬ РАБОТЫ: усовершенствование процесса навигации пользователей посредством разработки интерактивной карты НИУ «БелГУ».

ЗАДАЧИ ИССЛЕДОВАНИЯ: исследование предметной области, разработка интерактивной карты, отображающей расположение корпусов НИУ «БелГУ», создание описания подразделений НИУ «БелГУ», внедрение метода построения маршрута от текущего местоположения пользователя, до выбранного объекта.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: в результате работы была разработана интерактивная карта корпусов НИУ «БелГУ».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитическая часть.....	6
1.1 Исследование предметной области.....	6
1.2 Анализ существующих методов реализации	7
1.3 Анализ бизнес-процесса интерактивной карты НИУ «БелГУ» «Как есть»..	10
1.4 Построение модели бизнес-процесса интерактивной карты НИУ «БелГУ» «Как должно быть»	13
2 Обоснование проектных решений.....	16
2.2 Обоснование проектных решений по информационному обеспечению	18
2.3 Обоснование проектных решений по программному обеспечению	20
2.4 Обоснование выбора программных средств	24
2.5 Яндекс карты API.....	25
3 Разработка интерактивной карты	27
3.1 Описание информационной модели.....	27
3.2 Проектирование интерактивной карты.....	28
3.3 Разработка карты	31
3.3 Используемые методы реализации	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	44
ПРИЛОЖЕНИЕ А	47

ВВЕДЕНИЕ

В настоящее время сеть Интернет занимает лидирующую позицию в сфере средств массовой информации. По статистике, число пользователей, активно использующих интернет ресурсы для поиска различной информации с каждым годом возрастает.

Создание информационного ресурса в интернете обеспечивает новые возможности по расширению работы предприятия, повышение престижа учебных заведений или доступности информации, предоставляемой социальными и муниципальными службами.

Информационный ресурс должен предоставлять данные которые можно многократно использовать для решения проблем пользователей [1]. Грамотно построенный информационный ресурс способен выполнить множество различных функций. Таких как предоставление информации о деятельности организации, ее структурной организации, информационное содержание структурных подразделений. Так же необходимо обеспечивать простоту его нахождения исходя из запросов поисковых систем, так как целевая аудитория ресурса строится на пользователях, которые ищут конкретную информацию в Интернете. Существует большое количество информационно-справочных ресурсов, которые могут предоставить информацию, удовлетворяющую потребности пользователя, что в свою очередь упрощает и ускоряет процесс получения знаний.

Постоянное взаимодействие с пользователями ресурса позволяет в кратчайшие сроки реагировать на изменения направленности интересов и производить своевременную коррекцию. В дополнении к этому реклама в сети Интернет является менее финансово затратной нежели использование устоявшихся средств.

В современном мире информационный ресурс представляет собой не только средство получения информации, а способно позиционировать себя как

полноценный маркетинговый инструмент, с помощью которого возможно привлечение новых пользователей.

Таким образом, целью дипломной работы является разработка интерактивной карты корпусов НИУ «БелГУ».

Целевая аудитория: студенты очных и заочных отделений университета, абитуриенты, преподаватели, различного рода делегации, а также потенциальные работодатели.

Объектом исследования являются корпуса НИУ «БелГУ».

Предметом исследования являются методы разработки интерактивных карт.

Цель работы: усовершенствование процесса навигации пользователей посредством разработки интерактивной карты НИУ «БелГУ»

Задачи:

- исследование предметной области;
- разработка интерактивной карты, отображающей расположение корпусов НИУ «БелГУ»;
- создание описания подразделений НИУ «БелГУ»;
- внедрение метода построения маршрута от текущего местоположения пользователя, до выбранного объекта.

Пояснительная записка выполнена на 44 страницах без приложения, и 61 с приложением, содержит 19 рисунков, 1 таблицу и приложение.

1 Аналитическая часть

1.1 Исследование предметной области

Карта всегда являлась самым практичным средством для восприятия отображения пространства вокруг человека. До настоящего времени основное внимание уделялось географическим картам, отображающим трехмерное пространство в различных двумерных картографических проекциях на плоскости. Данный метод затруднял построение оптимальных маршрутов для использования, был трудоемким и неудобным, так как для построения нового маршрута необходимо было заново просчитывать маршрут следования и наносить его на карту. В период конца 20го начала 21 века был совершен рывок в области электронных вычислительных машин. Который позволил использовать новейшие средства для хранения, сбора и обработки информации. Благодаря этому стал возможен переход от бумажных карт к более качественным и детальным цифровым. С течением времени и совершенствованием технологий стали появляться электронные карты, которыми пользуются в настоящее время. Используя такие карты и современные технологии, пользователь с легкостью может ориентироваться в окружающем его пространстве. Так же данные технологии позволяют прокладывать интересующие его маршруты с минимальной затратой времени.

В настоящий момент популярность завоевали геоинформационные системы, предназначенные для сбора, хранения, анализа и географической визуализации пространственных данных [2]. Благодаря данным системам у пользователей появляется возможность обмениваться информацией под средствами интерактивных карт. Карты сами по себе являются универсальным средством передачи информации что позволяет вносить ясность в формы окружающего мира, а также они позволяют получать новые данные для обнаружения новых объектов и их интерпретации. Интерактивная карта – это карта, которой можно свободно управлять, то есть изменять место,

отображаемое на карте, фильтровать информацию, которая будет отображаться в нужный момент пользователю, поиск информации на карте используя форму поиска и фильтрации данных. Различие между картами характеризуется в визуальной составляющей и функциональными возможностями, в зависимости от задач для которых будет использована карта.

1.2 Анализ существующих методов реализации

В эру цифровых технологий подавляющее большинство медиа-гигантов, стараются развиваться в различных отраслях, затрагивающих жизнь современного человека. Большинство компаний занимаются разработкой различного программного обеспечения, информационных ресурсов, медиа порталов.

Благодаря стремительному развитию цифровых технологий некоторые корпорации организывают разработку программного интерфейса приложения, который представляет собой набор готовых классов, процедур, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах [3]. Ввиду этого у пользователей появляется возможность использовать некоторые компоненты коммерческих проектов в своих целях.

Наиболее популярными методами разработки интерактивных карт является создание небольших проектов на основании существующих больших используя программный интерфейс приложения или другие более упрощенные методы разработки.

Часто используемыми методами является создание собственных карт с использованием конструкторов предоставляемых компаниями как Google, Yandex, 2GIS, Bing и другие.

Конструкторы собственных веб карт могут быть представлены как онлайн решение или программное обеспечение. Благодаря использованию вышеприведенных способов упрощается разработка интерактивных карт.

Основными преимуществами использования таких методов является:

- простота создания интерактивных карт;
- возможность использования без навыков программирования;
- простота интегрирования в информационный ресурс

Преимущества использования онлайн конструктора на примере Яндекс карт, отображены на рисунке 1.1.

	Конструктор карт	Виджет Яндекс.Карт	Геоссылка
Показать объект на карте	✓	✓	✓
Показать маршрут на карте	—	✓	—
Добавить собственные объекты	✓	—	—
Отметить зоны доставки	✓	—	—
Поделиться ссылкой на карту	✓	—	—
Напечатать карту	✓	—	—
Технология добавления на сайт	script или iframe	iframe	script

Рисунок 1.1 – Преимущества использования конструктора

Недостатками являются:

- отсутствие более гибкой настройки карт;
- отсутствие возможности персонализировать карту под выполнение необходимых задач.

Более предпочтительным методом создания интерактивных карт, является разработка собственной карты с использованием API [3]. Используя этот метод разработчику предоставляется весь необходимый инструментарий для более детальной настройки.

Любая современная технология предполагает возможность работы с API. Для выбора используемой системы были рассмотрены несколько сервисов позволяющих работать с ГИС [2], особое внимание уделялось возможности построению маршрутов.

Google Maps. Функционал, предоставляемый компанией, позволяет находить маршруты во всем мире. Построение маршрута может быть осуществлен с использованием контекстного меню, определив географический объект как начальную точку движения или конечный пункт назначения, или используя поисковое поле в виде «Пункт отправления, Пункт назначения». Построенный маршрут будет отображен на карте в виде ломаной линии, которая может быть дополнена различной информацией, такой как приблизительное время прохождения маршрута, его расстояние и так далее.

Предоставляемый сервис позволяет строить оптимальный маршрут используя несколько точек, но в большинстве случаев сервис не способен корректно отобразить мелкие объекты или маршруты к ним. Данные, предоставляемые транспортными сервисами, имеют различную степень покрытия в разных странах, что может затруднить построение маршрута [4].

Яндекс Карты. В данном сервисе лучше всего представлена информация о городах России, Украины, Белоруссии, Казахстана. Набор инструментов для работы с картами является обширным, но более полный функционал доступен только для ограниченного числа городов. Яндекс позволяет строить наименее затратный маршрут между несколькими объектами с возможными промежуточными метками. Так же в функционал включена возможность построения маршрутов с учетом пробок на дороге. Представленное решение позволяет использовать разные варианты построения маршрута с использованием транспорта следующего по городским маршрутам, но эта функция доступна только для крупных городов России [5].

2Gis. Также имеет покрытие большинства стран СНГ. Предоставляет возможность построения маршрута с учетом личного и общественного транспорта, либо пешком. Представленная система вооружена большой степенью детализации для пригородных районов. Единственным значимым недостатком является возможность построения маршрутов только между двумя точками [6].

В ходе решения задачи было принято решение использовать сервис, предоставляющий API Яндекс Карт, так как данная система наиболее часто используется при работе с online картами в России и имеет большую площадь покрытия ее территории. К тому же Яндекс Карты наиболее корректно выполняют функцию построения маршрутов по городам России.

1.3 Анализ бизнес-процесса интерактивной карты НИУ «БелГУ» «Как есть»

Начальным этапом разработки интерактивной карты является анализ рассматриваемого бизнес-процесса. Этот этап позволит сформировать более наглядную структуру решаемой проблемы. Наиболее наглядным отображением протекания процесса является использование функциональной модели, отображающей структуру и функции системы IDEF0[].

Благодаря данной методологии предоставляется возможность создать лаконичное описание моделируемых процессов, что в свою очередь повышает понимание проблем, возникающих в момент протекания процесса.

Текущая схема протекания бизнес-процесса представлена на рисунке 1.2



Рисунок 1.2 – Контекстная диаграмма бизнес-процесса поиска информации о корпусах НИУ «БелГУ»

Данная контекстная диаграмма (рисунок 1.2) представляет собой верхний уровень протекающего бизнес-процесса. Входящими данными в который являются информация, полученная в ходе посещения социальных сетей и электронных ресурсов интрасети сетей. Управляющими элементами бизнес-процесса являются нормативные положения, которые представляют собой правила пользования информационными ресурсами и публикации информации в корпоративных сетях. Механизмами влияния на протекание процесса является пользователь, который в свою очередь использует процесс для извлечения некоторых знаний. Выходными данными бизнес-процесса является информация, полученная пользователем в процессе взаимодействия с процессом, в данном случае выходными данными будет являться информация о корпусах и их расположении. Контекстную диаграмму необходимо декомпозировать для более подробного представления проблемных областей процесса. Декомпозированный бизнес-процесс поиска информации о корпусах НИУ «БелГУ» представлен на рисунке 1.3.

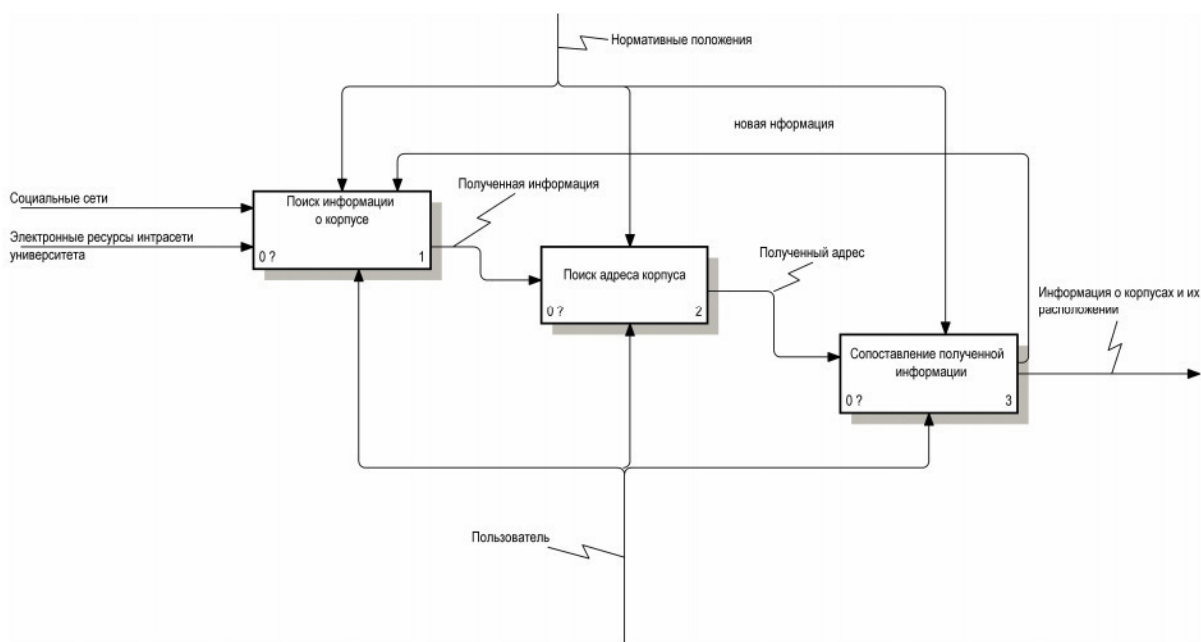


Рисунок 1.3 – Декомпозиция контекстной диаграммы поиска информации о корпусах НИУ «БелГУ»

Декомпозиция бизнес-процесса представляет собой три функциональных блока, которые отображают выполняемые действия. В данной декомпозиции выполняются следующие шаги:

- Поиск информации о корпусе происходит это в ходе посещения пользователем социальных сетей, а именно группы университета, которые могут содержать некоторую информацию о корпусах, так же поиск информации может быть осуществлен на электронных ресурсах интрасети в разделах с контактными данными.

Опираясь на информацию, извлеченную в ходе протекания бизнес-процесса, можно составить мнемосхему текущего процесса сбора информации о корпусах, которая изображена на рисунке 1.4, отображающая так называемую проблемную область.

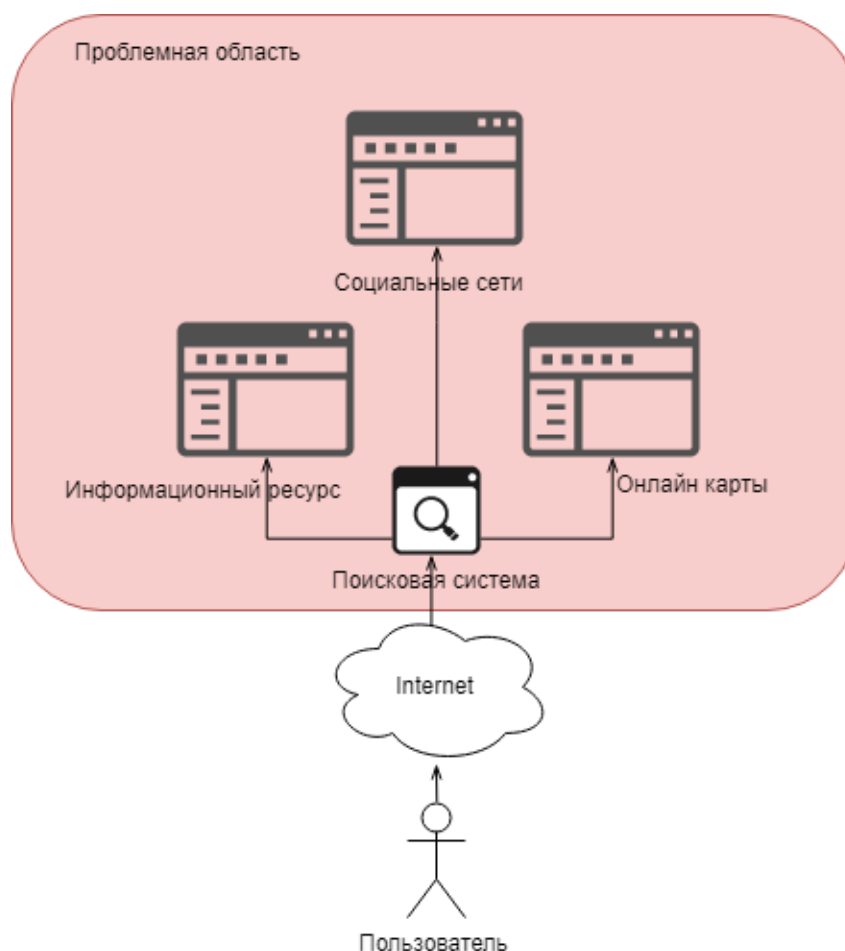


Рисунок 1.4 – Мнемосхема текущего процесса сбора информации о корпусах

На представленном рисунке отображена проблемная область, существующая в системе. Так как она содержит промежуточные узлы доставки информации от системы до конечного пользователя.

Данная модель повышает количество совершаемых действий пользователем, для достижения интересующего его результата. А именно поиск интересующего корпуса НИУ «БелГУ». Исходя из существующей схемы, пользователю необходимо воспользоваться несколькими информационными ресурсами для извлечения информации, такими как поисковая система для поиска адреса определенного корпуса. Либо использование социальных сетей и информационных ресурсов интрасети. А также существующими платформами для поиска расположения объекта на карте.

Таким образом, необходимо предоставить возможность пользователю получать необходимую информацию за наименьшее количество шагов. Предлагается сокращение излишних действий под видом создания карты учебных корпусов. Благодаря этому ресурсу сократится затрачиваемое пользователем время.

1.4 Построение модели бизнес-процесса интерактивной карты НИУ «БелГУ» «Как должно быть»

В ходе проведенного анализа существующих способов получения информации о расположении корпусов и структурных подразделений и получение информации о институтах и факультетах, расположенных в зданиях университета, необходимо выполнить построение диаграммы, которая будет наглядно отображать вносимые коррективы, основанные на мнемосхеме, изображенной на рисунке 1.4.

Контекстная диаграмма бизнес-процесса интерактивной карты НИУ «БелГУ» «Как должно быть» представлена на рисунке 1.5.



Рисунок 1.5 – Контекстная диаграмма разработки интерактивной карты НИУ «БелГУ»

Ключевым моментом оптимизации выбран подход сокращения действий пользователя для получения интересующей информации о корпусах университета. На представленной диаграмме (рисунок 1.5) можно проследить изменение перечня данных поступающих на вход рабочей функции.

Среди этих изменений можно заметить устранение проблемной области в виде данных получаемых пользователем используя более чем один информационный ресурс на котором он может ознакомиться с информацией о структурных подразделениях университета.

Для более полного представления внесенных изменений достаточно проанализировать диаграмму являющейся декомпозицией верхнего уровня иерархии проходящих процессов, представленной на рисунке 1.6.

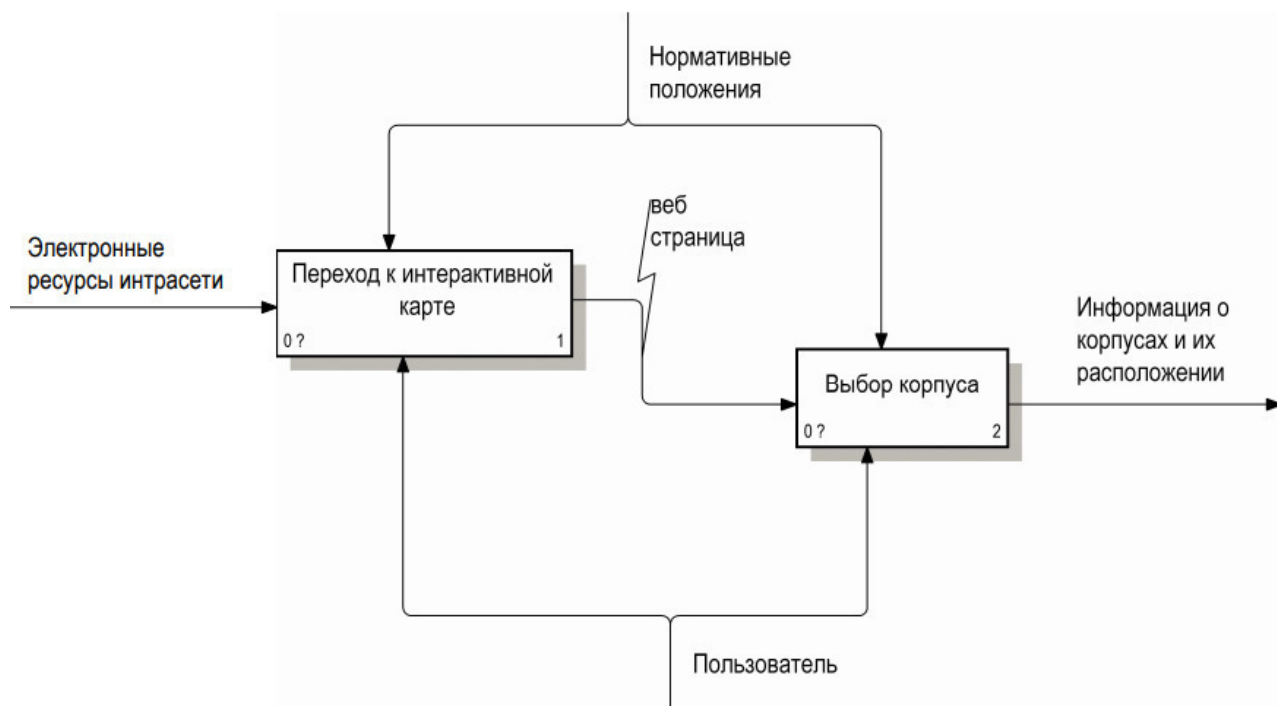


Рисунок 1.6 – Декомпозиция контекстной диаграммы

На рассматриваемой диаграмме наблюдается оптимизация в виде уменьшения количества выполняемых действий пользователем, что в последствии уменьшает затрачиваемое время на получение интересующей информации, связанной со структурными подразделениями университета.

Выводы по первому разделу

В первом разделе была исследована предметная область выбранной темы в соответствии с этим определены возможные и существующие методы реализации поставленной задачи. Был проведен анализ бизнес-процесса на основании уже существующей системы «Как есть». На основании данного анализа произвелось моделирование нового более совершенного процесса. На основании которого производилось модерирование схемы «Как должно быть».

2 Обоснование проектных решений

2.1 Обоснование проектных решений по техническому обеспечению (ТО)

Техническое обеспечение представляет собой комплекс технических средств, обеспечивающих работу информационных технологий [8].

Техническое обеспечение принято классифицировать по исполняемой роли при обработке информации:

- вычислительные машины либо компьютеры, которые занимают первостепенное место среди всех технических средств, позволяющих обрабатывать информацию;
- периферийные средства, занимающиеся вводом и выводом информации;
- сетевые коммуникации, для обеспечения передачи данных;
- оргтехника и иные средства связи.

В настоящий момент университет располагает всевозможными средствами, представленными в перечне выше.

На базе университета по состоянию на 2018 год содержится порядка 4000 компьютеров, расположенных в более чем 70 компьютерных классах [9]. Также на базе университета размещены два суперкомпьютера «Нежеголь» и «Кудесник».

Суперкомпьютер «Нежеголь» является суперкомпьютером с гибридной архитектурой. Комплекс включает в себя 20 вычислительных узлов. Для решения инженерных задач всевозможного характера и отраслевой специализации «Нежеголь» оснащен мощными программными средствами. Среди которых ANSYS и Matlab. Что позволяет эффективно разрешать задачи моделирования химических, физических и биологических процессов, а также позволяет решать задачи обработки сигналов.

В настоящее время суперкомпьютер способствует решению научных проектов университета.

В связи с различной направленностью работы некоторых компаний, возникает необходимость использования суперкомпьютера. Но таким организациям экономически не эффективно вкладывать средства в развертывание собственных вычислительных центров. Подходящим решением для таких задач является аренда уже имеющихся вычислительных мощностей. На сегодняшний день вычислительные мощности суперкомпьютера «Нежеголь» используют Федерально-региональный центр аэрокосмического и наземного мониторинга объектов и природных ресурсов города Белгорода и Научно-образовательный и инновационный центр «Наноструктурные материалы и нанотехнологии».

Технические характеристики суперкомпьютера:

- процессор – 20 Intel Sandy Bridge 2.4ГГц(x2);
- оперативная память 32ГБ;
- видеокарта – Tesla M2090
- сеть – QDR QSFP ДО 10ГБАЙТ/СЕК.
- высокопроизводительное хранилище данных – 24 Тбайт.

Так же на базе университета находятся сервера на которых уже успешно функционируют электронные информационные ресурсы интрасети. В дальнейшем на эти же сервера планируется установка интерактивной карты.

Для разработки интерактивной карты корпусов НИУ «БелГУ» был использован личный ноутбук, технические характеристики которого представлены в таблице 2.1.

Таблица 1.1 – Технические характеристики ноутбука

Наименование	Характеристики
Операционная система	Microsoft Windows 10 Pro x64
Оперативная память	Samsung SODIM DDR3 8 Гб
Процессор	Intel(R) Core(TM) i3-3110M CPU @2.40GHz
Видеокарта	NVIDIA Geforce 710M

Продолжение таблицы 1.1

Жесткий диск	HDD, 500 Гб
Оптический привод	DVD-RW
Мышь оптическая	Genius
Сетевой адаптер	Qualcomm Atheros

Для корректной работы интерактивной карты на компьютере пользователя должен быть установлен любой современный интернет браузер.

В настоящий момент популярно использование мобильных устройств, имеющих различный размер экрана, в ходе чего необходимо предусмотреть корректную работу интерактивной карты на разных размерах экранов пользователей. Разработанная интерактивная карта обладает гибким дизайном и корректно отображается на различных устройствах, под управлением всех известных операционных систем.

2.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение представляет собой структурированное создание информационных условий для функционирования системы, предоставление необходимой информации, обеспечение средствами поиска в системе, хранения, сбора, передачи, структурирования информации. Информационное обеспечение включает совокупность единой системы показателей, вариантов организации документооборота и различные информационные файлы, которые хранятся на машине и машинных носителях и имеющие различную степень организации [10]. На рисунке 2.1 отображена структура информационного обеспечения



Рисунок 2.1 – Структура информационного обеспечения

Информационное обеспечение информационных систем подразделяют на:

- внеашиинное;
- внутриашиинное.

Внеашиинное обеспечение включает в себя систему классификации и кодирования информации, системы документации, схемы информационных потоков документооборота и нормативно-справочные документы.

Внутриашиинным обеспечение является объединением данных, оцифрованных на машинных носителях, структурированных определенными признаками. Внутриашиинное обеспечение создает информационную среду системы.

Внеашиинное обеспечение относятся такие информационные документы:

- технические инструкции;
- ГОСТы;
- всевозможные договора;
- указания по работе с информационной системой.

Во время разработки входящих и исходящих документов разрешается использовать принципы оригинального проектирования, так как

преобладающее количество документов, принимающих участие в документообороте задачи, являются только внутренними и не выходят за пределы организации.

При построении функциональных частей интерактивной части необходимо руководствоваться общепринятыми положениями эргономичности и организации дружественного интерфейса интерактивной карты в отношении к пользователю. Информационная зона интерактивной карты должна отображаться вся информация, которую необходимо предоставить пользователю. В функциональной зоне необходимо отображать все возможные функции, которые будут доступны конечному пользователю. [11].

2.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение – наряду с аппаратными средствами, является важной составляющей информационных технологий, включает в себя программное обеспечение и данные, которые предназначены для решения узконаправленного круга задач и хранящиеся на жестких носителях. Программным обеспечением является набор данных которые используются в других программах или алгоритм, написанный в виде последовательных инструкций для выполнения некоторого процесса.

Программное обеспечение разделяется по на классы по способу распространения:

- свободное программное обеспечение (Freeware) – это программное обеспечение, которое может совершенно бесплатно устанавливаться и работать на любых компьютерах, без приобретения лицензии у разработчиков [12];

- открытое программное обеспечение (OpenSource) – это программное обеспечение, которое поставляется в составе открытого программного кода. Особенность такой поставки заключается в возможности

добавления некоторого функционала программному обеспечению пользователем в случае нехватки такового [13];

- закрытое программное обеспечение (Proprietary) – это программное обеспечение, исходный код которого полностью закрыт от вмешательства пользователя. Такое программное обеспечение защищено авторскими правами и также имеет ограничение на распространение и использование [14];

- коммерческое программное обеспечение это ПО которое распространяется под средством приобретения лицензии. Исходный код такого программного обеспечения является закрытым [15].

В разрабатываемой интерактивной карте нет отличий при работе на различных операционных системах. Интерактивная карта будет отображаться и функционировать на всех операционных системах, например, Windows, Linux, MacOS, ios или android.

Так же интерактивная карта разрабатывалась с учетом возможности работы в популярных интернет браузерах, таких как:

- Opera;
- Google Chrome;
- Firefox;
- Safari;
- Internet Explorer 8+;
- Microsoft Edge;
- Yandex-браузер.

Браузер – это специализированное программное обеспечение, которое позволяет пользователю находить, получать доступ и отображать веб-страницы. При обычном использовании веб-браузер обычно сокращается до «браузера». Браузеры используются в основном для отображения и доступа к веб-сайтам в Интернете, а также другой контент, созданный с использованием таких языков, как язык гипертекстовой разметки (HTML) и расширяемый язык разметки (XML).

Браузеры транслируют веб-страницы и веб-сайты, поставляемые с использованием протокола передачи гипертекста (HTTP), в человеко-читаемый контент. Они также имеют возможность отображать другие протоколы и префиксы, такие как защищенный HTTP (HTTPS).

Для корректного отображения веб ресурсов, на устройстве пользователя должен быть установлен Flash-player поставляемый компанией Adobe.

Flash Player – это программное обеспечение, которое позволяет просматривать и передавать мультимедийные данные в веб-браузерах.

Flash Player используется для воспроизведения мультимедийных файлов в Shockwave Flash Format (SWF). Эти файлы создаются с использованием основного Adobe Flash и других поддерживаемых программных средств.

В разработке интерактивной карты применялись языки программирования JavaScript, библиотека расширения языка JQuery, язык гипертекстовой разметки HTML пятой версии и таблица каскадных стилей CSS третьей версии. Так же был использован открытый стандарт обмена данными на основе синтаксиса JavaScript JSON.

JavaScript – это прототипированно-ориентированный сценарный язык программирования. JavaScript в основном используется как встраиваемый язык для обеспечения взаимодействия страницы с пользователем. На многих сегодняшних страницах есть меню, формы, слайд-шоу и даже изображения. JavaScript позволяет придать интерактивности статичным страницам любого ресурса.

JQuery - это краткая и быстрая библиотека JavaScript, которая может использоваться для упрощения обработки событий, перемещения HTML-документов, взаимодействия Ajax и анимации для быстрой разработки сайтов. JQuery упрощает HTML-клиентские скрипты, что упрощает разработку приложений Web.

HTML – это основной язык разметки гипертекста, используемый для отображения веб-страниц в Интернете. Другими словами, веб-страницы состоят из HTML, который используется для отображения текста, изображений или

других ресурсов через веб-браузер. Новые веб-разработчики могут ошибочно принимать HTML-код для языка программирования, когда он на самом деле является языком разметки. HTML используется с другими технологиями, потому что все HTML действительно делает документы. На стороне клиента JavaScript (JS) используется для обеспечения интерактивности. На стороне сервера используется платформа веб-разработки, такая как Ruby, PHP или ASP.NET.

Когда веб-разработчик создает приложение, работа выполняется на сервере, а необработанный HTML отправляется пользователю. Линия между разработкой на стороне сервера и развитием клиентской стороны размыта с такими технологиями, как AJAX. HTML никогда не был разработан для Интернета, который существует сегодня, поскольку это всего лишь язык разметки с серьезными ограничениями, с точки зрения контроля и проектирования. Для решения этой проблемы использовались многочисленные технологии – наиболее важным из которых является каскадная таблица стилей (CSS).

Каскадные таблицы стилей (CSS) - это стандарт (или язык), который описывает форматирование страниц разметки. CSS определяет форматирование для следующих типов документов:

- язык разметки гипертекста (HTML)
- расширяемый язык разметки гипертекста (XHTML)
- расширяемый язык разметки (XML)
- масштабируемая векторная графика (SVG)
- язык интерфейса пользователя XML (XUL)

CSS позволяет разработчикам разделять контент и визуальные элементы для большего контроля и гибкости страницы. Файл CSS обычно прикрепляется к файлу HTML с помощью ссылки в файле HTML.

Для редактирования и написания файлов, которые в будущем составят функциональные файлы интерактивной карты с расширениями js, html, css, json – использовался редактор исходного кода Visual Studio Code. Это редактор,

который позиционирует себя для разработки кроссплатформенной разработки веб ориентированных и облачных приложений. Функционал редактора содержит отладчик, инструменты позволяющие работать с Git, средства для рефакторинга. Распространяется как программное обеспечение с открытым исходным кодом.

2.4 Обоснование выбора программных средств

Локальный веб-сервер, который использовался для создания интерактивной карты это OpenServer 5.2.2. В состав которого входят:

- инсталлятор;
- PHP5 с поддержкой GD, MySQL, sqLite;
- MySQL5 с поддержкой транзакций;
- Apache 2;
- Nginx 1;
- MariaDB 5;
- MongoDB 2.4;
- FTP FileZilla;
- PHPMyAdmin 4.

Так же комплект поставки позволяет настроить виртуальный локальный почтовый клиент для настройки возможностей работы с почтой, имеющейся на разрабатываемом сайте.

Системные требования для корректной работы сервера:

Поддерживаемые версии Windows (32-бит и 64-бит): Windows 7 SP1 и все более новые версии;

Частично поддерживаемые версии Windows (32-бит и 64-бит): Windows XP SP3 и Windows Vista;

Минимальные аппаратные требования: 500 МБ свободной RAM и 3 ГБ свободного места на HDD;

Требуется наличие Microsoft Visual C++ 2005-2008-2010-2012-2013-2015 Redistributable Package;

Виртуальный веб сервер позволяет оставаться незаметным при работе в трее Windows;

Система управления виртуальными хостами, основанная на шаблонах. Чтобы создать новый хост, необходимо лишь добавить директорию в каталог /home, править конфигурационные файлы не требуется. По умолчанию уже поддерживаются схемы именования директорий многих популярных хостеров; новые можно без труда добавить;

2.5 Яндекс карты API

Для обозначения мест на карте разными цветами используются маркеры. В API Яндекс.Карт маркеры относятся к географическим объектам, которые реализуются с помощью базового класса `GeoObject`. Экземпляр карты размещает географические объекты в собственном хранилище, реализованном в виде коллекции, ссылка на которую находится в поле `geoObjects`. Добавление географического объекта на карту, его изменение и удаление производится с помощью обращения к этой коллекции. Для создания маркера используется конструктор класса `ymaps.Placemark()`, в который передаются координаты точки и список параметров, а затем с помощью метода карты `addOverlay()` добавляется маркер на карту. Для удаления маркера используется метод `myMap.geoObjects.remove(balloon)` [17].

Для обеспечения взаимодействия пользователя с системой API Яндекс.Карт предоставляют интерфейс для обработки и создания обработчиков событий карты, подробнее описанные в приложении А.

Для построения маршрута реализована функция `route`, в которую передаётся массив точек и дополнительные опции. Данная функция статическая, прокладывает маршрут через заданные точки, а также возвращает объект, который подтверждается при успешном построении маршрута, и

получает в качестве параметра экземпляр класса `router.Route`, соответствующий построенному маршруту. Вычисление маршрута происходит асинхронно в связи с необходимостью обмениваться данными с сервером. Построенный маршрут представляется в виде упорядоченной коллекции путей (`router.Path`), попарно последовательно соединяющих точки остановки. Получить пути маршрута можно с помощью метода `getPaths`.

Расчёт попарных маршрутов между многими точками. Поддерживает режимы передвижения на автомобиле, пешком и с использованием общественного транспорта. При расчёте маршрутов учитывается текущая загруженность дорог или прогноз пробок на нужное время в будущем. Сервис работает по HTTPS-протоколу.

Возможности матрицы состояний:

- Построение маршрутов на территории: России, Абхазии, Азербайджана, Армении, Беларуси, Грузии, Казахстана, Кыргызстана, Молдовы, Таджикистана, Турции, Узбекистана и Украины.
- Учёт текущих, прогнозируемых и исторических пробок.
- Построение маршрутов на заданное время в будущем.
- Построение маршрутов для пешеходов, автомобилистов и общественного транспорта.
- Мультимодальная маршрутизация (комбинации режимов: пешком и на автомобиле, пешком и на общественном транспорте).

Так как API выбранных карт удовлетворяет требованиям, то выбор сервисов, предоставляющих API, был сделан верно.

Выводы по второму разделу

Во втором разделе были произведены обоснование проектных решений, обоснование выбора программных средств реализации, анализ наиболее популярных методов реализации.

3 Разработка интерактивной карты

Определившись с архитектурой, способами хранения информации и взаимодействия с пользователем, необходимо приступить к разработке визуального оформления.

3.1 Описание информационной модели

В основе любого интернет ресурса лежит информация. Основная задача посетителя ресурса – ознакомление с материалами, расположенными на нем. Если посетитель не найдет нужной информации или она будет не достоверной или не полной, он покинет ресурс, и в дальнейшем возможно не вернется к его использованию. Его не остановит даже тот факт, что ресурс имеет запоминающийся дизайн, удобную и продуманную навигацию.

Благодаря информационному наполнению интернет ресурса – оригинальные статьи, актуальные новости, корректные и качественные описания способны привлечь посетителя. Правильно составленное содержание ресурса способно привлекать пользователей и может обеспечить возвращение их на данный ресурс.

Формирование информационного материала для ресурса непосредственно связано с информационной моделью, которая в итоге будет определять, где и какая информация будет предоставлена пользователю.

Информационная модель подразделяется на три части. На обеих сторонах создается информационный дизайн ресурса. Рассматриваются вопросы представления информации с целью ее лучшего восприятия пользователем. В ходе этого определяется дизайн интерфейса, то есть вычисляется наиболее корректное расположение элементов интерфейса [16]. Исходя из информационной стороны уровня необходимо заниматься дизайном навигации, элементами, облегчающими пользователю перемещение внутри ресурса.

Не зависимо от того с каким продуктом работает программным или информационным, стоит уделять большое внимание визуальной составляющей ресурса.

На всех этапах элементы должны работать согласованно. Информационный дизайн, внешний вид навигации, дизайн интерфейса. Все элементы одного уровня должны выполнять единую функцию.

Представленная модель представления разработки с разделением на уровни упрощает процесс создания ресурса. Но не всегда есть возможность разграничить уровни разработки.

Так же не менее важным аспектом при разработке интернет ресурса – это технология. Зачастую именно от технологии будет зависеть, как с ресурсом будет работать пользователь. В ходе развития веб технологий большинство ресурсов используют базы данных. Что в свою очередь позволяет решать более сложные задачи. Такие как создание динамической навигации, которая меняется в зависимости от действия, совершенного пользователем. Но в случае с разрабатываемой интерактивной картой, было принято решение не использовать базу данных, а применить расширенный синтаксис языка программирования, что на начальном этапе позволит уменьшить потребление необходимых ресурсов для интерактивной карты.

3.2 Проектирование интерактивной карты

Перед тем как приступить непосредственно к разработке интерактивной карты, необходимо составить схему разрабатываемого ресурса.

В ходе исследования предметной области, описанной в пункте 1.1, было принято решение составить схему бизнес-процесса создания интерактивной карты и показать основные уровни разработки под видом декомпозиции основных процессов создания.

Верхним уровнем разработки будет являться контекстная диаграмма, изображенная на рисунке 3.1.

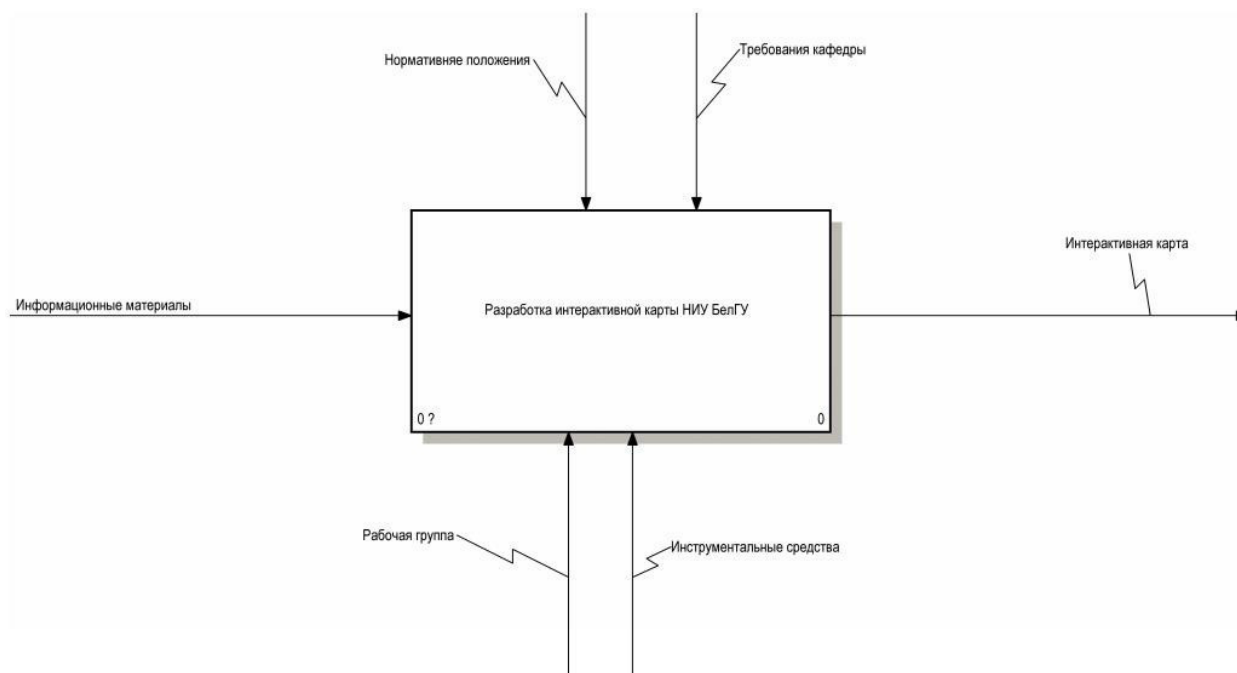


Рисунок 3.1 – Контекстная диаграмма разработка интерактивной карты НИУ «БелГУ»

На диаграмме, представленной на рисунке 3.1 изображена контекстная диаграмма разработки интерактивной карты НИУ «БелГУ» на основании которой будет происходить разработка проекта.

Основными процессом контекстной диаграммы является процесс разработки интерактивной карты.

Входными данными верхнего уровня являются информационные материалы, которые представляют собой информацию, предоставляемую администрацией университета содержащую описание внутренней инфраструктуры университета. А именно адреса корпусов и других структурных подразделений университета. Так же в их состав включено описание каждого корпуса, этажи и адреса их расположения.

Управляющими элементами, влияющими на ход выполнения процесса, являются нормативные положения, которые включают в себя правила построения интерактивных ресурсов, нормы распространения информации, правила предоставления информации [17]. Так же управляющим элементом могут рассматриваться требования, которые могут быть представлены

непосредственно кафедрой, а именно пожелания по внешнему, функционалу и так далее.

Механизмами управления головным процессом является рабочая группа и инструментальные средства.

Под рабочей группой подразумевается разработчик интерактивной картой, так же к этой группе можно отнести разработчиков, занимающихся разработкой API используемых систем. В эту группу можно включить персонал обслуживающий информационные ресурсы интрасети университета.

Выходными данными процесса разработки будет являться конечный продукт, разрабатываемой в ходе дипломной работы являющийся интерактивной картой НИУ «БелГУ».

Для более полного понимания процесса разработки необходимо более детально рассмотреть следующий уровень разработки, представленный в ходе декомпозиции верхнего уровня контекстной диаграммы. Декомпозиция верхнего уровня представлена на рисунке 3.2.

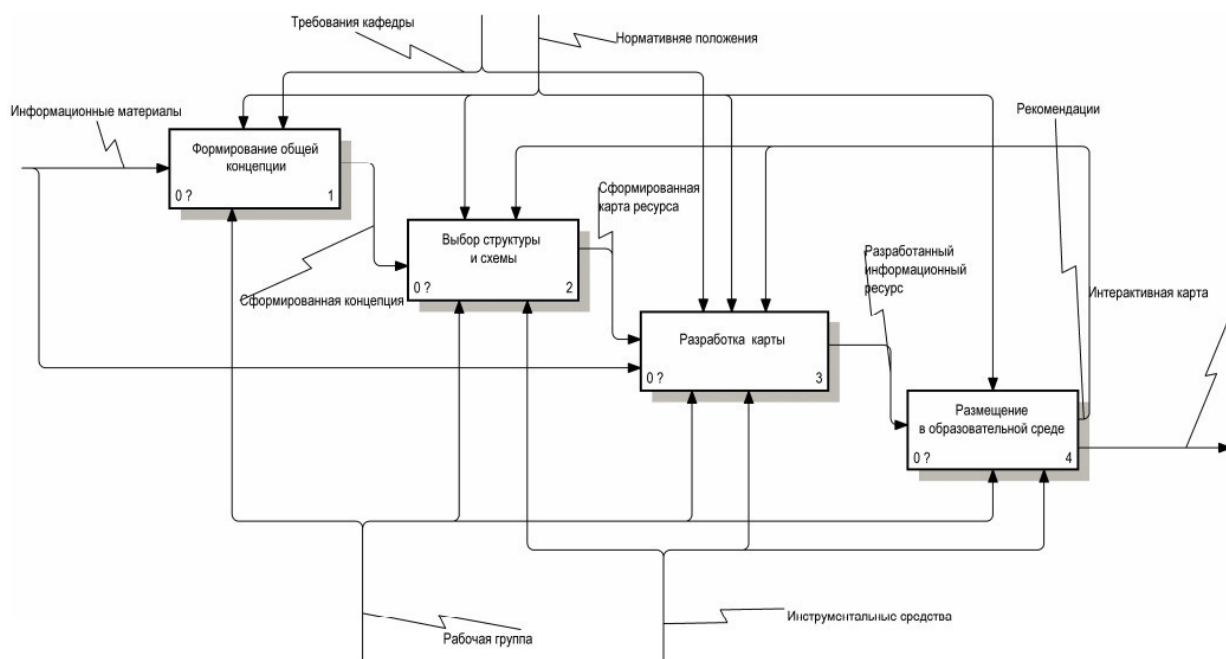


Рисунок 3.2 – Декомпозиция разработки интерактивной карты НИУ «БелГУ»

Принимая во внимание представленную диаграмму, изображенную на рисунке 3.2, можно описать последовательность формирования последовательности разработки карты.

Первым процессом в разработке является формирование общей концепции, что является сбор информации о существующих методах решения поставленной задачи. Структуризация понимания о предоставляемых интерактивных картах, ГИС.

Следующим этапом является выбор структуры и схемы, что представляет собой концептуальную модель будущей карты. Под этим подразумевается разработка и применение методов, повышающих скорость и простоту получения информации пользователем [18].

3.3 Разработка карты

Первым этапом при разработке интерактивной карты является выбор стилового оформления. Исходя из требований, являющихся управляющим действием необходимо соблюдать стилистическое единство разрабатываемой карты с существующими ресурсами интрасети. Было принято решение на страницу с расположением интерактивной карты добавить герб и расшифровку названия университета. Так же необходимо добавить favicon являющийся миниатюрой герба университета. Так же, как и на официальном сайте университета, логотип расположен в левом верхнем углу. В соответствии со шрифтовыми гарнитурами, используемыми на сайте университета, к разрабатываемой карте были применены аналогичные шрифты, представляющие собой семейство Sans, так как исходя из стандартов оформления веб-версий ресурсов данное семейство является предпочтительными из рода шрифтов без засечек. Данная шрифтовая гарнитура была подключена с использованием онлайн-сервиса Google fonts. Но в связи с возможными проблемами подключения к ресурсу, содержащему шрифт, было принято решение продублировать подключение шрифтовой гарнитуры через

шрифтовой файл, что обеспечит одинаковое отображение ресурса для пользователей у которых представленный шрифт не установлен [19]. Выбранная концепция представлена на рисунке 3.3

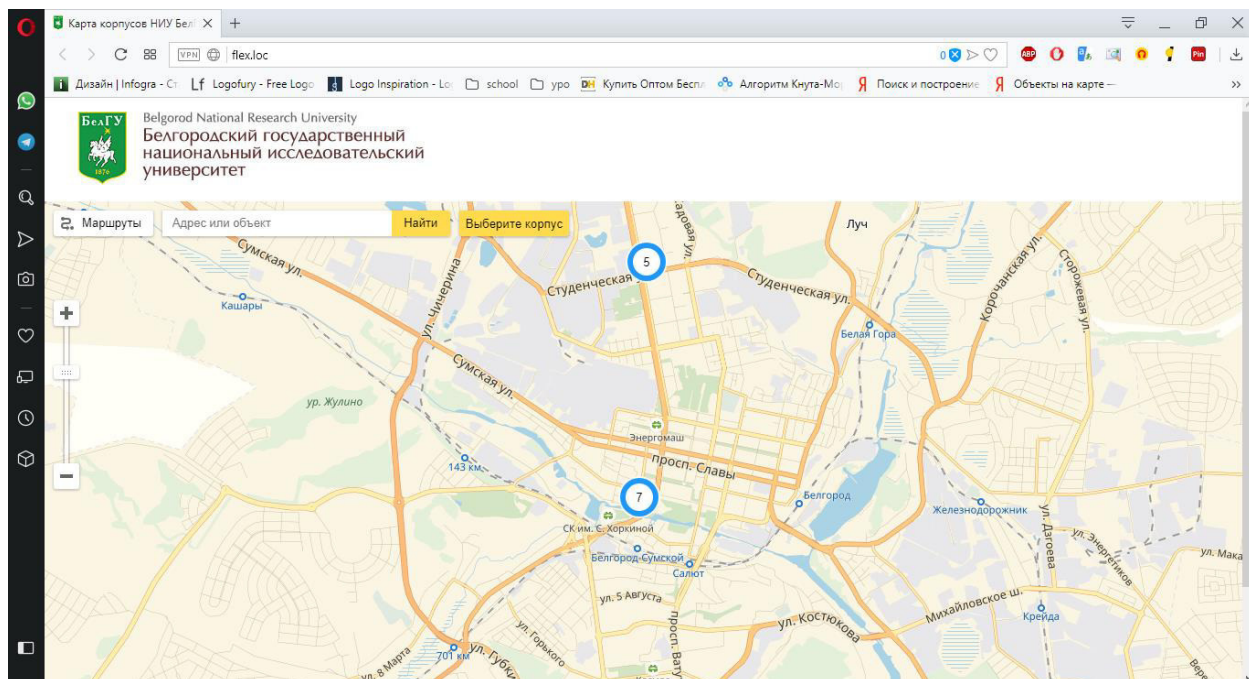


Рисунок 3.3 – Концепция представления интерактивной карты пользователю.

На рисунке 3.3 представлена концепция, выбранная для представления пользователю интерактивной карты. Дизайн шапки интерактивной карты практически полностью повторяет дизайн официального сайта университета, а именно расположением логотипа университета, используемых шрифтовых гарнитур.

Следующим шагом при разработке интерактивной карты является разработка интерактивного интерфейса. Данный этап предполагает добавление динамики разработанной сайте под средством предоставляемого API. На основании, которого и сформирована работа интерактивной карты. Этот этап позволяет оживить страницу при использовании пользователем, под средством отображения самой географической карты, выбора интересующих объектов на ней, построение маршрутов [20]. Для того что бы у пользователя была

возможность взаимодействовать с каждым объектом, необходимо каждому элементу присвоить уникальный идентификатор.

Главным инструментом для работы с объектами предоставляющий структурированное представление документа и определение того, как эта структура может иметь доступ из вне программы DOM (Document Object Model) [21].

На примере кода, можно более точно понять структуру DOM дерева представленной на рисунке 3.4.

```
<html>
  <head>
    |   <title>Заголовок документа</title>
  </head>
  <body>
    |   Тело документа
  </body>
</html>
```

Рисунок 3.4 – Пример кода DOM-дерева

Эту же структуру можно отобразить в виде схемы изображенной на рисунке 3.5.

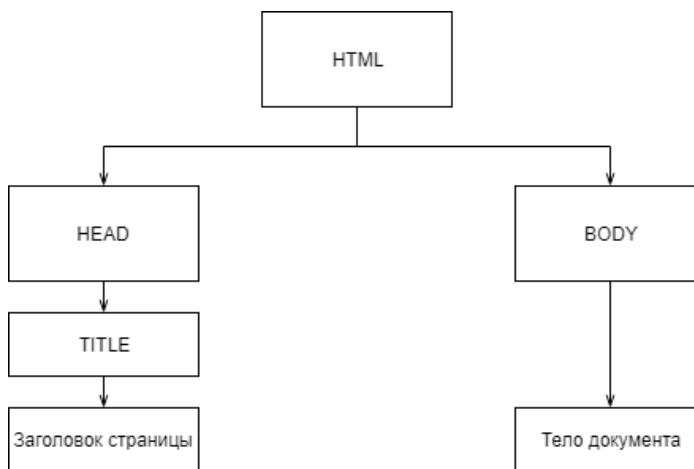


Рисунок 3.5 – Простое DOM-дерево

Верхним тегом дерева изображенном на рисунке 3.5, является тег <html>, и оно начинает расти из него.

Внутри <html> содержится обязательный уровень <head>, который отвечает за заголовок <body> являющимся телом страницы.

Все представленные теги составляют узлы (element-node), текст представленный обычным текстом, не входящим в состав тегов (text-node).

После рассмотрения структуры HTML страницы, можно начать к разработке отображаемых элементов и наделение их интерактивными свойствами.

В ходе загрузки HTML необходимо определиться со сценарием отображения каждого элемента, который будет выполняться в ходе определенного действия [22]. Так же необходимо предусмотреть инициализацию данных при загрузке страницы. Типовой пример такой инициализации представлен на рисунке 3.6.

```
ymaps.ready(init)

function init () {
  var listBoxItems = []
  var myMap = new ymaps.Map('map', {
    center: [50.59424709568916, 36.577003783652714],
    zoom: 13,
    controls: ['searchControl', 'routeButtonControl', 'zoomControl']
  }, {
    searchControlProvider: 'yandex#search'
  })
}
```

Рисунок 3.6 – Пример инициализации карты.

На изображении 3.6 показан один из методов инициализации интерактивной карты. В ходе загрузки DOM дерева при подключении к управляющему файлу карты через инициализированную ссылку в html документе, происходит инициализация интерактивной карты. Этот метод инициализации позволяет облегчить загрузку страницы на стороне пользователя, так как при использовании такой инициализации отображение карты станет возможным только после загрузки страницы полностью

3.3 Используемые методы реализации

Веб-стандарты – это общепринятые технологии, установленные W3C и используемые для интерпретации веб-контента [23]. Данные технологии разработаны для опубликованных во всемирной системе объединенных компьютерных сетях [24].

Основная карта, которая содержит в себе информацию о расположении корпусов, которые имеют неправильную форму, описаны с помощью элемента `polygon`. Координаты каждого такого элемента является полигонами, которые обрамляют контур каждого корпуса. Данный метод отображения предоставляется встроенными возможностями API Яндекс карт изображен на рисунке 3.7.

```
{
  "type": "Feature",
  "id": 1,
  "geometry": { "type": "Point", "coordinates": [50.621442, 36.577685] },
  "properties": {
    "balloonContent": "Контент балуна",
    "iconContent": "Номер балуна",
    "clusterCaption": "Контент представления при кластеризации",
    "hintContent": "Контент показываемый при наведении"
  }
}
```

Рисунок 3.7 – Типовой метод построения метки на карте

Опция `geometry` – позволяет определить тип отображаемого элемента на карте после инициализации. Так же ее входными параметрами служат координаты центра точки, на которой в дальнейшем будет отображен элемент, объявленный в поле `“type”`. Количество координат непосредственно зависит от объекта, который в последствии будет изображаться. В случае с объектом `Point` достаточно двух координат. Пример отображения метода `Point` изображен на рисунке 3.8.

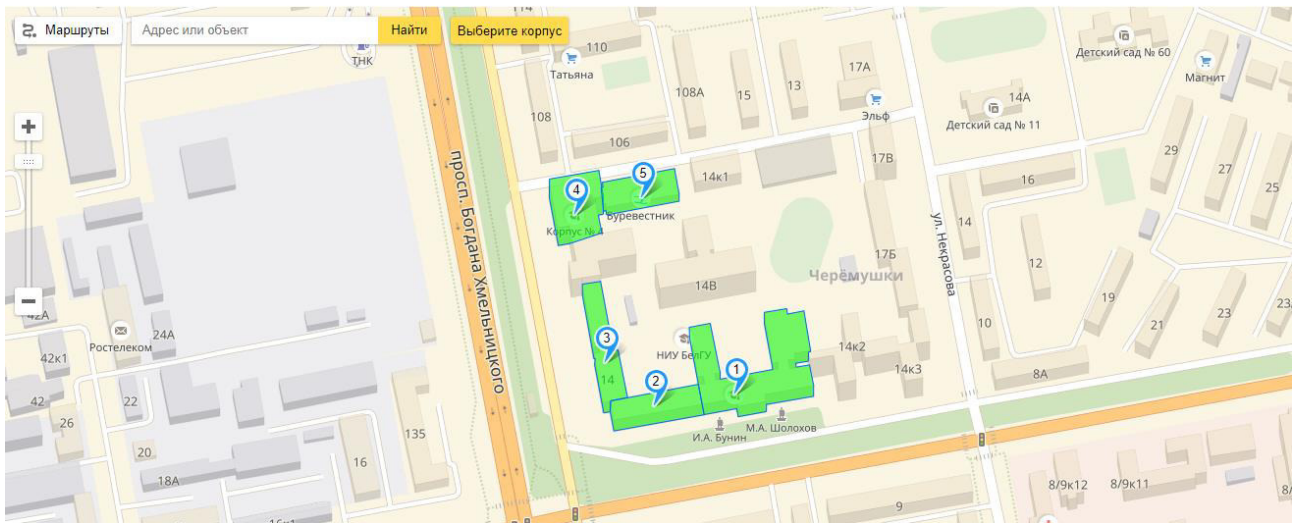


Рисунок 3.8 – Пример отображения метода Point

В данном примере представлен метод отображения Point что является собой маркер, который будет отображать номер корпуса.

Следующей опцией является `balloonContent`. Данная опция позволяет отображать краткое описание корпуса при наведении курсора мыши на контур корпуса или при наведении на номер корпуса.

`iconContent` – опция позволяющая выводить номер корпуса на интерактивной карте в Point.

Для объединения близко расположенных меток в группу (кластер) используется кластеризатор.

Кластеризатор реализуется классом Clusterer. В качестве параметров конструктору можно передать опции кластеризатора: стиль отображения значка кластера, размер его ячейки и т. д.

Для добавления геообъектов в кластеризатор используется метод add. В качестве параметра может быть передан как отдельный геообъект, так и массив геообъектов.

Часто возникает задача показать информацию об объекте в составе кластера. В таких случаях можно открыть балун кластера с нужной выбранной меткой изображен на рисунке 3.9.

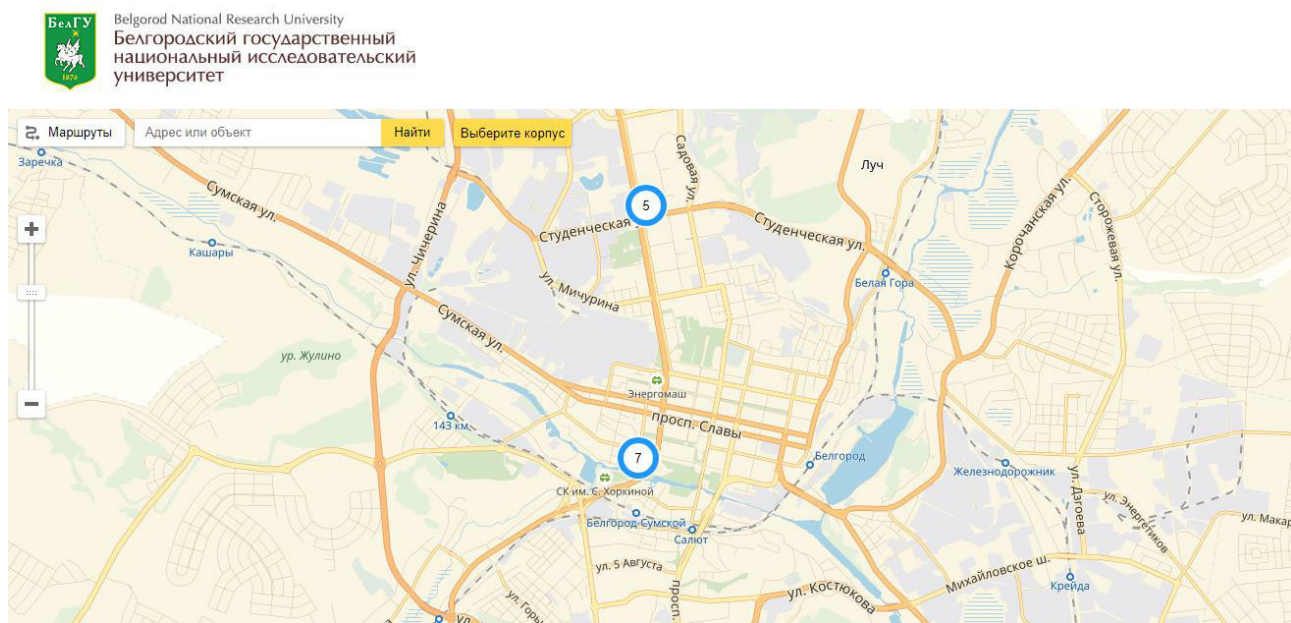


Рисунок 3.9 – Пример отображения кластеризации.

Кластеризация, рисунок 3.9, необходима для более быстрого доступа к списку корпусов без применения метода скроллинга карты либо увеличения масштаба.

Так же был разработан класс `ListBoxLayout` для создания элемента управления в виде раскрывающегося списка. Стандартный макет выпадающего списка изменяет свой внешний вид в зависимости от размеров карты. Если карта имеет большую ширину, то в заголовке выпадающего списка отображается текст. Если карта небольшого размера, то в заголовке списка отображается только иконка [25]. Если для кнопки не задана иконка, то во всех состояниях будет отображаться текст и наоборот, для более быстрого перехода к определенному корпусу представлен на рисунке 3.10.

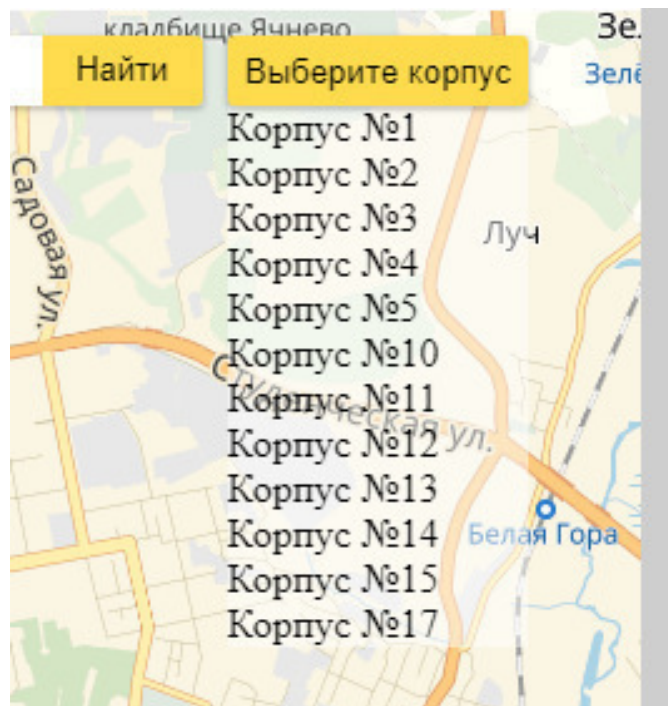


Рисунок 3.10 – Пример класса ListBox layout

Так же было произведено внедрение метода построения маршрута от места положения пользователя `control.routePanel.geolocate('from')` [26]. Работа метода представлена на рисунке 3.11

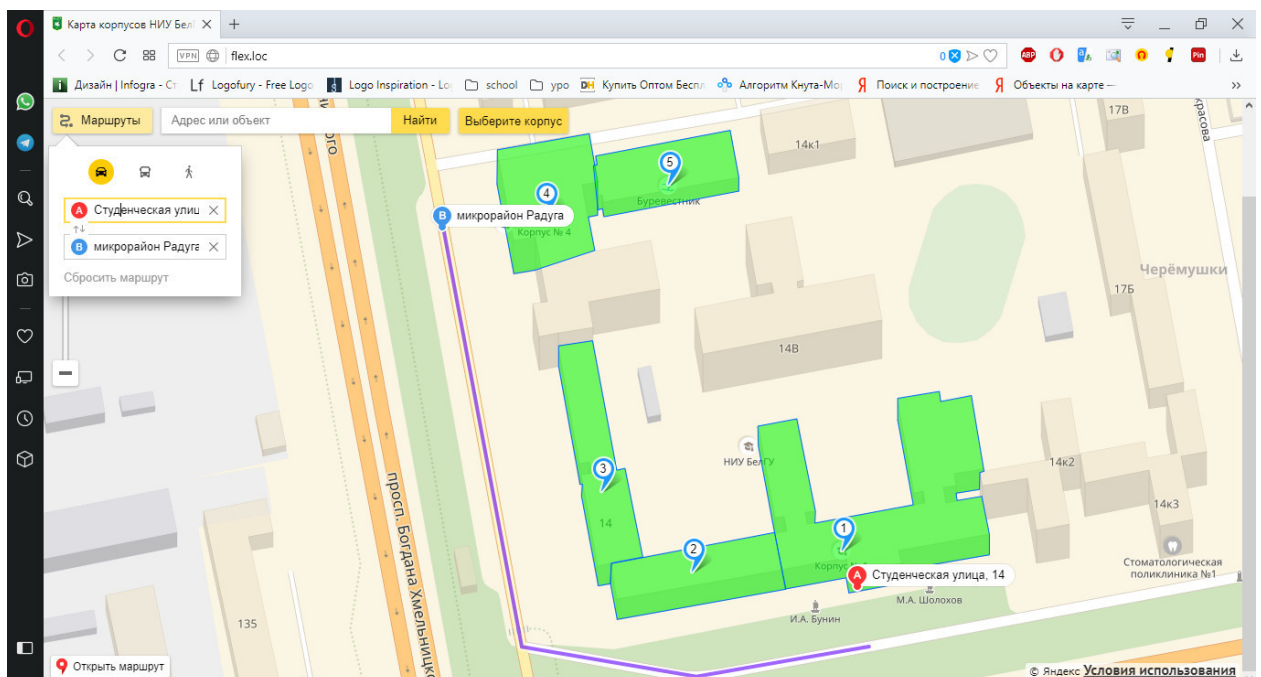


Рисунок 3.11 – Метод построения маршрута

Данный метод рисунок 3.11, позволяет определять местонахождения пользователя основываясь на данных передаваемых мобильной сетью. Точность поиска местоположения пользователя может повыситься на основании совместной работой мобильных сетей и сетей, предоставляемых к сети интернет wi-fi.

Так же было разработано описание каждого корпуса под средством открытия балуна который реагирует на действие OneClick. При клике на балун открывается описание определенного корпуса. Описание корпуса представлено на рисунке 3.12.

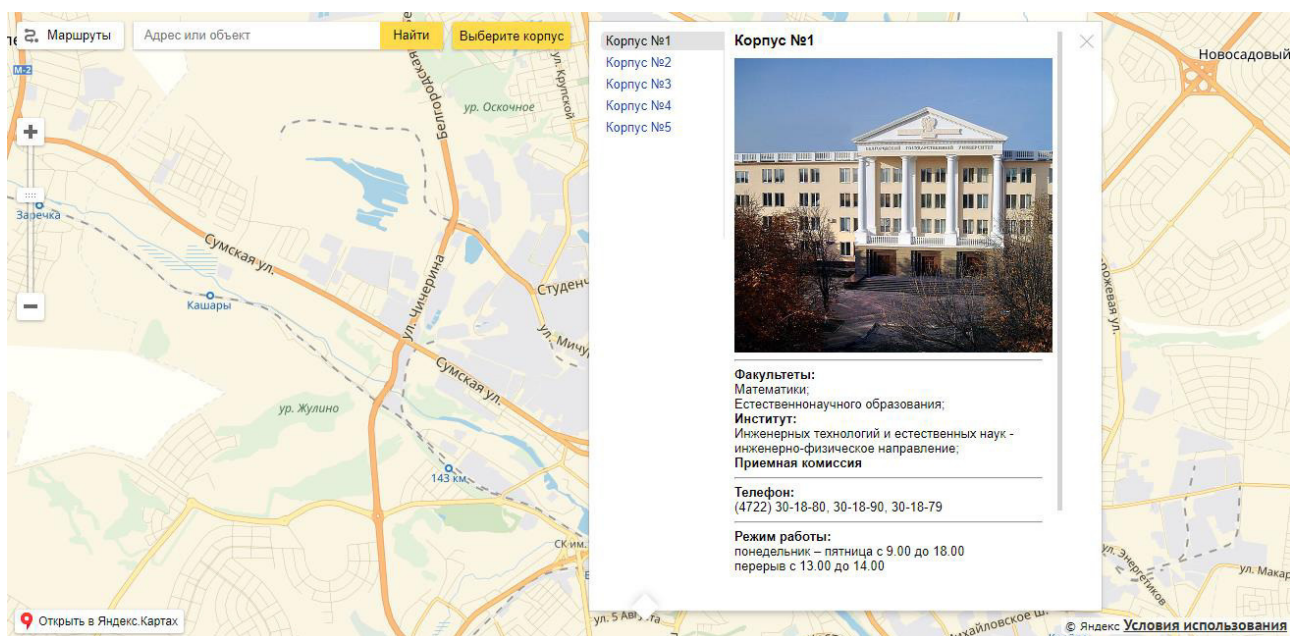


Рисунок 3.12 – Описание корпуса

Содержание описания корпуса связано с интерактивной картой посредством серверного текстового документа, который является расширением функциональности языка JavaScript – JSON. Связь с данным документом осуществляется под средством асинхронных запросов методами AJAX [27].

В ходе разработки интерактивной карты было принято решение в качестве метода расположения контента на странице использовать новый метод таблиц каскадных стилей – flexbox.

Спецификация CSS Flexible Box Layout Module призвана кардинально изменить ситуацию в лучшую сторону при решении огромного количества задач. Flexbox позволяет контролировать размер, порядок и выравнивание элементов по нескольким осям, распределение свободного места между элементами и многое другое.

Основные преимущества flexbox:

- все блоки очень легко делаются “резиновым”, что уже следует из названия “flex”. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство;
- выравнивание по вертикали и горизонтали, базовой линии текста работает шикарно;
- расположение элементов в html не имеет решающего значения. Его можно поменять в CSS. Это особенно важно для некоторых аспектов responsive верстки;
- элементы могут автоматически выстраиваться в несколько строк/столбцов, занимая все предоставленное место;
- множество языков в мире используют написание справа налево rtl (right-to-left), в отличии от привычного нам ltr (left-to-right). Flexbox адаптирован для этого. В нем есть понятие начала и конца, а не права и лева. Т.е. в браузерах с локалью rtl все элементы будут автоматически расположены в реверсном порядке;
- синтаксис CSS правил очень прост и осваивается довольно быстро [30].

Данный метод позволяет с легкостью моделировать адаптивные версии сайтов под различные расширения экранов пользователей, что актуально в настоящий момент. Пример поддержки адаптивности представлен на рисунке 3.13.

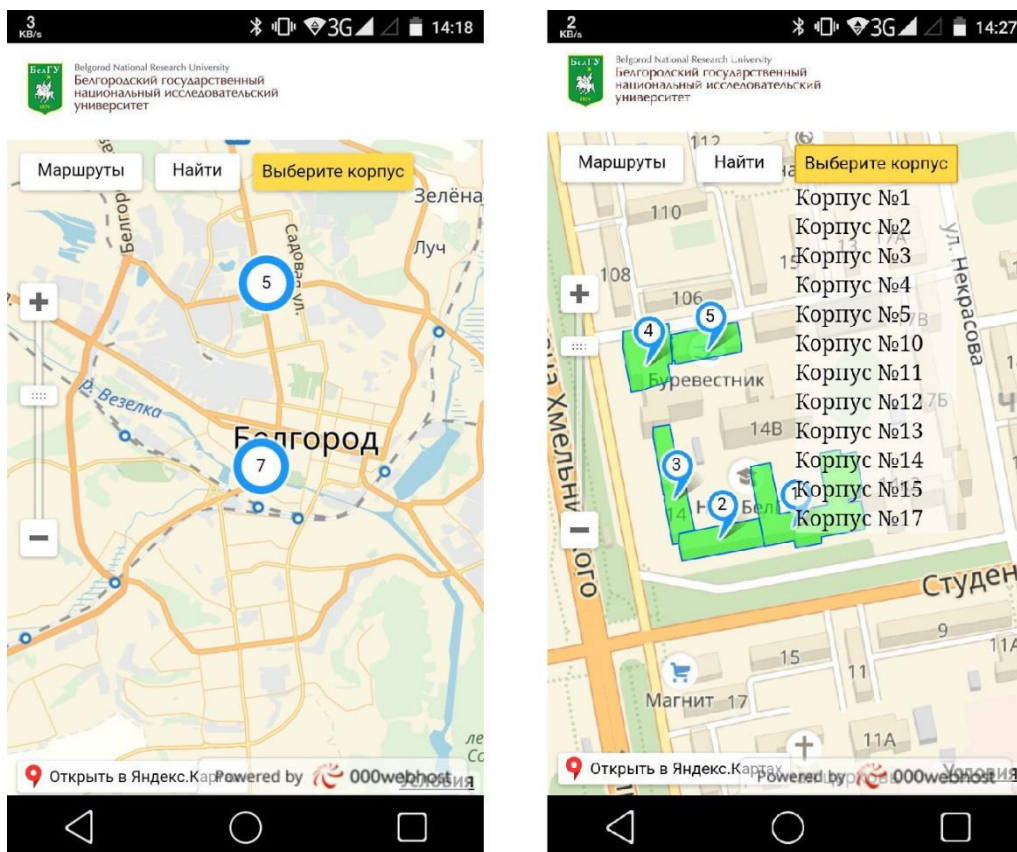


Рисунок 3.13 – Пример адаптивности интерактивной карты.

На рисунке 3.13 можно заметить, что при изменении размера экрана, отображение и масштабы интерактивной карты изменяются для более комфортного представления пользователю. Данный метод, автоматически масштабирует текст, кнопки и другие элементы сайтов для корректного и читаемого отображения на устройстве конечного пользователя. Применение CSS методов flexbox позволяет при минимальном количестве написания кода, представленного в приложении А, подбирать разрешения для каждого экрана. Благодаря этому пользователи с различных устройств будут способны пользоваться интерактивной картой одинаково.

Выводы по третьему разделу

В третьем разделе представлена проектная часть разработки интерактивной карты. Описание информационной модели карты, представлено проектирование интерактивной карты представлены результаты разработки интерактивной карты. Также описаны применяемые методы разработки.

ЗАКЛЮЧЕНИЕ

В процессе написания выпускной квалификационной работы были закреплены и углублены знания, полученные в процессе теоретического обучения, а также были приобретены необходимые умения, навыки и опыт практической работы по специальности. Была разработана интерактивная карта корпусов НИУ «БелГУ».

В ходе выполнения работы решены следующие задачи:

- произведен анализ предметной области;
- разработана интерактивная карта отображающая расположение корпусов;
- создано описание каждого корпуса под средством модальных окон;
- внедрен метод построения маршрута от местоположения пользователя до интересующего объекта.

Целью разработки интерактивной карты является повышение ориентированности абитуриентов, студентов и преподавателей относительно расположения корпусов университета. Получение информации о местонахождении института или факультета при выполнении наименьшего количества действий и в самые кратчайшие сроки.

Так же разработанная интерактивная карта несет собой социально значимую составляющую существования университета. Так как при наличии интерактивной карты есть вероятность повышения престижа университета, что в дальнейшей перспективе будет способствовать приросту новых абитуриентов и молодых преподавателей.

Пояснительная записка состоит из введения, 3 разделов, заключения, списка используемой литературы и приложения.

Во введении рассматривается актуальность и проблематика выбранной темы, определяются предмет и объект исследования, ставится цель формулируются задачи, приводится структура выпускной квалификационной работы.

В первом разделе исследуется предметная область выбранной темы в соответствии с этим определяются возможные и существующие методы реализации поставленной задачи. Проводится анализ бизнес-процесса на основании уже существующей системы «Как есть». На основании данного анализа производится моделирование нового более совершенного процесса. На основании которого производится модерирование схемы «Как должно быть».

Во втором разделе производится обоснование проектных решений, обоснование выбора программных средств реализации, анализ наиболее популярных методов реализации.

В третьем разделе представлена проектная часть разработки интерактивной карты. Описание информационной модели карты, представлено проектирование интерактивной карты представлены результаты разработки интерактивной карты. Также описаны применяемые методы разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Информационный ресурс и его составляющие [Электронный ресурс] / petsu.ru. – Режим доступа: <http://dssp.petsu.ru/p/tutorial/informatics/chapter1/4/44.htm>
2. Тумпаров К.М., Миронова Ю. Н. Геоинформационная система (ГИС) [Электронный ресурс] / К.М. Тумпаров //Научный электронный архив. – Режим доступа: <http://econf.rae.ru/article/8256>.
3. Online Tech Dictionary for Students, Educators and IT Professionals [Электронный ресурс]: Технический словарь для студентов. – Режим доступа: <https://www.webopedia.com/TERM/A/API.html>
4. Svennerberg G. Beginning Google Maps API 3. / G. Svennerberg – Apress, 2010.
5. Яндекс карты [Электронный ресурс]. / yandex.ru. – Яндекс – 2004.
6. 2GIS [Электронный ресурс]. /info.2gis.ru. – 2GIS – 1999.
7. Методология функционального моделирования IDEF0 [Электронный ресурс]: Руководящий документ. – Режим доступа: <https://nsu.ru/smk/files/idef.pdf>
8. Техническое обеспечение университета [Электронный ресурс] Режим доступа: <https://www.bsu.edu.ru/bsu/info/today/>
9. Технологическое обеспечение [Электронный ресурс] / edulib.pgta.ru. – Режим доступа: <http://edulib.pgta.ru/20ИТ.pdf>
10. Дудукало, А.В. HTML и CSS. Разработка и дизайн веб-сайтов, / А.В. Дудукало. – М., 2013. - 480 с.
11. Энди Бадд. Мастерская CSS. Профессиональное применение Web-стандартов/Б. Энди. –Петербург 2007 г 267 стр.
12. Интернет издание [Электронный ресурс] / lifewire.com. – Режим доступа: <https://www.lifewire.com/freeware-definition-4154271>
13. Opensource.com [Электронный ресурс] / opensource.com. – Режим доступа: <https://opensource.com/resources/what-open-source>

14. The IT program for leaders [Электронный ресурс] / Многопредмет. науч. журн.– //Электрон. журн. – Режим доступа: <https://www.techopedia.com/definition/4333/proprietary-software>
15. The IT program for leaders [Электронный ресурс] / Многопредмет. науч. журн.– //Электрон. журн. – Режим доступа: <https://www.techopedia.com/definition/4245/commercial-software>.
16. Филимоненкова Т.Н., UX/UI дизайн и его роль в разработке пользовательского интерфейса сайта/ Т.Н. Филимоненкова – 2017. – С. 158-160.
17. Дронов В.А. PHP, MySQL, HTML 5 и CSS 3. Разработка современных динамических Web-сайтов./ В.А Дронов. – БХВ-Петербург, 2016.
18. Mankov A.O. ПРАВИЛА SEO 2.0 И ПРОГНОЗИРОВАНИЕ ОТКЛИКА /А.О. Манков. –Экономика. Бизнес. Информатика. – 2017. – Т. 3. – №. 4. – С. 351-356.
19. Петрова В.В. Применение типографики для подготовки контента /В.В. Петорова: Вестник Московского государственного университета печати. – 2015. – №. 5.
20. Новиков Д.Ю. Использование интерактивной карты на сайте //Теория. Практика. Инновации. – 2017. – №. 11. – С. 22-26.
21. Медведев Ю.С. Проектирование интерактивных web-приложений /Ю.С. Медведев. – Современные проблемы науки и образования. – 2015. – №. 1-1.
22. Зеленко О.В. Обзор современных Web-технологий /О.В. Зеленко. – Вестник Казанского технологического университета. – 2015. – Т. 18. – №. 2.
23. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера.(+ ftp) 4-е изд./ Н.А. Прохоренко. – БХВ-Петербург, 2015.
24. Куприяновский В.П. и др. Веб Вещей и Интернет Вещей в цифровой экономике /В.П. Куприяновский. –International Journal of Open Information Technologies. – 2017. – Т. 5. – №. 5.

25. Anderson E.L..Internet use and Problematic Internet Use: A systematic review of longitudinal research trends in adolescence and emergent adulthood /E.L. Anderson. –International Journal of Adolescence and Youth. – 2017. – Т. 22. – №. 4. – С. 430-454.
26. API карт [Электронный ресурс]: Документация.– Режим доступа: <https://tech.yandex.ru/maps/doc/jsapi/2.1-dev/ref/reference/control.ListBox-docpage/>
27. RoutePanel [Электронный ресурс]: Документация. – Режим доступа: <https://tech.yandex.ru/maps/doc/jsapi/2.1/ref/reference/control.RoutePanel-docpage/>.
28. Терехина М. О., Лаврентьев Г. В. Разработка Web-приложения для построения логико-смысловых моделей визуализации учебной информации./М.О. Терехина – 2016.
29. Яндекс.Маршрутизация [Электронный ресурс]: Все технологии. – Режим доступа: <https://tech.yandex.ru/maps/doc/jsapi/2.1/ref/reference/control.RoutePanel-docpage/>.
30. The world's largest web developer site [Электронный ресурс] / w3school.com. – Режим доступа: https://www.w3schools.com/Css/css3_flexbox.asp

ПРИЛОЖЕНИЕ А

Листинг страниц

Index.html.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no, maximum-scale=1" />
  <link rel="stylesheet" href="reset.css">
  <link rel="stylesheet" href="style.css">
  <link rel="shortcut icon" href="image/favicon.png" type="image/png">
  <title>Карта корпусов НИУ БелГУ</title>
</head>

<body>
  <div class="wrapper">
    <div class="header">
      <div class="logo">
        <a href="#"!>
          
        </a>
      </div>
      <!-- /.logo -->
    </div>
    <!-- /.header -->
    <div id="map"></div>
    <!-- /#map -->
    <div class="footer"></div>
    <!-- /.footer -->
  </div><!-- /.wrapper -->

  <script src="https://api-maps.yandex.ru/2.1/?lang=ru-RU" type="text/javascript"></script>
  <script src="https://yandex.st/jquery/2.2.3/jquery.min.js" type="text/javascript"></script>
  <script src="object_manager_balloon.js" type="text/javascript"></script>
  <!-- <script src="groups.js" type="text/javascript"></script>
  <script src="object_list.js" type="text/javascript"></script> -->
</body>
</html>
```

Листинг скриптовых файлов

object_manager_balloon.js

```
ymaps.ready(init)
```

```
function init () {  
  var listBoxItems = []  
  var myMap = new ymaps.Map('map', {  
    center: [50.59424709568916, 36.577003783652714],  
    zoom: 13,  
    controls: ['searchControl', 'routeButtonControl', 'zoomControl']  
  }, {  
    searchControlProvider: 'yandex#search'  
  })  
}
```

```
var control = myMap.controls.get('routeButtonControl')
```

```
// Зададим координаты пункта отправления с помощью геолокации.
```

```
control.routePanel.geolocate('from')
```

```
// Контейнер для меню.
```

```
menu = $('<ul class="menu"/>')
```

```
myMap.behaviors.disable([  
  'scrollZoom'
```

```
])
```

```
objectPointManager = new ymaps.ObjectManager({  
  clusterize: true,  
  clusterDisableClickZoom: true,  
  showInAlphabeticalOrder: true  
})
```

```
objectPolyManager = new ymaps.ObjectManager({  
  
})
```

```
myMap.geoObjects.add(objectPointManager)  
myMap.geoObjects.add(objectPolyManager)
```

```
$.getJSON('data.json').done(function (data) {  
  objectPointManager.add(data.point)  
  myMap.geoObjects.add(objectPointManager)  
  objectPolyManager.add(data.poly)  
  myMap.geoObjects.add(objectPolyManager)  
  for (i in data.point.features) {  
    if (data.point.features[i].properties && data.point.features[i].geometry.type == 'Point') {  
      listBoxItems.push(  
        new ymaps.control.ListBoxItem({  
          data: {  
            content: data.point.features[i].properties.balloonContent,
```

```

        center: data.point.features[i].geometry.coordinates,
        zoom: 18
    }
  )))
}
}

```

```

ListBoxLayout = ymaps.templateLayoutFactory.createClass(
"<button id='my-listbox-header' class='btn' data-toggle='dropdown'" +
"{{ data.title }} <span class='caret'></span>" +
'</button>' +
// Этот элемент будет служить контейнером для элементов списка.
// В зависимости от того, свернут или развернут список, этот контейнер будет
// скрываться или показываться вместе с дочерними элементами.
"<ul id='my-listbox'" +
" class='dropdown-menu' role='menu' aria-labelledby='dropdownMenu'" +
" style='display: {% if state.expanded %}block{% else %}none{% endif %};></ul>", {
  build: function () {
    // Вызываем метод build родительского класса перед выполнением
    // дополнительных действий.
    ListBoxLayout.superclass.build.call(this)

    this.childContainerElement = $('#my-listbox').get(0)
    // Генерируем специальное событие, оповещающее элемент управления
    // о смене контейнера дочерних элементов.
    this.events.fire('childcontainerchange', {
      newChildContainerElement: this.childContainerElement,
      oldChildContainerElement: null
    })
  },

  // Переопределяем интерфейсный метод, возвращающий ссылку на
  // контейнер дочерних элементов.
  getChildContainerElement: function () {
    return this.childContainerElement
  },

  clear: function () {
    // Заставим элемент управления перед очисткой макета
    // откреплять дочерние элементы от родительского.
    // Это защитит нас от неожиданных ошибок,
    // связанных с уничтожением dom-элементов в ранних версиях ie.
    this.events.fire('childcontainerchange', {
      newChildContainerElement: null,
      oldChildContainerElement: this.childContainerElement
    })
    this.childContainerElement = null
    // Вызываем метод clear родительского класса после выполнения
    // дополнительных действий.
    ListBoxLayout.superclass.clear.call(this)
  }
}

```

```

    }
  }
}),

// Также создадим макет для отдельного элемента списка.
ListBoxItemLayout = уmaps.templateLayoutFactory.createClass(
  '<li><a>{{ data.content }}</a></li>'
),

// Создадим 2 пункта выпадающего спис
// Теперь создадим список, содержащий 2 пункта.
listBox = new уmaps.control.ListBox({
  items: listBoxItems,
  data: {
    title: 'Выберите корпус'
  },
  options: {
    // С помощью опций можно задать как макет непосредственно для списка,
    layout: ListBoxLayout,
    // так и макет для дочерних элементов списка. Для задания опций дочерних
    // элементов через родительский элемент необходимо добавлять префикс
    // 'item' к названиям опций.
    itemLayout: ListBoxItemLayout
  }
})

listBox.events.add('click', function (e) {
  // Получаем ссылку на объект, по которому кликнули.
  // События элементов списка пропагируются
  // и их можно слушать на родительском элементе.
  var item = e.get('target')
  // Клик на заголовке выпадающего списка обрабатывать не надо.
  if (item != listBox) {
    myMap.setCenter(
      item.data.get('center'),
      item.data.get('zoom')
    )
  }
})

myMap.controls.add(listBox)
})
}

```

data.json

```

{
  "point": {
    "type": "FeatureCollection",

```

```

"features": [
  {
    "type": "Feature",
    "id": 0,
    "geometry": { "type": "Point", "coordinates": [50.621523, 36.578564] },
    "properties": {
      "balloonContent": "Корпус №1",
      "iconContent": "1",
      "clusterCaption": "Корпус №1",
      "hintContent": "Корпус №1",
      "balloonContentBody": "<img class='image' src='image/korp_1.jpg' /> <hr> <div
class='mark'>Факультеты:</div><ul><li>Математики;</li><li>Естественнонаучного образования;</li></ul>
<p><div class='mark'>Институт:</div> Инженерных технологий и естественных наук - инженерно-
физическое направление;</p> <div class='mark'>Приемная комиссия</div> <hr> <div
class='mark'>Телефон:</div>(4722) 30-18-80, 30-18-90, 30-18-79 <hr> <div class='mark'>Режим
работы:</div>понедельник – пятница с 9.00 до 18.00 <br>перерыв с 13.00 до 14.00 ",
      "balloonContentHeader": "Корпус №1"
    }
  },
  {
    "type": "Feature",
    "id": 1,
    "geometry": { "type": "Point", "coordinates": [50.621442, 36.577685] },
    "properties": {
      "balloonContent": "Контент балуна",
      "iconContent": "Номер балуна",
      "clusterCaption": "Контент представления при кластеризации",
      "hintContent": "Контент показываемый при наведении"
    }
  },
  {
    "type": "Feature",
    "id": 2,
    "geometry": { "type": "Point", "coordinates": [50.621743, 36.577154] },
    "properties": {
      "balloonContent": "Корпус №3",
      "iconContent": "3",
      "clusterCaption": "Корпус №3",
      "hintContent": "Корпус №3"
    }
  },
  {
    "type": "Feature",
    "id": 3,
    "geometry": { "type": "Point", "coordinates": [50.622769, 36.576818] },
    "properties": {
      "balloonContent": "Корпус №4",
      "iconContent": "4",
      "clusterCaption": "Корпус №4",

```

```

"balloonContentBody": "<img class='image' src='image/korp_4.jpg' /> <hr> <div
class='mark'>Факультеты:</div><ul><li>Историко-филологический;</li><li>Психологии</li></ul><hr>
<div class='mark'>Телефон:</div>(4722) 30-18-51<hr>",
  "hintContent": "Корпус №4"
}
},
{
  "type": "Feature",
  "id": 4,
  "geometry": { "type": "Point", "coordinates": [50.622885, 36.577542] },
  "properties": {
    "balloonContent": "Корпус №5",
    "iconContent": "5",
    "clusterCaption": "Корпус №5",
    "hintContent": "Корпус №5"
  }
},
{
  "type": "Feature",
  "id": 5,
  "geometry": { "type": "Point", "coordinates": [50.593511, 36.576539] },
  "properties": {
    "balloonContent": "Корпус №10",
    "iconContent": "10",
    "clusterCaption": "Корпус №10",
    "hintContent": "Корпус №10"
  }
},
{
  "type": "Feature",
  "id": 6,
  "geometry": { "type": "Point", "coordinates": [50.594515, 36.576719] },
  "properties": {
    "balloonContent": "Корпус №11",
    "iconContent": "11",
    "clusterCaption": "Корпус №11",
    "hintContent": "Корпус №11"
  }
},
{
  "type": "Feature",
  "id": 7,
  "geometry": { "type": "Point", "coordinates": [50.594093, 36.577247] },
  "properties": {
    "balloonContent": "Корпус №12",
    "iconContent": "12",
    "clusterCaption": "Корпус №12",
    "balloonContentBody": "<img class='image' src='image/korp_12.jpg' /> <hr> <div
class='mark'>Институт:</div><ul><li>Межкультурной коммуникации и международных

```

отношений<div class='mark'>Администрация</div><hr> <div class='mark'>Телефон:</div>(4722) 24-54-01, 24-54-00<hr>",

```
"hintContent": "Корпус №12"
  }
},
{
  "type": "Feature",
  "id": 8,
  "geometry": { "type": "Point", "coordinates": [50.594112, 36.575986] },
  "properties": {
    "balloonContent": "Корпус №13",
    "iconContent": "13",
    "clusterCaption": "Корпус №13",
    "hintContent": "Корпус №13"
  }
},
{
  "type": "Feature",
  "id": 9,
  "geometry": { "type": "Point", "coordinates": [50.593757, 36.575176] },
  "properties": {
    "balloonContent": "Корпус №14",
    "iconContent": "14",
    "clusterCaption": "Корпус №14",
    "hintContent": "Корпус №14"
  }
},
{
  "type": "Feature",
  "id": 10,
  "geometry": { "type": "Point", "coordinates": [50.593343, 36.575841] },
  "properties": {
    "balloonContent": "Корпус №15",
    "iconContent": "15",
    "clusterCaption": "Корпус №15",
    "hintContent": "Корпус №15"
  }
},
{
  "type": "Feature",
  "id": 11,
  "geometry": { "type": "Point", "coordinates": [50.594556, 36.575675] },
  "properties": {
    "balloonContent": "Корпус №17",
    "iconContent": "17",
    "clusterCaption": "Корпус №17",
    "balloonContentBody": "<img class='image' src='image/korp_17.jpg' /> <hr> <div
class='mark'>Факультеты:</div><ul><li>Журналистики;</li><li>Фармацевтический;</li></ul><div
class='mark'>Институт:</div><ul><li>Межкультурной коммуникации и международных
отношений</li></ul><hr> <div class='mark'>Телефон:</div>(4722) 24-54-01, 24-54-00<hr>",
```

```

    "hintContent": "Корпус №17"
  }
}
],
},
"poly": {
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": 12,
      "options": { "fillColor": "#00FF0088" },
      "properties": {},
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [50.622009273633964, 36.578889376590396],
            [50.62209117595371, 36.578889376590396],
            [50.622116770398094, 36.57913345761075],
            [50.622079231874885, 36.57914150423779],
            [50.62210141373315, 36.5793212122418],
            [50.62181917140516, 36.57939555744998],
            [50.62181405248484, 36.57937141756884],
            [50.621750919088115, 36.57936873535982],
            [50.621737268612755, 36.579231942700076],
            [50.62170996765009, 36.5792373071181],
            [50.621725324447155, 36.57938214640492],
            [50.62158391942547, 36.57942704060463],
            [50.6215071351756, 36.57942704060463],
            [50.621450826645564, 36.57891205647376],
            [50.62140089209172, 36.57891088517659],
            [50.62136505931725, 36.57860243113987],
            [50.62142136795053, 36.57858633788579],
            [50.621379736048375, 36.57822944516085],
            [50.62191381201173, 36.578071194828944],
            [50.621987182993024, 36.57806851261993],
            [50.6220144837939, 36.57829650038618],
            [50.621669810013564, 36.5784064709558],
            [50.62162544587714, 36.57840378874678],
            [50.62169369837728, 36.57898046368498],
            [50.622009273633964, 36.578889376590396]
          ]
        ]
      }
    }
  ]
}
},
{
  "type": "Feature",
  "id": 13,
  "options": { "fillColor": "#00FF0088" },

```

```

"properties": {},
"geometry": {
  "type": "Polygon",
  "coordinates": [
    [
      [50.621590466453775, 36.578166413249],
      [50.62137888291218, 36.57823346847438],
      [50.62126626547557, 36.57724373334788],
      [50.62140106510259, 36.577203500212654],
      [50.6214744368882, 36.57720886463068],
      [50.621590466453775, 36.578166413249]
    ]
  ]
},
{
  "type": "Feature",
  "id": 14,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.62139988525295, 36.57720315099515],
        [50.62139135364229, 36.57713341356079],
        [50.621725791615155, 36.57703417182722],
        [50.62177015566717, 36.57703953624522],
        [50.621770155656385, 36.57702612520018],
        [50.6222103811842, 36.576900061376435],
        [50.62228716427964, 36.576900061376435],
        [50.62230593346156, 36.57709854484354],
        [50.62182305120698, 36.57724070192134],
        [50.62182987643264, 36.5772889816836],
        [50.62149885184086, 36.57739358783517],
        [50.62147666969684, 36.57721119762217],
        [50.62139988525295, 36.57720315099515]
      ]
    ]
  }
},
{
  "type": "Feature",
  "id": 15,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [

```



```

    [50.62256968898317, 36.57676326871643],
    [50.62255603874678, 36.576637204892734],
    [50.622890468392015, 36.57653796315919],
    [50.62301843828696, 36.576540645368205],
    [50.62307815745143, 36.57707976938019],
    [50.62290070599649, 36.57712804914245],
    [50.622774442051295, 36.57712804914245],
    [50.62276761696347, 36.57707440496216],
    [50.62264305893563, 36.57711195588837],
    [50.62256968898317, 36.57676326871643]
  ]
]
}
},
{
  "type": "Feature",
  "id": 16,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.62279406419483, 36.57713073135144],
        [50.62279918300789, 36.577157553441594],
        [50.62276847012106, 36.57716560006864],
        [50.6228640212583, 36.57795685172801],
        [50.6229169155539, 36.57795685172801],
        [50.62309095442691, 36.577905889756735],
        [50.62299540375327, 36.57711732030637],
        [50.622962984730364, 36.577125366933416],
        [50.622959572200344, 36.57710390926128],
        [50.62279406419483, 36.57713073135144]
      ]
    ]
  }
},
{
  "type": "Feature",
  "id": 17,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.593622499469575, 36.57627451478483],
        [50.59405957519912, 36.57643544732577],
        [50.594029270277474, 36.5766567295695],
        [50.593992136050254, 36.57665002404693],

```

```

    [50.59398786774143, 36.57666611730106],
    [50.593904208898046, 36.57667148171908],
    [50.59369768352715, 36.576595361851055],
    [50.59358164616434, 36.576556792909344],
    [50.593622499469575, 36.57627451478483]
  ]
]
}
},
{
  "type": "Feature",
  "id": 18,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.594362162299255, 36.576278861940146],
        [50.59441679612729, 36.57627617973113],
        [50.594502161359635, 36.57631373065734],
        [50.5945106978703, 36.576297637403265],
        [50.59458581912909, 36.57629495519425],
        [50.59465581837541, 36.57633518832947],
        [50.59465069649389, 36.57637005704662],
        [50.594688257015015, 36.57637810367371],
        [50.59458752642904, 36.57705402034544],
        [50.594409966902276, 36.57697891849302],
        [50.59425118713976, 36.576973554074996],
        [50.59436216229925, 36.576278861940146]
      ]
    ]
  }
},
{
  "type": "Feature",
  "id": 19,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.59424709568916, 36.577003783652714],
        [50.59425221762766, 36.57697427935352],
        [50.59441099738669, 36.57698232598056],
        [50.594597093400765, 36.57706011004207],
        [50.594585995959655, 36.57712045974488],
        [50.59461886147053, 36.57712917692417],
        [50.594608831090156, 36.57717376864903],

```



```

    ]
  ]
}
},
{
  "type": "Feature",
  "id": 21,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.59376706481484, 36.57531232396876],
        [50.59380462605589, 36.575044103067285],
        [50.59423828547478, 36.575194306772126],
        [50.5942297489108, 36.57525063316143],
        [50.59435779720703, 36.57529623071468],
        [50.59433560219411, 36.575526900689965],
        [50.59409657831121, 36.5754276589564],
        [50.59393267551482, 36.57536596814907],
        [50.59376706481484, 36.57531232396876]
      ]
    ]
  }
},
{
  "type": "Feature",
  "id": 22,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.5938043302427, 36.57504338084728],
        [50.5935426814204, 36.57495620905429],
        [50.593458594940294, 36.574952185740806],
        [50.59345091190669, 36.574991077771486],
        [50.59342274076042, 36.574976325621954],
        [50.59332542215099, 36.57498169003993],
        [50.593310056027775, 36.57511043607269],
        [50.59323322537642, 36.575107753863676],
        [50.59309151516722, 36.57612967549834],
        [50.59316663870673, 36.57616186200647],
        [50.593141028423986, 36.57633352338347],
        [50.59329298257354, 36.57638850866825],
        [50.593284445839984, 36.57645422278912],
        [50.59357981601417, 36.57655614673159],
        [50.5938043302427, 36.57504338084728]
      ]
    ]
  }
}

```

```

    ]
  ]
}
},
{
  "type": "Feature",
  "id": 23,
  "options": { "fillColor": "#00FF0088" },
  "properties": {},
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [50.59452792799162, 36.57607435735981],
        [50.5945149098023, 36.57606949436216],
        [50.59454340040404, 36.57586039309992],
        [50.59449959742862, 36.57586026728774],
        [50.594445257140045, 36.575836566388354],
        [50.59440826996661, 36.57577087896759],
        [50.594401727517194, 36.575708530958025],
        [50.594406992825384, 36.57561566614591],
        [50.59445913733842, 36.57562019571115],
        [50.59446301462063, 36.57558222735852],
        [50.59433372197016, 36.57552816575183],
        [50.594342258515184, 36.57548525040759],
        [50.5943610389124, 36.575276038104406],
        [50.59443786771306, 36.57527872031342],
        [50.59475201084574, 36.57538869088304],
        [50.59480664421887, 36.57545574610842],
        [50.59482371713489, 36.5755442590059],
        [50.594825424426155, 36.57561131423128],
        [50.59476054731451, 36.57608874743594],
        [50.594734937903574, 36.57611288731706],
        [50.5946888409286, 36.576115569526074],
        [50.59465469499194, 36.576332828456295],
        [50.594586403043934, 36.576297959739094],
        [50.59450786718059, 36.576297959739094],
        [50.59451981829871, 36.57622017567766],
        [50.5945044525822, 36.57621481125956],
        [50.5945279279916, 36.57607435735981]
      ]
    ]
  }
}
]
}
}
}

```

Листинг таблицы каскадных стилей

Style.css

```
@import
url('https://fonts.googleapis.com/css?family=PT+Sans:400,700&subset=cyrillic,cyri
lic-ext');

.wrapper {
  display: -webkit-box;
  display: -moz-box;
  display: -ms-flexbox;
  display: -webkit-flex;
  display: flex;
  -webkit-flex-flow: row wrap;
  flex-flow: row wrap;
}

.wrapper > * {
  padding: 0;
  flex: 1 100%;
}

.header {
}

.logo img{
  padding: 15px;
  padding-left: 36px;
  width: 380px;
  height: 80px;
}

.btn text{
  display: inline-block;
  position: relative;
  border: none;
  text-decoration: none;
  white-space: nowrap;
  font-size: 13px;
  line-height: 26px;
  padding: 0 12px;
}

.btn{
  box-shadow: 0 1px 2px 1px rgba(0,0,0,.15), 0 2px 5px -3px rgba(0,0,0,.15);
  border-color: transparent;
  background-color: #ffdb4d;
}
```

```

    box-sizing: border-box!important;
    border-width: 1px;
    border-style: solid;
    border-radius: 3px;
    background-clip: border-box;
    color: #000;
    vertical-align: middle;
    text-decoration: none;
    font-family: Arial,Helvetica,sans-serif;
    cursor: pointer;
    -webkit-transition: background-color .15s ease-out,border-color .15s ease-
out,opacity .15s ease-out;
    transition: background-color .15s ease-out,border-color .15s ease-out,opacity
.15s ease-out;
    display: inline-block;
    text-align: left;
    height: 28px;
}

#my-listbox{
    width: 100%;
    height: 100%;
    background-color: rgba(255, 255, 255, 0.5);
    cursor: pointer;
}

#map {
    padding: 0;
    margin: 0;
    width: 100%;
    height: 100vh;
}

.image {
    display: flex;
    align-items: center;
    width: 400px;
}

.container{
    display: flex;
    flex-direction: column;
    align-content: center
}

.description{
}

.mark{
    font-weight: bold;
}

```

```

}

a:link {
  font-size: 16px;
  font-weight: bold;
  text-decoration: none;
  color: #3F4BA2;
  font-family: 'PT Sans', Tahoma, Verdana, Helvetica, Arial, sans-serif;
}

.ballon_header {
  font-size: 16px;
  margin-top: 0;
  margin-bottom: 10px;
  color: #708090;
}

.ballon_body {
  font-size: 14px;
  text-align: left;
}

.ballon_footer {
  font-size: 12px;
  text-align: right;
  border-top: 1px solid #7D7D7D;
  color: #7D7D7D;
  margin-top: 10px; }

.footer {
  background: lightgreen;
}

@media all and (min-width: 800px) {
  .header {
    order: 1;
  }

  #map{
    order: 2;
  }

  .footer {
    order: 3;
  }
}

@media all and (max-width: 625px) {
  .logo img{
    padding: 15px;

```



```
padding-left: 15px;
width: 190px;
height: 40px;
}

#map {
  width: 100%;
  height: 500px;
}

.menu{
  padding-top: 0;
  padding-left: 15px;
  margin: 0;
  margin-right: 0;
}

  .btn{
    margin-top: 10px
  }
}

@media all and (max-width: 400px) {
  .btn{
    margin-top: 0;
  }
}
```


Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« ____ » _____ Г.

(подпись)

(Ф.И.О.)