МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гродненский государственный университет имени Янки Купалы»

Факультет математики и информатики

Кафедра программного обеспечения интеллектуальных и компьютерных систем

УЛЕЗЛО ДМИТРИЙ СЕРГЕЕВИЧ

**Реализация Интернет-ресурса социально-значимого характера с учетом требований стандарта «Среда обитания для физически ослабленных лиц»**

Дипломная работа

студента 5 курса специальности

1-40 01 01 «Программное обеспечение информационных технологий»

дневной формы обучения

|  |  |
| --- | --- |
| Допущен к защите  Заведующий кафедрой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Рудикова Л. В.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2015 г. | **Научный руководитель**  Кадан Александр Михайлович,  Заведующий кафедрой **cистемного программирования и компьютерной безопасности,**  канд. тех. наук, доцент |
|  |  |

Гродно, 2015

**РЕЗЮМЕ**

Улезло Дмитрий Сергеевич.

Реализация Интернет-ресурса социально-значимого характера с учетом требований стандарта «Среда обитания для физически ослабленных лиц».

Ключевые слова – JavaScript, Java, JQuery, Spring MVC, Hibernate.

Цель дипломной работы – спроектировать и реализовать Интернет-ресурса социально-значимого характера с учетом требований стандарта «Среда обитания для физически ослабленных лиц».

Предмет исследования – информационно-справочные системы.

Объект исследования – информационные сервисы на основе клиент-серверных web-приложений, концепция Model-View-Controller.

Методы исследования − анализ предметной области, моделирование и проектирование программных модулей.

Проект представляет собой Интернет-ресурс информационного поиска и визуализации данных на основе единой политики интеграции информационно-новостных ресурсов, баз данных организаций, информационных и картографических сервисов компании Google.

**SUMMARY**

Ulezla Dzmitry Siarheevich.

Implementation of socially important Internet resource, which matches the standards "Habitat for people with physical disabilities".

48 страниц, 34 иллюстраций, 1 диаграмма, 1 схема, 12 таблиц, 1 приложения.

Keywords – JavaScript, Java, JQuery, Spring MVC, Hibernate.

The purpose of diploma - design and implement socially important Internet resource, which matches the standards "Habitat for people with physical disabilities".

Subject of study – information and referral system.

Object of research – information services based on client-server web-based application, the concept of Model-View-Controller.

Research methods – domain analysis, modeling and design of software modules.

The project is an online resource of information searching and data visualization based on a common policy of integration of information and news resources, organizations databases, information and mapping services of Google

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 4](#_Toc414882424)

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 5](#_Toc414882425)

[ВВЕДЕНИЕ 6](#_Toc414882426)

[ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ И ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ 9](#_Toc414882427)

[1.1 Обзор существующих решений 9](#_Toc414882428)

[1.1.1 Проект «Доступная жизнь» 9](#_Toc414882429)

[1.1.2 Социальный проект «Город без барьеров» 10](#_Toc414882430)

[1.2 Выбор программных средств 11](#_Toc414882431)

[1.3. Выводы по главе 1 18](#_Toc414882432)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ приложения 19](#_Toc414882433)

[ГЛАВА 3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ 20](#_Toc414882434)

[ЗАКЛЮЧЕНИЕ 21](#_Toc414882435)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc414882436)

[ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 23](#_Toc414882437)

# ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ОС – Операционная система.

ИТ – информационные технологии.

ПК – персональный компьютер.

ПО – программное обеспечение.

ПС – программное средство.

HTML (HyperText Markup Language) − язык для структурирования и представления содержимого всемирной паутины.

SQL (Structured Query Language) – формальный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных.

БД – база данных.

СУБД – Система управления базами данных.

MVC - Model-View-Controller

# ВВЕДЕНИЕ

Формирование информационного общества является приоритетным направлением для любого современного государства. Именно поэтому Президентом Республики Беларусь ключевое внимание отведено вопросам информатизации и интеграции информационно-коммуникационных технологий не только в деятельность государственных органов, но и в жизнь белорусского общества.

Необходимо развивать программы и проекты, устраняющие информационные неравенства для граждан, имеющих ограниченные материальные и физические возможности. Данного курса придерживаются многие страны мира.

Так одним из направлений крупнейшей программы информатизации Европейского Союза eEurope («Электронная Европа») стала подпрограмма e-Inclusion – обеспечение компьютерного образования и вовлечение в жизнь общества людей с различными физическими недостатками. Данная подпрограмма реализовывалась во всех странах – участницах Евросоюза.

В Российской Федерации в 2011 году была принята к реализации государственная программа «Доступная среда», которая предусматривает комплексное оборудование зданий, территорий, транспорта с учётом потребностей людей-инвалидов во всех регионах страны, а также их информационное обеспечение.

По информации Республиканской ассоциации инвалидов-колясочников, на сегодняшний день в Беларуси проживает около 20 000 инвалидов-колясочников. И таких людей ежегодно становится на 400 человек больше.

Государство предпринимает ряд мер для того, чтобы люди в инвалидных креслах чувствовали себя равноправными членами общества, включенными во все сферы жизни. Чтобы они жили полноценной жизнью, свободно передвигались по улицам городов, обучались в университетах, посещали государственные учреждения, торговые центры, театры, спортивные мероприятия, кафе.

Важным фактором успешной социальной адаптации и полноценной жизни является свободный доступ к актуальной информации об различных объектах и инфраструктурах города. Интернет-ресурс социально-значимого характера с учетом требований стандарта «Среда обитания для физически ослабленных лиц» необходим для улучшения качества жизни людей с ограниченными возможностями.

Социально-значимые цели проекта:

1. Предоставление людям с ограниченными физическими возможностями прямого доступа к информации, информационная поддержка;
2. Информирование учреждений, организаций, служб и общества о потребностях людей с ограниченными физическими возможностями;
3. Мотивация учреждений, организаций, служб к созданию и совершенствованию в них безбарьерной среды;

Техническая цель проекта – реализация Интернет-ресурса с использованием технологий разработки Интернет-ресурсов, информационного поиска и визуализации данных на основе единой политики интеграции информационно-новостных ресурсов, баз данных организаций, концепции MVC, информационных и картографических сервисов компании Google.

Задачи проекта:

1. Подтверждение активной позиции и деятельности ГрГУ им. Я. Купалы в решении задач социальной реабилитации людей с ограниченными физическими возможностями;
2. Вовлечение людей с ограниченными физическими возможностями в активную социальную жизнь;
3. Провести обзор существующих приложений для информирования лиц с ограниченными возможностями;
4. Рассмотрение существующих фреймворков для работы с картографическим контентом и определить среду реализации;
5. Проектирование сервиса представления данных об элементах безбарьерной среды организаций г. Гродно;
6. Реализация программных компонентов сервиса, формулировка базы данных сервиса;
7. Проведение тестирования работы приложения;
8. Внедрение разработку;

Направление работы актуально и имеет акцентированную социальную значимость. Проект ориентирован в первую очередь на информирование инвалидов-колясочников о наличии элементов безбарьерной среды в городе Гродно, учреждениях и организациях. В дальнейшем планируется расширение его географии в пределах Республики Беларусь. Потенциальная аудитория проекта – инвалиды-колясочники, их семьи, а также все те пользователи сети Интернет, которым интересны и полезны ресурсы проекта и небезразличны проблемы людей с ограниченными возможностями.

# ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ И ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

## 1.1 Обзор существующих решений

Существует большое количество систем с возможностью демонстрации объектов и их описания, но все они имеют направленность на физически здорового человека.

На данный момент систем с такой объёмной и актуальной информационной базой и направленностью на лица с ограниченными возможностями в г. Гродно нет.

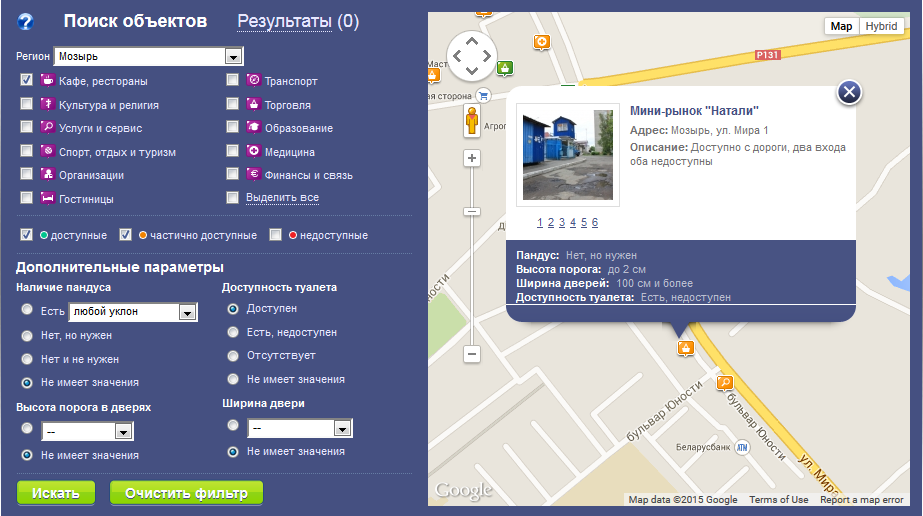
Рассмотрим похожие решения:

### 1.1.1 Проект «Доступная жизнь»

Сайт – Жизнь без барьеров является результатом реализации проекта «Доступная жизнь», который был реализован в 2010 году при поддержке отдела информации, образования и культуры посольства США в Беларуси.

Реализация проекта и идея проекта принадлежит Мозырской межрайонной организации общественного объединения "Республиканская ассоциация инвалидов-колясочников".

Спроектировала и реализовала этот проект в Интернете компании Astronim.

[](http://www.freeway.by/)

**Рисунок 1.1 – Проект «Доступная жизнь»**

Основные преимущества:

* Бесплатный.
* Простой в управлении.

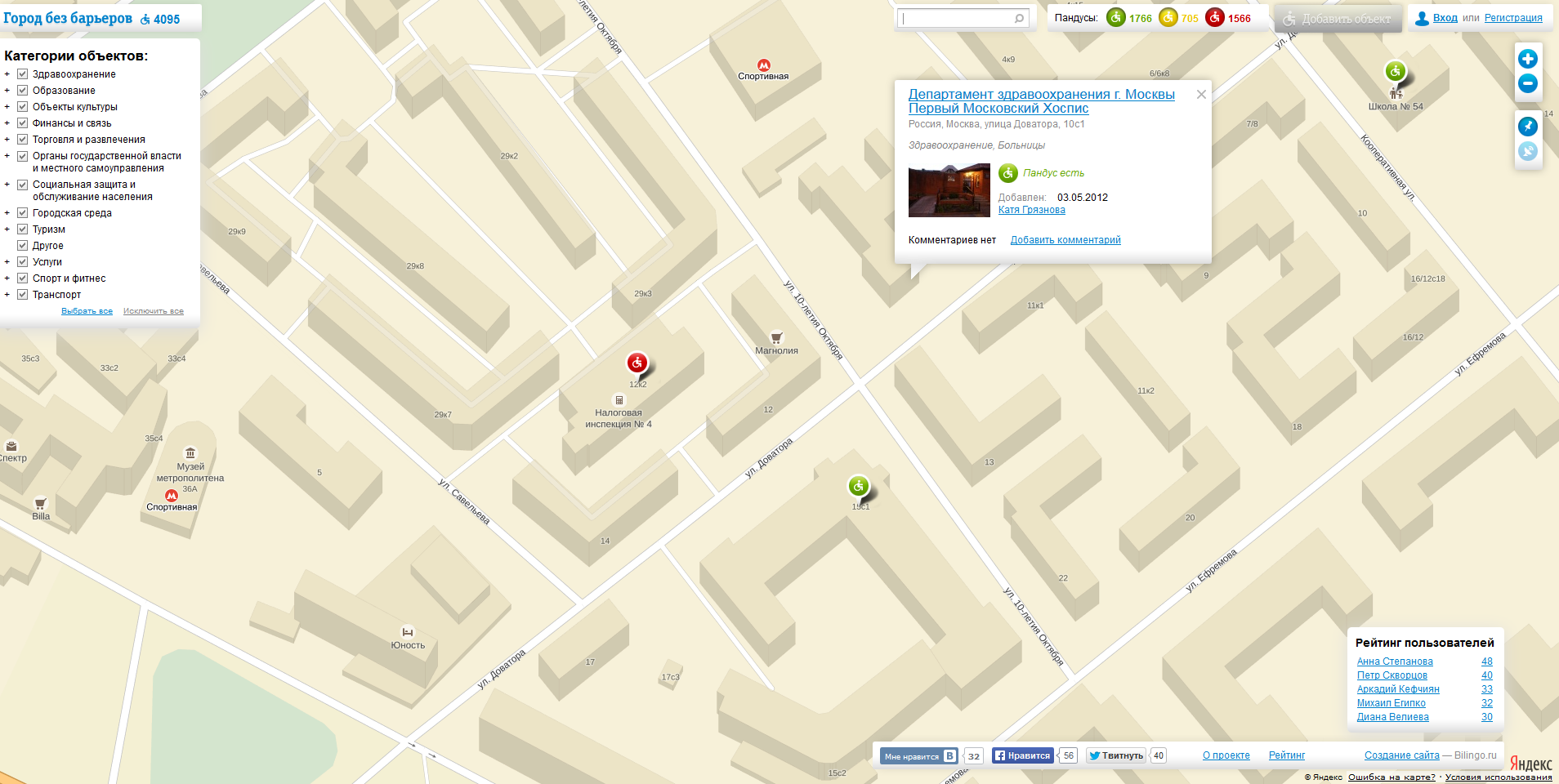
Недостатки:

* Небольшое количество объектов по регионам.
* Недостаточно подробное описание инфраструктуры объектов.
* Неоптимизированная работа с картой.

### 1.1.2 Социальный проект «Город без барьеров»

С октября 2011 года в сети Интернет стартовал проект «Город без барьеров». На сайте размещается альтернативная электронная карта Москвы. От других ее отличает социальная направленность, поскольку предназначена она для лиц с ограниченными возможностями здоровья.

На карте отмечены объекты, оборудованные пандусами, а так же проблемные участки, где пандусы отсутствуют и передвижение затруднено. Каждая отметка сопровождается описанием и фотоснимком пандуса.



**Рисунок 1.2 – Социальный проект «Город без барьеров»**

Основные преимущества:

* Бесплатный.
* Простой в управлении.
* Большое количество объектов.
* Возможность добавить свой объект.

Недостатки:

* Узкая направленность, только пандусы.
* Отсутствие описание инфраструктуры объектов.
* Информация об единственном городе.

## 1.2 Выбор программных средств

Выбор программных средств зависит от поставленной задачи и требованием к ней.

Главным требованием является то, что данная система должна быть web-приложением. Поэтому возникает необходимость в выборе языка программирования, который поддерживает такую возможность.

Учитывая вышеперечисленные требования, для разработки дипломного проекта были выбраны следующие программные средства и технологии:

- среда разработки – IntelliJ Idea;

- база данных – MySQL;

- языки программирования – Java, JavaScript;

- фреймворки – Spring, Hibernate;

- библиотека – jQuery.

### 1.2.1. Среда разработки IntelliJ Idea

IntelliJ IDEA — коммерческая [интегрированная среда разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) на многих языках программирования, в частности [Java](https://ru.wikipedia.org/wiki/Java), [JavaScript](https://ru.wikipedia.org/wiki/JavaScript), [Python](https://ru.wikipedia.org/wiki/Python), разработанная компанией [JetBrains](https://ru.wikipedia.org/wiki/JetBrains). Предоставляет условные лицензии в образовательных целях, а так же бесплатные версии с ограниченным функционалом.

Среда приобрела популярность благодаря широкому набору интегрированных инструментов для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3), которые позволяют программистам быстро реорганизовывать исходные тексты программ. Дизайн среды ориентирован на продуктивность работы программистов, позволяя им сконцентрироваться на разработке функциональности, в то время как IntelliJ IDEA берёт на себя выполнение рутинных операций.

IntelliJ IDEA предоставляет интегрированный инструментарий для разработки графического пользовательского интерфейса.

Среди прочих возможностей, IntelliJ IDEA хорошо совместима со многими популярными [свободными](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) инструментами разработчиков, такими как [CVS](https://ru.wikipedia.org/wiki/Concurrent_Versions_System), [Subversion](https://ru.wikipedia.org/wiki/Subversion), [Apache Ant](https://ru.wikipedia.org/wiki/Apache_Ant), [Maven](https://ru.wikipedia.org/wiki/Maven) и [JUnit](https://ru.wikipedia.org/wiki/JUnit).

Реализована поддержка [Java EE](https://ru.wikipedia.org/wiki/Java_EE), [UML](https://ru.wikipedia.org/wiki/UML)-диаграмм, подсчёт [покрытия кода](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BA%D1%80%D1%8B%D1%82%D0%B8%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B0), а также поддержка других систем управления версиями, языков и [фреймворков](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA).

### 1.2.2. Язык программирования Java

Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

### 1.2.3. Язык программирования JavaScript

JavaScript – это интерпретируемый, лёгкий, объектно-ориентированный, кроссплатформенный язык. Он более известен как язык скриптов для веб-страниц, но используется и в других программных продуктах.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам [приложений](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](http://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Основные архитектурные черты: [динамическая типизация](http://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), слабая типизация, [автоматическое управление памятью](http://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0), [прототипное программирование](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), функции как [объекты первого класса](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0).

JavaScript является [объектно-ориентированным](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) языком, но используемое в языке [прототипирование](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих [функциональным языкам](http://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – функции как [объекты первого класса](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0), объекты как списки, [анонимные функции](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8), [замыкания](http://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) – что придаёт языку дополнительную гибкость.

[Синтаксис](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81) языка JavaScript во многом напоминает синтаксис С и [Java](http://ru.wikipedia.org/wiki/Java), [семантически](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) же язык гораздо ближе к [Self](http://ru.wikipedia.org/w/index.php?title=Self&action=edit&redlink=1), [Smalltalk](http://ru.wikipedia.org/wiki/Smalltalk) или даже Lisp.

Структурно JavaScript можно представить в виде объединения трёх чётко различимых друг от друга частей:

1. ядро (ECMAScript),
2. объектная модель браузера ([Browser Object Model или BOM](http://ru.wikipedia.org/w/index.php?title=Browser_Object_Model&action=edit&redlink=1) ([de](http://de.wikipedia.org/wiki/Browser_Object_Model))),
3. объектная модель документа ([Document Object Model или DOM](http://ru.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model)).

Если рассматривать JavaScript в отличных от браузера окружениях, то объектная модель браузера и объектная модель документа могут не поддерживаться.

Объектную модель документа иногда рассматривают как отдельную от JavaScript сущность, что согласуется с определением DOM как независимого от языка интерфейса документа. В противоположность этому ряд авторов находят BOM и DOM тесно взаимосвязанными.

### 1.2.4 MySQL

MySQL — [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94). MySQL является решением для малых и средних приложений. Входит в состав серверов [WAMP](https://ru.wikipedia.org/wiki/WAMP), [AppServ](https://ru.wikipedia.org/w/index.php?title=AppServ&action=edit&redlink=1), [LAMP](https://ru.wikipedia.org/wiki/LAMP) и в портативные сборки серверов [Денвер](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BD%D0%B2%D0%B5%D1%80_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%29), [XAMPP](https://ru.wikipedia.org/wiki/XAMPP), [VertrigoServ](https://ru.wikipedia.org/w/index.php?title=VertrigoServ&action=edit&redlink=1). Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа [MyISAM](https://ru.wikipedia.org/wiki/MyISAM), поддерживающие полнотекстовый поиск, так и таблицы [InnoDB](https://ru.wikipedia.org/wiki/InnoDB), поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

### 1.2.5 MVC

Model-view-controller — схема использования нескольких [шаблонов проектирования](https://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), с помощью которых модель приложения, [пользовательский интерфейс](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения [архитектурного каркаса](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D0%BA%D0%B0%D1%81_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), когда переходят от теории к реализации в конкретной предметной области.



**Рисунок 1.3 – MVC**

В оригинальной концепции была описана сама идея и роль каждого из элементов: модели, представления и контроллера. Но связи между ними были описаны без конкретизации. Кроме того, различали две основные модификации:

1. Пассивная модель — модель не имеет никаких способов воздействовать на представление или контроллер, и используется ими в качестве источника данных для отображения. Все изменения модели отслеживаются контроллером и он же отвечает за перерисовку представления, если это необходимо.
2. Активная модель — модель оповещает представление о том, что в ней произошли изменения, а представления, которые заинтересованы в оповещении, подписываются на эти сообщения. Это позволяет сохранить независимость модели как от контроллера, так и от представления.

Классической реализацией концепции MVC принято считать версию именно с активной моделью.

Основная цель применения этой концепции состоит в разделении бизнес-логики (модели) от её визуализации (представления, вида). В частности, выполняются следующие задачи:

1. К одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели.
2. Не затрагивая реализацию видов, можно изменить реакции на действия пользователя, для этого достаточно использовать другой контроллер.
3. Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический интерфейс, либо разрабатывают бизнес-логику. Поэтому возможно добиться того, что программисты, занимающиеся разработкой бизнес-логики (модели), вообще не будут осведомлены о том, какое представление будет использоваться.

Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

1. Модель - предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
2. Представление - отвечает за отображение информации (визуализацию).
3. Контроллер - обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

### *1.2.6 Фреймворк Hibernate*

Hibernate – инструмент объектно-реляционного отображения (Object-Relational Mapping, ORM) данных для Java-окружения. Целью Hibernate является освобождение разработчика от большинства общих работ, связанных с задачами получения, сохранения данных в СУБД. Эта технология помогает удалить или инкапсулировать зависящий от поставщика SQL-код, а также решает стандартную задачу преобразования типов Java-данных в типы данных SQL и наборов данных из табличного представления в объекты Java-классов.

Целью Hibernate является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования по обеспечению хранения объектов в реляционной базе данных. Разработчик может использовать Hibernate как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей базой данных.

Hibernate не только решает задачу связи классов Java с таблицами базы данных (и типов данных Java с типами данных SQL), но также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL- и JDBC-кода. Hibernate автоматизирует генерацию SQL-запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос (портирование) приложения на любые базы данных SQL.

Mapping (сопоставление, проецирование) Java-классов с таблицами базы осуществляется с помощью конфигурационных XML файлов или Java-аннотаций. При использовании файла XML Hibernate может генерировать скелет исходного кода для классов длительного хранения (persistent). В этом нет необходимости, если используется аннотация. Hibernate может использовать файл XML или аннотации для поддержки схемы базы данных.

Обеспечиваются возможности по организации отношения между классами «один-ко-многим» и «многие-ко-многим». В дополнение к управлению связями между объектами, Hibernate также может управлять рефлексивными отношениями, где объект имеет связь «один-ко-многим» с другими экземплярами своего собственного типа данных.

## 1.3. Выводы по главе 1

Аналитический обзор существующих решений выявил тот факт, что недостатками подобных приложений зачастую является информационная составляющая, выбор критериев поиска. Кроме того, отсутствуют региональные аналоги, позволяющие получить необходимую информацию в г. Гродно и области.

После сделанного анализа и выбора технологий разработки было принято решение акцентировать внимание на недостатках существующих решений и разработать подобную систему для г. Гродно. Выбранный набор технологий позволяет эффективно реализовать требуемую систему с предоставлением необходимого функционала. Разработка такого программного продукта является целью данной дипломной работы.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ приложения

# **ГЛАВА 3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ**

# **ЗАКЛЮЧЕНИЕ**

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ