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».

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

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.

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

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 4](#_Toc390947265)

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 6](#_Toc390947266)

[ВВЕДЕНИЕ 7](#_Toc390947267)

[ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ И ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ 9](#_Toc390947268)

[1.1 Обзор существующих решений 9](#_Toc390947269)

[1.1.1 Проект «Доступная жизнь» 9](#_Toc390947270)

[1.1.2 Социальный проект «Город без барьеров» 10](#_Toc390947271)

[1.1.3 «Барьеров нет» 12](#_Toc390947272)

[1.2 Выбор программных средств 13](#_Toc390947273)

[1.2.1. Среда разработки Eclipse PDT 13](#_Toc390947274)

[1.2.2. Язык программирования PHP 16](#_Toc390947275)

[1.2.3. Язык программирования JavaScript 18](#_Toc390947276)

[1.3. Выводы по главе 1 20](#_Toc390947277)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ приложения 21](#_Toc390947278)

[2.1. Основные требования к приложению 21](#_Toc390947279)

[2.2. Общие подходы к реализации системы 24](#_Toc390947280)

[2.3. Модель данных 26](#_Toc390947281)

[2.4. Диаграмма вариантов использования 29](#_Toc390947282)

[2.5. Проектирование интерфейса 30](#_Toc390947283)

[2.5. Выводы 31](#_Toc390947284)

[ГЛАВА 3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ 32](#_Toc390947285)

[3.1 Папка CSS 32](#_Toc390947286)

[3.2 Папка js 33](#_Toc390947287)

[3.3 Папка php 33](#_Toc390947288)

[3.4 Папка lib 35](#_Toc390947289)

[3.5 Папка resources 36](#_Toc390947290)

[3.3 Выводы по главе 3 36](#_Toc390947291)

[ЗАКЛЮЧЕНИЕ 37](#_Toc390947292)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 38](#_Toc390947293)

[ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 40](#_Toc390947294)

# 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.

# ВВЕДЕНИЕ

На сегодняшний день в современном обществе большую роль играет время. Люди стараются распланировать свой рабочий день, поездку, встречу, а также дальнейший ход принятия решений, тем самым стараясь оградить себя от каких либо неприятностей и лишней утечки времени.

Чтобы не затрачивать свое время на поиск местонахождения нужных объектов в городе существует множество приложений, способных, по нажатию нескольких кнопок, вывести нам нужную, и что самое важное, актуальную информацию мест с описанием их инфраструктуры, расположением и другой полезной информацией. Изобразив все это на карте и нарисовав маршрут. Зачастую вопрос состоит в выборе, а не в поиске как провести свой отдых, и в какое место сходить в свободное время. Данная тема актуальна и не только для отдыха, но и для любого вида деятельности. Можно легко выбрать и найти нужную гостиницу, аптеку, или даже магазин, или автомастерскую.

Имея широкие возможности, у человека не встает вопроса больше в недостачи информации. Как и в любые времена, существуют люди, которые имеют ограниченные возможности.

Данная часть населения имеет аналогичную потребность, но имея ограниченность в своих возможностях, нуждаются в более подробном описании инфраструктуры того или иного объекта.

Данная система представляет собой специальную программу, содержащую базу данных организаций, предприятий и учреждений города с указанием наличия безбарьерной среды и описанием ее основных параметров. Подобный путеводитель необходим для улучшения качества жизни людей с ограниченными возможностями и облегчения их социальной адаптации.

Итак, сформулируем цель и задачи дипломной работы.

Цель работы – создание информационного web-сервиса, представляющего данные об элементах без барьерной среды организаций г.Гродно.

Для достижения поставленной цели необходимо решить следующие задачи:

* Провести обзор существующих приложений для информирования лиц с ограниченными возможностями
* Рассмотреть существующие фреймворки для работы с картографическим контентом и определить среду реализации
* Сформулировать требования к сервису и условиям его эксплуатации
* Спроектировать сервис представления данных об элементах безбарьерной среды организаций г.Гродно
* Реализовать программные компоненты сервиса, сформировать базу данных сервиса
* Провести тестирование работы приложения
* Внедрить разработку

Направление работы актуально и имеет акцентированную социальную значимость. Работа выполнялась по заказу отдела социальной защиты Гродненского городского исполнительного комитета. Заказчиком была предоставлена информация об элементах безбарьерной среды г.Гродно, к которым относятся инженерные, архитектурные и дизайнерские решения, позволяющие людям с ограниченными возможностями передвижения (инвалиды-колясочники, инвалиды по зрению) получать информацию о возможностях передвижения по определенным объектам городской инфраструктуры. К настоящему времени аналогичных информационных сервисов в других, кроме г.Гродно, городах Республики Беларусь нет.

# ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ И ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

# 1.1 Обзор существующих решений

Существует большое количество систем с прорисовкой маршрута, показом объектов и их описанием, 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»

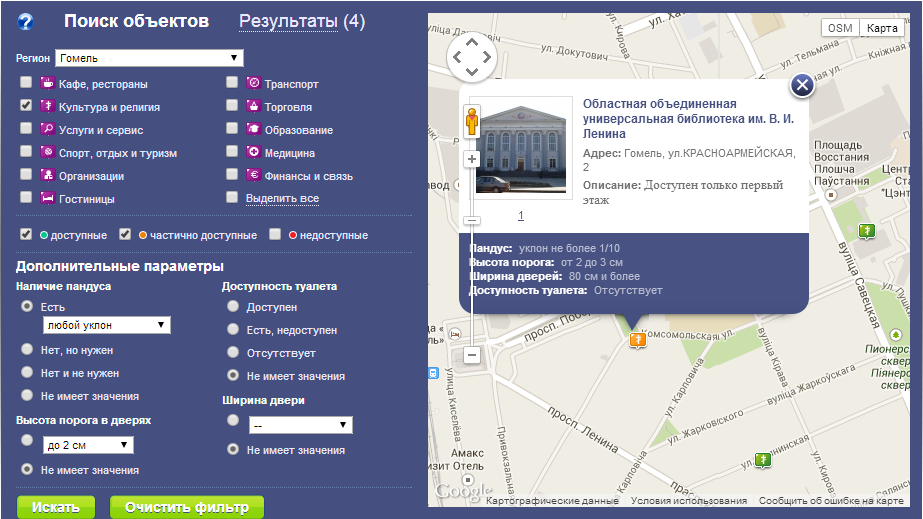
Сайт [www.freeway.by](http://www.freeway.by/) – **Жизнь без барьеров** является результатом реализации проекта «Доступная жизнь» (см. рисунок 1.1), который был реализован в 2010 году при поддержке отдела информации, образования и культуры посольства США в Беларуси.

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

Также данный сайт получил свою жизнь благодаря тесному сотрудничеству трёх организаций:

* Мозырской межрайонной организации общественного объединения "Республиканская ассоциация инвалидов-колясочников";
* Общественного объединения "Республиканская ассоциация инвалидов-колясочников";
* РУП Информационного центра земельно-кадастровых данных и мониторинга земель".

Спроектировала и реализовала этот проект в Интернете компании Astronim.



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

Основные преимущества:

* Бесплатный.
* Простой в управлении.

Недостатки:

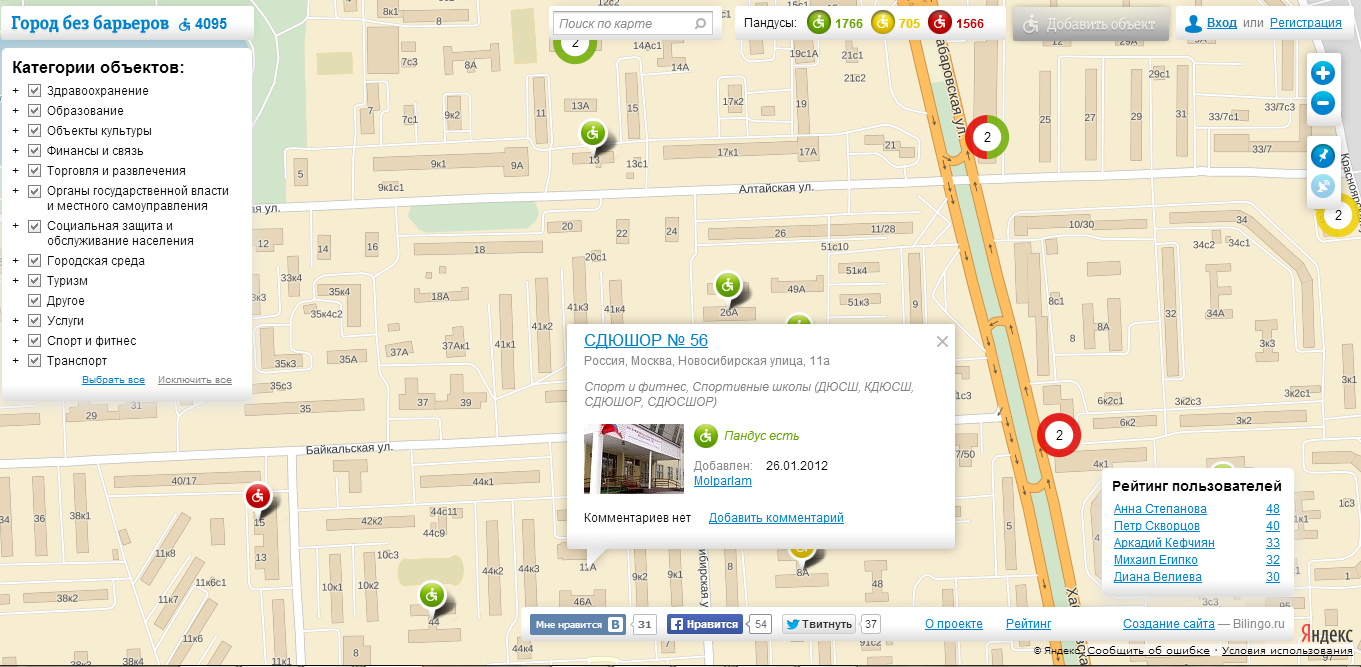
* Небольшое количество объектов по регионам.
* Недостаточно подробное описание инфраструктуры объектов.

## 1.1.2 Социальный проект «[Город без барьеров](http://pandus.molparlam.ru/)»

С октября 2011 года в сети Интернет стартовал проект «Город без барьеров» (см. рисунок 1.2). На сайте pandus.molparlam.ru 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.

В настоящий момент база данных объектов по всем районам пополняется за счет молодых активистов программы «Молодежный кадровый резерв».

****

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

Основные преимущества:

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

Недостатки:

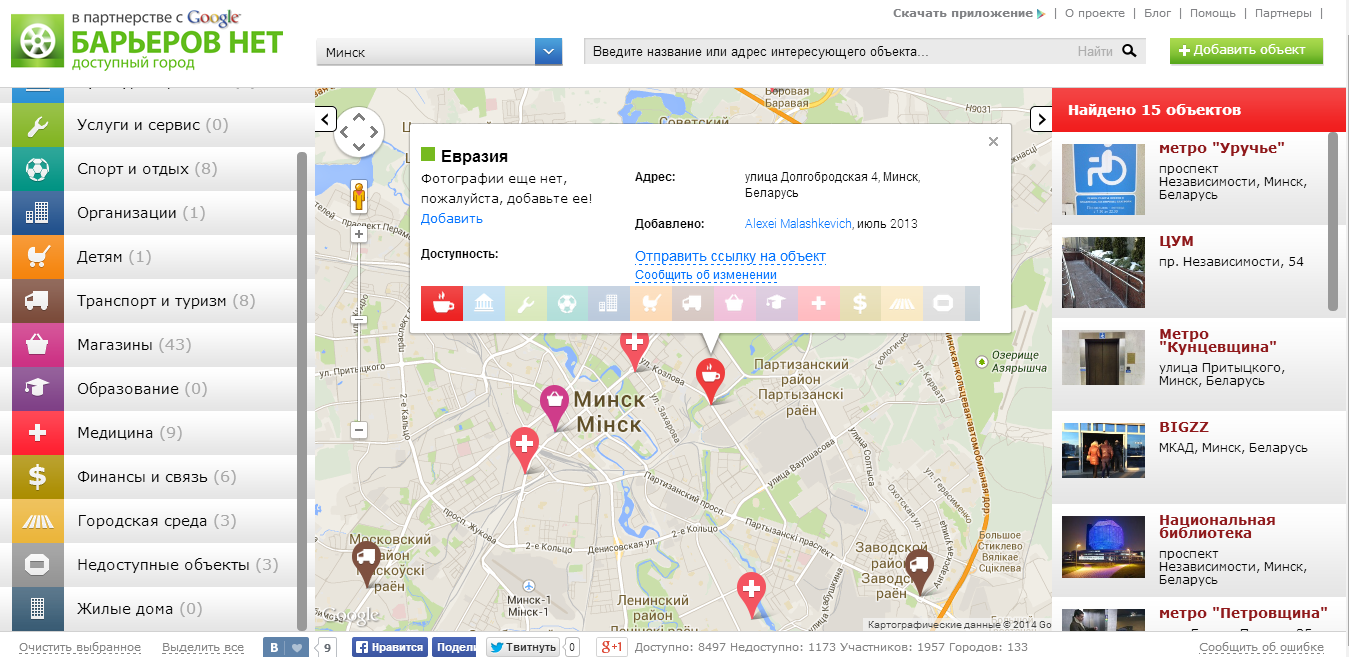
* узкая направленность, только пандусы.
* Отсутствие описание инфраструктуры объектов.

## 1.1.3 «Барьеров нет»

Проект "Доступный город – БАРЬЕРОВ.НЕТ" (см. рисунок 1.3) реализуется группой энтузиастов в партнёрстве:

* фонд ["Без Барьеров"](http://barrier-free.ru/);
* [Google](https://www.google.ru/intl/ru/about/);
* [Ottobock](http://www.ottobock.ru/).

**Цель проекта –** Дать возможность всем, кто пользуется инвалидными и детскими колясками, передвигаться по городу.

**Рисунок 1.3 – «Барьеров нет»**

Основные преимущества:

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

Недостатки:

* Небольшое количество объектов по некоторым регионам.
* Недостаточное описание инфраструктуры объектов.

# 1.2 Выбор программных средств

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

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

Во-вторых, БД будет находиться на хостинге phpMyAdmin.

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

- среда разработки – Eclipse PDT;

- база данных – MySQL;

- языки программирования – PHP, JavaScript;

- библиотека – jQuery**.**

### 1.2.1. Среда разработки Eclipse PDT

Eclipse – это расширяемая среда разработки (Integrated Development Environment, далее IDE) с открытым исходным кодом (далее OpenSource). Проект был запущен в ноябре 2001 года, когда IBM передала исходный код своего Websphere Studio Workbench стоимостью 40 миллионов долларов на OpenSource и сформировала консорциум Eclipse для управления продолжающейся разработкой.

Наиболее известные приложения на основе [Eclipse Platform](http://ru.wikipedia.org/wiki/Eclipse_Platform" \o "Eclipse Platform) — различные «Eclipse [IDE](http://ru.wikipedia.org/wiki/%D0%A1%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_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)» для разработки ПО на множестве языков (например, наиболее популярный «[Java](http://ru.wikipedia.org/wiki/Java" \o "Java) IDE», поддерживавшийся изначально, не полагается на какие-либо закрытые расширения, использует стандартный открытый [API](http://ru.wikipedia.org/wiki/API) для доступа к Eclipse Platform).

Цель создания Eclipse была сформулирована следующим образом: "разработать богатую, полнофункциональную индустриальную платформу коммерческого качества для разработки сильно-интегрированных инструментов". Для достижения этой цели консорциум нацелен на три главных проекта:

1. Проект [*The Eclipse*](http://www.eclipse.org/eclipse/index.html) непосредственно Eclipse IDE ("платформы", содержащей и исполняющей инструменты Eclipse), инструментов разработки для Java (Java Development Tools, далее JDT) и среды разработки Plug-In (Plug-In Development Environment, далее PDE), позволяющих расширять платформу.

2. Проект [*Eclipse Tools*](http://www.eclipse.org/tools/index.html) имеет своей целью создание инструментов для платформы Eclipse (в текущей разработке находятся подпроекты создания IDE для Cobol, IDE для C/C++, а также инструмента для построения EMF моделей).

3. Проект [*The Eclipse Technology*](file:///D:\Dropbox\The%20Eclipse%20Technology) ответственен за технологические разработки, эволюцию и обучение использованию платформы Eclipse.

Платформа Eclipse в сочетании с JDT включает многие из возможностей, которые включаются в коммерческие IDE: редактор с подсветкой синтаксиса, инкрементальная компиляция кода, потокобезопасный отладчик, навигатор по классам, менеджеры файлов и проектов, а также интерфейсы к стандартным системам контроля исходных текстов, таким как CVS и ClearCase.

Проект Eclipse представляет собой первую столь мощно поддержанную мировым IT-сообществом попытку создания единой открытой интегрированной платформы разработки приложений, обладающей надежностью, функциональностью и уровнем качества коммерческого продукта. Фактически эта платформа предназначена для всего и ни для чего конкретно: она представляет собой основу, имеющую блочную структуру и интегрирующую инструменты разработки ПО различных производителей для создания приложений на любом языке, с использованием любых технологий и для любой программной платформы. Вокруг проекта Eclipse в настоящее время сформировано сообщество крупнейших IT-компаний, среди которых Borland, IBM, SAP AG, RedHat и другие.

Также Eclipse предлагает множество уникальных возможностей, например [рефакторинг кода](http://www.refactoring.com/), автоматическое обновление/установка кода (с помощью Менеджера Обновлений), список текущих задач, отладку модулей с помощью [JUnit](http://www.junit.org/)и интеграцию с инструментом компоновки Jakarta [Ant](http://jakarta.apache.org/ant/index.html)*.*

Несмотря на большое число стандартных возможностей, Eclipse отличается от традиционных IDE по ряду особенностей. Наверное, самое интересное в Eclipse то, что она *полностью* независима от платформы и языка. Помимо языков, поддерживаемых консорциумом в настоящий момент (Java, Cobol, C/C++), ведутся разработки по добавлению в Eclipse поддержки таких языков, как Python, Eiffel, PHP, Ruby, и C#.

Консорциум предоставляет готовые исполняемые файлы для Windows, Linux, Solaris, HP-UX, AIX, QNX и Mac OS X. Большой интерес в Eclipse представляет plug-in архитектура, а также богатый API, предоставляемый PDE, позволяющий расширять Eclipse. Добавление поддержки для нового редактора, представления или языка программирования является достаточно простым, благодаря грамотно разработанным API и большим строительным блокам, предоставляемым Eclipse.

Я в своем проекте буду использовать среду разработки Eclipse PDT.

Eclipse PDT — Free OpenSource среда разработки для PHP.

 Плагин PDT превращает Eclipse в полноценную среду для разработки проектов на PHP. Возможности плагина:

* Возможность легкой интеграции Zend Debugger или XDebug (все сводится, в основном, к установке плагина штатными средствами Eclipse)
* Поддержка сворачивания блоков кода
* Поддержка автозавершения набираемых строк
* Поддержка шаблонов PHPDoc
* Поддержка аннотаций
* Поддержка выделения синтаксиса цветом
* Базовый рефакторинг (!)
* Интегрированный редактор HTML, CSS и Javascript
* Поддержка отладки, брейкпоинтов

Похоже, что это повод забыть про Zend Developer Studio, хотя Zend, кажется, приветствует выпуск PDT. Среда на самом деле весьма удобна и проста в обращении (в отличие, скажем, от JBuilder 2007, который построен на Eclipse)

Учитывая сотни проектов по разработке plug-in, ведущихся в настоящее время, таких индустриальных гигантов как IBM, HP и Rational, предоставляющих ресурсы, а также проектных тяжеловесов вроде Erich Gamma, помогающих направлять процесс эволюции в нужное русло – у Eclipse большое будущее.

### 1.2.2. Язык программирования PHP

PHP([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) PHP: Hypertext Preprocessor – «PHP: [препроцессор](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) [гипертекста](http://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82)»; первоначально Personal Home Page Tools – «Инструменты для создания персональных веб-страниц») – [скриптовый язык](http://ru.wikipedia.org/wiki/%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) программирования, созданный для генерации [HTML](http://ru.wikipedia.org/wiki/HTML)-страниц на [веб-сервере](http://ru.wikipedia.org/wiki/%D0%B2%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и работы с [базами данных](http://ru.wikipedia.org/wiki/%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). На сегодняшний момент поддерживается подавляющим большинством представителей [хостингов](http://ru.wikipedia.org/wiki/%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3). Входит в [LAMP](http://ru.wikipedia.org/wiki/LAMP) – «стандартный» набор для создания [вебсайтов](http://ru.wikipedia.org/wiki/%D0%B2%D0%B5%D0%B1%D1%81%D0%B0%D0%B9%D1%82).

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с [открытым кодом](http://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%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). Проект распространяется под [собственной лицензией](http://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_PHP), несовместимой с [GNU GPL](http://ru.wikipedia.org/wiki/GNU_GPL).

В области [программирования для сети Интернет](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%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), PHP – один из популярных [сценарных языков](http://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) (наряду с [JSP](http://ru.wikipedia.org/wiki/JSP), [Perl](http://ru.wikipedia.org/wiki/Perl) и языками, используемыми в [ASP.NET](http://ru.wikipedia.org/wiki/ASP.NET)) благодаря своей простоте, скорости выполнения, богатой функциональности, [кроссплатформенности](http://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) и распространению исходных кодов на основе [лицензии PHP](http://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_PHP).

Популярность в области построения [веб-сайтов](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

* автоматическое извлечение [POST](http://ru.wikipedia.org/wiki/HTTP#POST) и [GET](http://ru.wikipedia.org/wiki/HTTP#GET)-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
* взаимодействие с большим количеством различных систем управления базами данных ([MySQL](http://ru.wikipedia.org/wiki/MySQL), [MySQLi](http://ru.wikipedia.org/wiki/MySQLi), [SQLite](http://ru.wikipedia.org/wiki/SQLite), [PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL), [Oracle (OCI8](http://ru.wikipedia.org/w/index.php?title=Oracle_(OCI8&action=edit&redlink=1)), [Oracle](http://ru.wikipedia.org/wiki/Oracle_(%D0%A1%D0%A3%D0%91%D0%94)), [Microsoft SQL Server](http://ru.wikipedia.org/wiki/Microsoft_SQL_Server), [Sybase](http://ru.wikipedia.org/wiki/Sybase), [ODBC](http://ru.wikipedia.org/wiki/ODBC), [mSQL](http://ru.wikipedia.org/wiki/MSQL), [IBM DB2](http://ru.wikipedia.org/wiki/IBM_DB2), [Cloudscape](http://ru.wikipedia.org/wiki/Cloudscape) и [Apache Derby](http://ru.wikipedia.org/wiki/Apache_Derby), [Informix](http://ru.wikipedia.org/wiki/Informix), [Ovrimos SQL](http://ru.wikipedia.org/w/index.php?title=Ovrimos_SQL&action=edit&redlink=1), [Lotus Notes](http://ru.wikipedia.org/wiki/Lotus_Notes), [DB++](http://ru.wikipedia.org/w/index.php?title=DB%2B%2B&action=edit&redlink=1), [DBM](http://ru.wikipedia.org/wiki/DBM), [dBase](http://ru.wikipedia.org/wiki/DBase), [DBX](http://ru.wikipedia.org/wiki/DBX), [FrontBase](http://ru.wikipedia.org/w/index.php?title=FrontBase&action=edit&redlink=1), [FilePro](http://ru.wikipedia.org/w/index.php?title=FilePro&action=edit&redlink=1), [Ingres II](http://ru.wikipedia.org/w/index.php?title=Ingres_II&action=edit&redlink=1), [SESAM](http://ru.wikipedia.org/w/index.php?title=SESAM&action=edit&redlink=1),[Firebird](http://ru.wikipedia.org/wiki/Firebird) / [InterBase](http://ru.wikipedia.org/wiki/InterBase), [Paradox File Access](http://ru.wikipedia.org/w/index.php?title=Paradox_File_Access&action=edit&redlink=1), [MaxDB](http://ru.wikipedia.org/w/index.php?title=MaxDB&action=edit&redlink=1), [Интерфейс](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) [PDO](http://ru.wikipedia.org/wiki/PDO));
* автоматизированная отправка [HTTP-заголовков](http://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BA%D0%B8_HTTP);
* работа с HTTP-авторизацией;
* работа с [cookies](http://ru.wikipedia.org/wiki/Cookies) и сессиями;
* работа с локальными и удалёнными файлами, [сокетами](http://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BA%D0%B5%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81));
* обработка файлов, загружаемых на сервер;
* работа с [XForms](http://ru.wikipedia.org/wiki/XForms).

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу корпорации TIOBE, базирующемся на данных поисковых систем, в июне 2013 года PHP находился на 5 месте среди языков программирования. К крупнейшим сайтам, использующим PHP, относятся [Facebook](http://ru.wikipedia.org/wiki/Facebook),[Wikipedia](http://ru.wikipedia.org/wiki/Wikipedia) и др.

Входит в [LAMP](http://ru.wikipedia.org/wiki/LAMP) — распространённый набор программного обеспечения для создания и [хостинга](http://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) [веб-сайтов](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) ([Linux](http://ru.wikipedia.org/wiki/Linux" \o "Linux), [Apache](http://ru.wikipedia.org/wiki/Apache), [MySQL](http://ru.wikipedia.org/wiki/MySQL), PHP).

### 1.2.3. Язык программирования JavaScript

**JavaScript** (иногда просто JS) – это интерпретируемый, лёгкий, объектно-ориентированный, кроссплатформенный язык. Он более известен как язык скриптов для веб-страниц, но используется и в других программных продуктах, например [node.js](http://nodejs.org/) или [Apache CouchDB.](http://couchdb.apache.org/)

JavaScript основывается на [ECMAScript](https://developer.mozilla.org/ru/docs/JavaScript/Language_Resources). На настоящий 2014 год, все современные браузеры поддерживают ECMAScript 5.1. Старые версии браузеров поддерживают по крайней мере ECMAScript 3. В данный момент ведется работа над новой, 6-ой версией стандарта (ECMA Harmony).

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).

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования [непрограммистами](http://ru.wiktionary.org/wiki/%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C-%D0%BD%D0%B5%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82). Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным [товарным знаком](http://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%B2%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D0%B7%D0%BD%D0%B0%D0%BA) компании [Oracle Corporation](http://ru.wikipedia.org/wiki/Oracle_Corporation).

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, [карринг](http://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%80%D0%B8%D0%BD%D0%B3), [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.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком [Си](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%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%D1%8F)) имеет коренные отличия:

* [объекты](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%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)), с возможностью [интроспекции](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D1%80%D0%BE%D1%81%D0%BF%D0%B5%D0%BA%D1%86%D0%B8%D1%8F_(%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);
* автоматическое [приведение типов](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D1%82%D0%B8%D0%BF%D0%BE%D0%B2);
* автоматическая [сборка мусора](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);
* [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).

В языке отсутствуют такие полезные вещи, как:

* [модульная система](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%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)): JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
* [стандартная библиотека](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0): в частности, отсутствует [интерфейс программирования приложений](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%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%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для [бинарных данных](http://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB);
* стандартные [интерфейсы](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) к [веб-серверам](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и [базам данных](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85);
* [система управления пакетами](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82%D0%B0%D0%BC%D0%B8), которая бы отслеживала зависимости и автоматически устанавливала их.

[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 Си 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 [Лиспу](http://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%81%D0%BF).

В JavaScript:

* все [идентификаторы](http://ru.wikipedia.org/wiki/%D0%98%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80) регистрозависимы,
* в названиях переменных можно использовать [буквы](http://ru.wikipedia.org/wiki/%D0%91%D1%83%D0%BA%D0%B2%D0%B0), [подчёркивание](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%87%D1%91%D1%80%D0%BA%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [символ доллара](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB_%D0%B4%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B0), [арабские цифры](http://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D1%8B),
* названия переменных не могут начинаться с цифры,
* для оформления однострочных [комментариев](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_(%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)) используются //, многострочные и внутристрочные комментарии начинаются с /\* и заканчиваются \*/.

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

* ядро (ECMAScript),
* 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))),
* 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.

## 1.3. Выводы по главе 1

Аналитический обзор существующих решений выявил тот факт, что недостатками подобных приложений зачастую является информационная составляющая, а так же выбор критериев поиска.

После сделанного анализа и выбора технологий разработки было принято решение принять во внимание данные недостатки существующих систем и разработать подобную систему для г. Гродно. Разработка такого программного обеспечения является целью данной дипломной работы.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ приложения

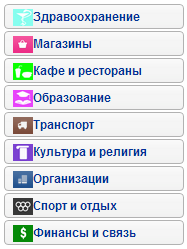
## 2.1. Основные требования к приложению

Исходя из поставленной цели, а так же рассмотренных ранее решений и их реализаций было принято решение спроектировать и разработать автоматизированную информационную систему, которая давала бы возможность получения актуальной информации для выбранного объекта.

Основные возможности, которые должны поддерживаться системой:

1. Наличие меню выбора категории (см. рисунок 2.1);

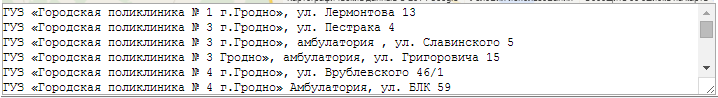
Названия категорий должны браться из БД.



**Рисунок 2.1 ­ Меню «Категории»**

1. Наличие поля для вывода информации об объектах (см. рисунок 2.2);

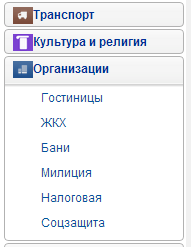
Название и адрес объектов выводить в информационном поле.



**Рисунок 2.2 ­ Информационное поле краткого описания объектов**

1. Наличие списка подкатегорий (см. рисунок 2.3)

При нажатии на одну из категорий, внизу показывать список подкатегорий, относящийся к данной подкатегории.



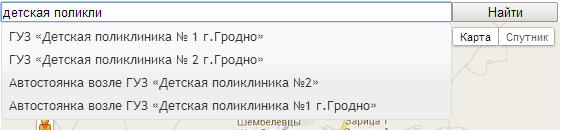
**Рисунок 2.3 ­ Список подкатегорий**

1. Поиск объектов по выбранной подкатегории;

После нажатия на выбранную подкатегорию, на карте отображать объекты данной подкатегории. В поле внизу выводить краткую информацию об объектах.

1. Поиск нужного объекта по названию (см. рисунок 2.4);

При вводе названия объекта, в текстовом поле, показывать подсказки, в виде данных взятых из базы данных.



**Рисунок 2.4 ­ Поиск по названию объекта**

1. Наглядное отображение местонахождения объектов на карте;

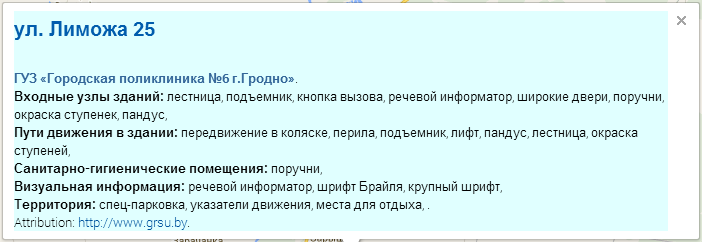
Все объекты должны отображаться на карте, в виде маркеров (см. рисунок 2.5).



**Рисунок 2.5 ­ Маркер**

1. Информационное окно описания объекта;

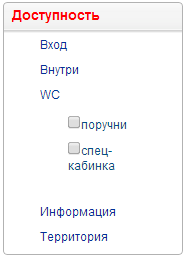
При клике на маркер, открывается новое окно с описанием объекта (см. рисунок 2.6).



**Рисунок 2.6 ­ Информационное окно с описанием объекта**

1. Наличие критериев поиска.

Внизу, в виде выпадающего меню, организовать критерии поиска объектов (см. рисунок 2.7).



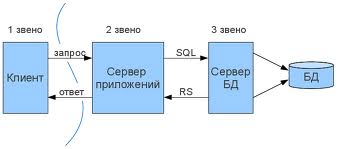
**Рисунок 2.7 ­ Меню выбора «Критериев поиска»**

## 2.2. Общие подходы к реализации системы

Учитывая предъявляемые требования к системе для обработки данных, связанных с организацией путеводителе для лиц с ограниченными возможностями, необходимо разработать общие подходы к реализации данной системы. Во-первых, необходимо узнать потребности и интересы людей с ограниченными возможностями, а также. Во-вторых – разработать архитектуру, наиболее подходящую для реализации всех возможностей системы.

Общая архитектура.

Проанализировав требования, предъявляемые к вариантам использования данной системы, была выбрана классическая 3-х уровневая архитектура, представленная на рисунке 2.8.

****

**Рисунок 2.8 ­ Клиент серверная модель**

Трёхуровневая архитектура — архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиентского приложения, сервера приложений, к которому подключено клиентское приложение, и сервера [базы данных](http://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), с которым работает сервер приложений.

[Клиент](http://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) — это интерфейсный компонент, который представляет первый уровень, собственно приложение для конечного пользователя. Первый уровень не должен иметь прямых связей с базой данных (по требованиям безопасности), быть нагруженным основной [бизнес-логикой](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) (по требованиям [масштабируемости](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C)) и хранить [состояние приложения](http://ru.wikipedia.org/wiki/ASP.NET_state_management) (по требованиям надежности). На первый уровень может быть вынесена и обычно выносится простейшая бизнес-логика: интерфейс [авторизации](http://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции (сортировка, группировка, подсчет значений) с данными, уже загруженными на терминал.

[Сервер приложений](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) располагается на втором уровне. На втором уровне сосредоточена бо́льшая часть бизнес-логики. Вне его остаются фрагменты, экспортируемые на терминалы, а также погруженные в третий уровень хранимые процедуры и триггеры.

Сервер базы данных обеспечивает хранение данных и выносится на третий уровень. Обычно это стандартная [реляционная](http://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) или [объектно-ориентированная](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%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) [СУБД](http://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94). Если третий уровень представляет собой базу данных вместе с [хранимыми процедурами](http://ru.wikipedia.org/wiki/%D0%A5%D1%80%D0%B0%D0%BD%D0%B8%D0%BC%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0), [триггерами](http://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D0%B3%D0%B3%D0%B5%D1%80_(%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)) и схемой, описывающей приложение в терминах реляционной модели, то второй уровень строится как программный интерфейс, связывающий клиентские компоненты с прикладной логикой базы данных.

В простейшей конфигурации физически сервер приложений может быть совмещён с сервером базы данных на одном компьютере, к которому по сети подключается один или несколько терминалов.

Достоинства:

- [масштабируемость](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C).

- конфигурируемость (изолированность уровней друг от друга позволяет (при правильном развертывании архитектуры) быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней).

- высокая безопасность.

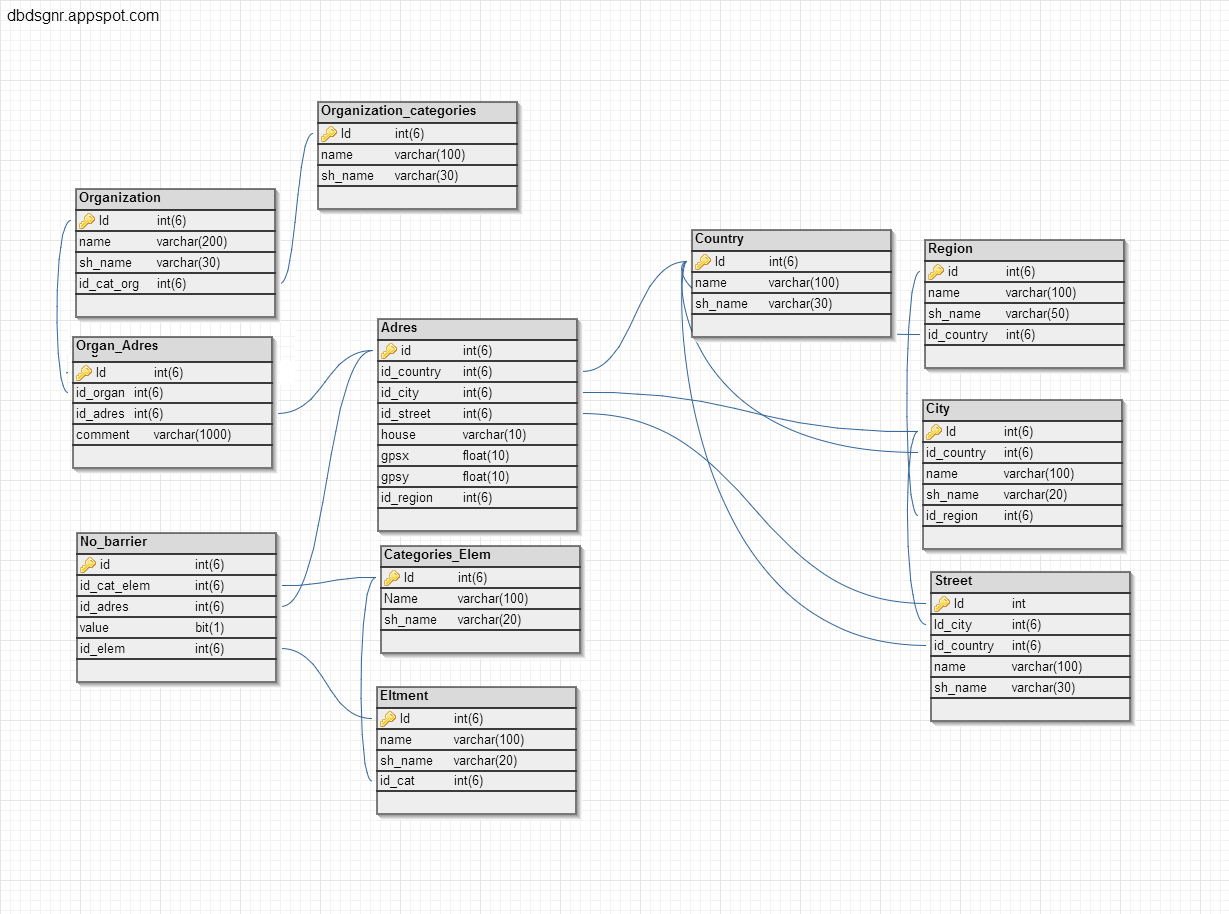
- высокая надёжность.

- низкие требования к скорости канала (сети) между терминалами и сервером приложений.

- низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости. Терминалом может выступать не только компьютер, но и, например, мобильный телефон.

## 2.3. Модель данных

С учетом анализа предметной области, выделены сущности. Средствами Sybase Power Designer была построена концептуальная модель (рис. 2.9).



**Рисунок 2.9 – Концептуальная модель**

**Описание таблиц БД «nobarrier\_grsu\_by»**

**Таблица 2.1 – Categories\_Elem**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(20) |  |

**Таблица 2.2 – Adres**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| id\_country | int(6) | Вторичный ключ |
| id\_city | int(6) | Вторичный ключ |
| id\_street | int(6) | Вторичный ключ |
| id\_region | int(6) | Вторичный ключ |
| House | varchar(20) |  |
| Gpsx | decimal(12,8) |  |
| Gpsy | decimal(12,8) |  |

**Таблица 2.3 – Country**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(30) |  |

**Таблица 2.4 – Element**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(30) |  |
| id\_cat | int(6) | Вторичный ключ |

**Таблица 2.5 – Region**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(50) |  |
| id\_country | int(6) | Вторичный ключ |

**Таблица 2.6 – City**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(20) |  |
| id\_country | int(6) | Вторичный ключ |
| id\_region | int(6) | Вторичный ключ |

**Таблица 2.7 – Organization**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(200) |  |
| sh\_name | varchar(200) |  |
| id\_cat\_org | int(6) | Вторичный ключ |
| id\_subcat | int(6) | Вторичный ключ |
| Actual | int(1) | Вторичный ключ |

**Таблица 2.8 – Organization\_categories**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(30) |  |
| link | varchar(150) |  |
| Color | varchar(12) |  |

**Таблица 2.9 – Organ\_Adres**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(11) | Первичный ключ |
| id\_organ | int(6) | Вторичный ключ |
| id\_adres | int(6) | Вторичный ключ |

**Таблица 2.10 – No\_barrier**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| id\_cat\_elem | int(6) | Вторичный ключ |
| id\_organ\_adres | int(6) | Вторичный ключ |
| id\_organ | int(6) | Вторичный ключ |
| id\_adres | int(6) | Вторичный ключ |
| id\_elem | int(6) | Вторичный ключ |
| Value | int(1) |  |
| Actual | int(1) |  |

**Таблица 2.11 – Street**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(100) |  |
| sh\_name | varchar(30) |  |
| id\_country | int(6) | Вторичный ключ |
| id\_region | int(6) | Вторичный ключ |
| id\_city | int(6) | Вторичный ключ |

**Таблица 2.12 – Subcategories**

|  |  |  |
| --- | --- | --- |
| **Идентификатор** | **Тип** | **Комментарий** |
| Id | int(6) | Первичный ключ |
| Name | varchar(50) |  |
| sh\_name | varchar(30) |  |
| id\_cat | int(6) | Вторичный ключ |

## 2.4. Диаграмма вариантов использования

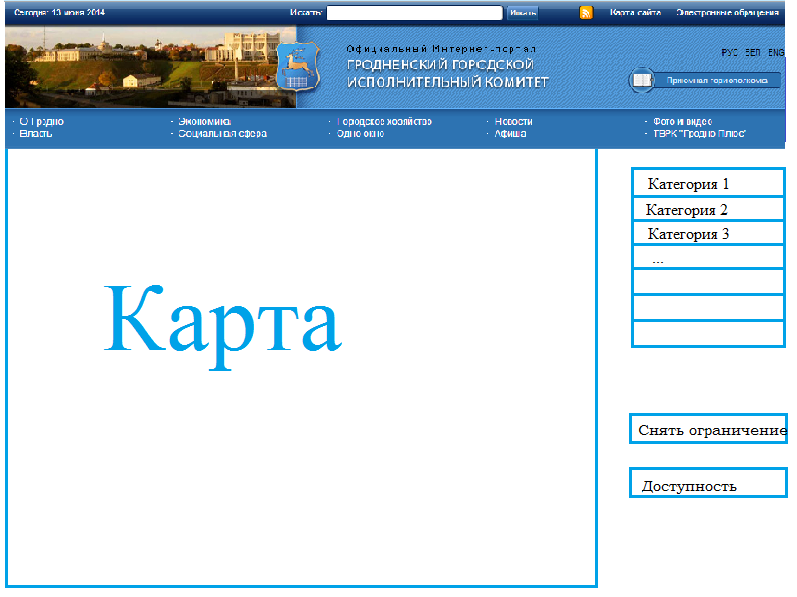
Рассмотрим функциональное назначение системы с помощью диаграммы вариантов использования (см рисунок 2.11). Данная диаграмма разработана в контексте использования системы в целом.

****

**Рисунок 2.10 – Диаграмма вариантов использования**

## 2.5. Проектирование интерфейса

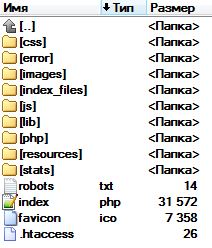
Для реализации приложения было принято решения использовать интерфейс сайта Гродненского городского исполнительного комитета (см. рисунок 2.10).

 **Рисунок 2.11 – Интерфейс пользователя.**

## 2.5. Выводы

# **ГЛАВА 3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ**

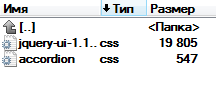
На рис. 3.1 изображены папки проекта, входящие в информационный сервис для лиц с ограниченными возможностями.



**Рисунок 3.1 ­ Структура папок проекта**

## 3.1 Папка CSS

В папке css содержатся файлы с расширением css, которые отвечают за описания внешнего вида документа (см. рисунок 3.2). В этих файлам мы описываем, как будет выглядеть тот или иной элемент страницы.



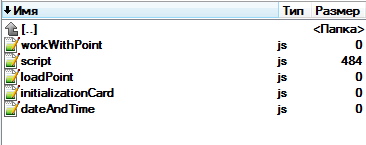
**Рисунок 3.2 ­ Структура файлов папки css**

accordion.css – здесь мы описываем внешний вид правых меню «Категории» и «Доступность».

jquery-ui-1.10.4.custom.css – описание внешнего вида области подсказки для поля «Поиск объекта».

## 3.2 Папка js

В папке js содержатся файлы с расширением js, которые отвечают за программный доступ к объектам приложения (см. рисунок 3.3). Эти скрипты обеспечивает уровень взаимодействия с веб-страницей, которого невозможно достичь с помощью простых документов HTML.



**Рисунок 3.3 ­ Структура файлов папки js**

dateAndTime.js – скрипт для показа времени и даты.

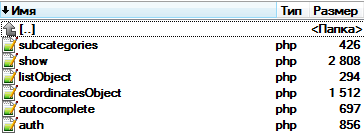
initializationCard.js – скрипт инициализации карты.

loadPoint – загрузка точек.

workWithPoint – этот скрипт содержит работу с маркерами, отправку запросов на получение данных на сервер и обработку ответов сервера.

## 3.3 Папка php

В папке php содержатся файлы с расширением php, которые отвечают за генерации [HTML](http://ru.wikipedia.org/wiki/HTML)-страниц на [веб-сервере](http://ru.wikipedia.org/wiki/%D0%B2%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) и работы с [базами данных](http://ru.wikipedia.org/wiki/%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (см. рисунок 3.4). Эти скрипты обеспечивает уровень взаимодействия с веб-сервером.



**Рисунок 3.4 ­ Структура файлов папки js**

Auth – это скрипт подключения к серверу. В данном скрипте устанавливается соединение с БД с помощью команды *mysql\_connect()*.Это команда включает в себя параметры: *hostname* – название(путь) сервера с MySQL, *username* – имя пользователя, *password* – пароль пользователя, *dbName* – название БД.

Листинг 3.1 – Подключение к серверу

<?php

$hostname = "127.0.0.1";

$username = "name";

$password = "password";

$dbName = "nobarier\_grsu\_by";

mysql\_connect($hostname, $username, $password) or die ("Не могу создать соединение");

mysql\_query('SET NAMES utf8') or header('Location: Error');

/\* Выбираем базу данных. Если произойдет ошибка - вывести ее \*/

mysql\_select\_db($dbName) or die (mysql\_error());

?>

Show – этот скрипт обрабатывает запрос на сервере на получение выбранных объектов. В данном скрипте происходит обработка запроса. Далее строится запрос, а полученный результат сохраняется в массив. Затем поолученный массив запаковывается в json с помощью функции json\_encode(). Передоваеммый Масив включает в себя:

namCat – название категории организации;

idOrgan – ид организации;

nameStreet – название улицы

houseAdres – номер дома;

gpsxAdres – x-координаты объекта;

gpsyAdres – y-координаты объекта;

nameOrganization – название организации;

elemMass – массив наличие инфраструктуры объекта.

Subcategories – этот скрипт обрабатывает запрос на сервере на получение списка подкатегорий выбранной категории. В данном скрипте происходит обработка запроса. Далее строится запрос, а полученный результат сохраняется в массив. Затем поолученный массив запаковывается в json с помощью функции json\_encode(). Передоваеммый Масив включает в себя:

* Id – ид подкатегории;
* sh\_name – краткое название подкатегории;

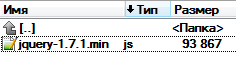
listObject

coordinatesObject

autocomplete – этот скрипт обрабатывает запрос на сервере на получение выбранных объектов. В данном скрипте происходит обработка запроса. Далее строится запрос, а полученный результат сохраняется в массив. Затем поолученный массив запаковывается в json с помощью функции json\_encode().

## 3.4 Папка lib

Папка lib содержит внешние библиотеки (см. рисунок 3.3).

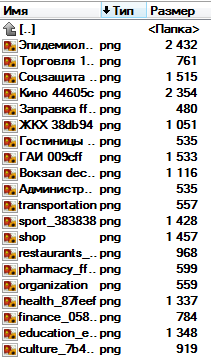
****

**Рисунок 3.5 – Структура файлов папки lib**

jquery-1.7.1.min.js – меню-аккордеон. Данный скрипт обеспечивает функциональность левого меню.

## 3.5 Папка resources

В папке resources содержатся картинки с расширением png, которые в дальнейшем используются для отображения их на странице.



**Рисунок 3.6 ­ Структура файлов пакета resources**

## 3.3 Выводы по главе 3

На этапе реализации информационного сервиса для лиц с ограниченными возможностями:

* реализованы и описаны основные компоненты приложения;
* рассмотрены основные принципы функционирования и особенности реализации этих компонентов;
* приведена основная структура папок входящих в программную реализацию;
* описана необходимость всех папок в программном продукте;
* полностью описана архитектура комплекса реализованной системы;

# **ЗАКЛЮЧЕНИЕ**

Были решены следующие задачи:

* Обсуждение с представителем Гродненского городского исполнительного комитета (горисполкома) и с председателем Гродненского отделения республиканской ассоциации инвалидов-колясочников сущности проекта.
* Анализ и обобщение материала, связанного с организацией путеводителя для лиц с ограниченными возможностями.
* Подбор технологий, программных средств и проектирование системы.

Разработанное приложение позволяет лицам с ограниченной ответственностью подобрать нужный объект (организацию) для той или иной функции, согласно своим возможностям.

В результате окончания работы данная система успешно запущена и используется горисполкомам. Система будет продолжать свое развитее и расширение функциональности.

Перспективы:

* Прорисовка маршрута;
* Возможность оставить отзыв, или задать вопрос;
* Возможность добавить новый объект;
* Рейтинг актуальность информации об объекте.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

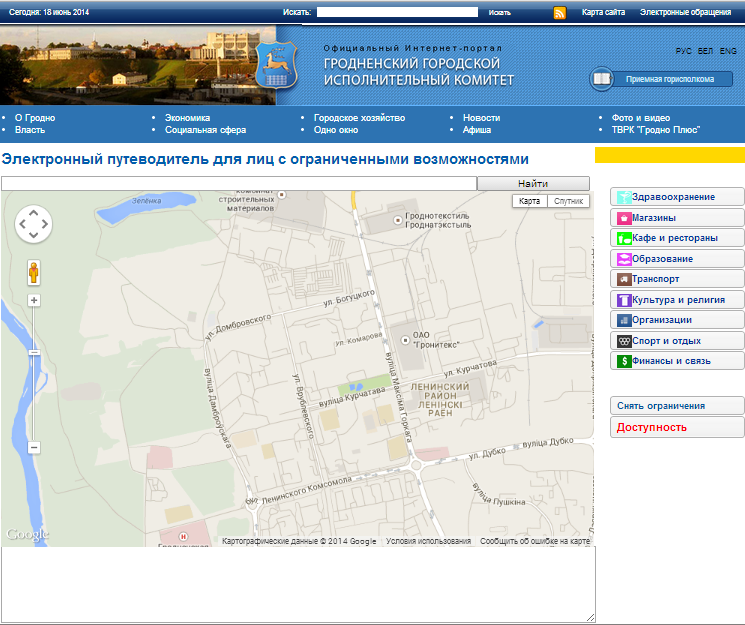
1. Whois, Domain Counts & Internet Statistics [Электронный ресурс] – Режим доступа: http://www.whois.sc/internet-statistics – Дата доступа: 17.02.14.
2. Binkley, D. Semantics guided regression test cost reduction / IEEE Transaction on Software Engineering. – 1997. – Т. 23. – № 8. – С. 498-516.
3. Автоматизированное тестирование [Электронный ресурс] – Режим доступа:http://ru.wikipedia.org/wiki/Автоматизированное\_тестирование – Дата доступа: 04.02.14.
4. Fuzz testing [Электронный ресурс] – Режим доступа: http://en.wikipedia.org/wiki/Fuzz\_testing – Дата доступа: 04.02.14.
5. Статистика уязвимостей web-приложений 2012 / Positive Technologies. – Москва, 2013. – 33 с.
6. WEBSITE SECURITY STATISTICS REPORT, MAY 2013/ WhiteHat Security. – 2013. – С. 21 – 33 с.
7. SQL-Injection от А до Я / Positive Technologies, Дмитрий Евтеев. – Москва, 2008. – 61 с.
8. Information Leakage – WASC Threat Classification [Электронный ресурс] – Режим доступа: http://projects.webappsec.org/w/page/13246936/Information%Leakage – Дата доступа: 05.02.14.
9. Statistics Show Average Website Has Serious Vulnerabilities Almost Every Day of the Year [Электронный ресурс] – Режим доступа: http://www.cunatechnologycouncil.org/news/4186.html – Дата доступа: 07.02.14.
10. Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е издание. – Вильямс, 2013. – 1311 с.;
11. Список сроков поддержки по продуктам [Электронный ресурс] – Режим доступа: http://support.microsoft.com/gp/lifeselectindex/ru – Дата доступа: 29.01.14.
12. Supported Platforms – Mono [Электронный ресурс] – Режим доступа: http://www.mono-project.com/Supported\_Platforms – Дата доступа: 30.01.14.

# ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Рассмотрим несколько способов использования:

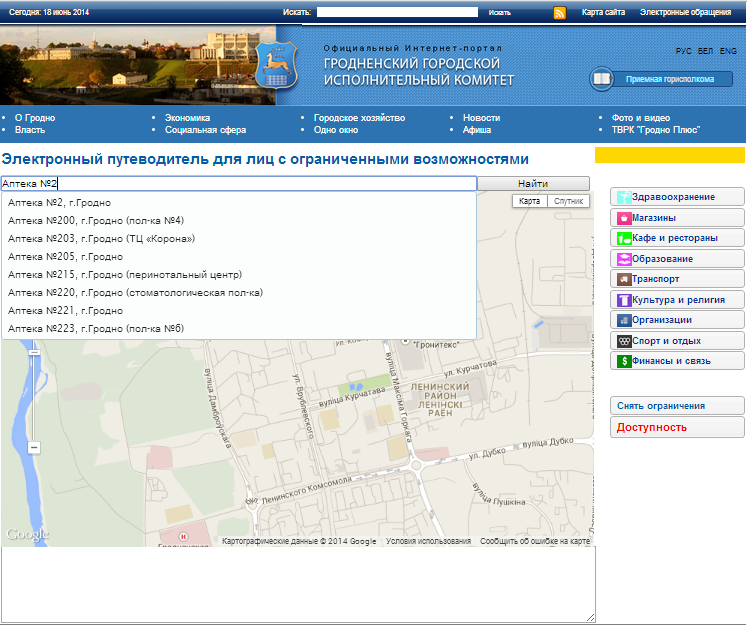
1. Поиск объекта по названию;
2. Поиск объектов по подкатегории;
3. Поиск с выбором наличия инфраструктуры.
4. Поиск объекта по названию.

Сервис находится по адресу nobarier.grsu.by (см. рисунок 1).



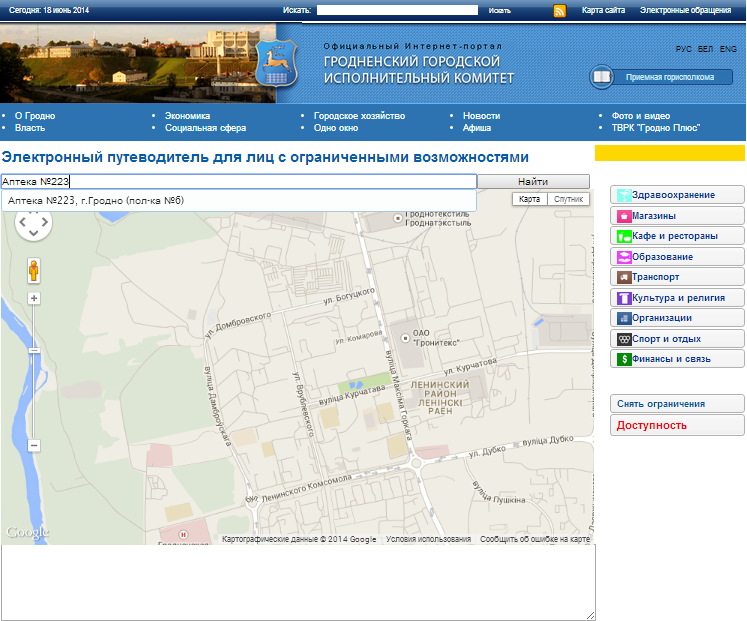
**Рисунок 1 – Вид программы**

Найдем аптеку №223. Введем в поле поиска название объекта (см. рисунок 2).



**Рисунок 2 – Ввод названия объекта в поле поиска**

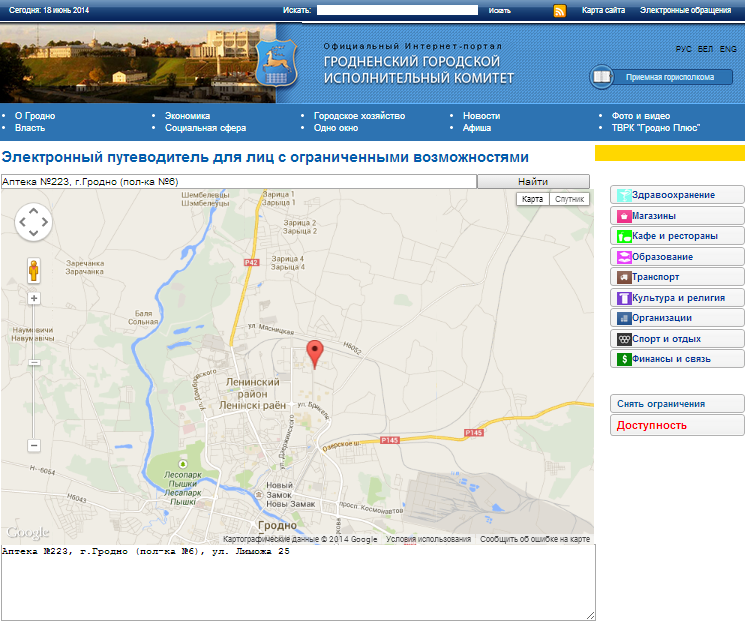
При вводе названия появляется подсказка с возможными вариантами выбора. Можно выбрать один из предложенных вариантов, переместив курсор и щелкнув по нужной позиции (см. рисунок 3).



**Рисунок 3 – Выбор нужного объекта**

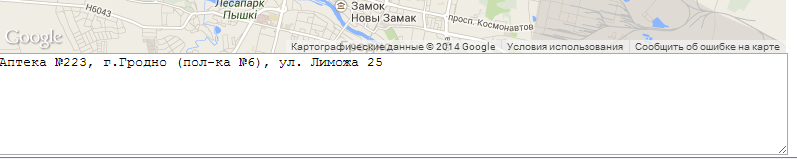
После того, как выбран нужный объект, нажимаем кнопку «Найти».

В результате найденный объект отобразился маркером на карте (см. рисунок 4).



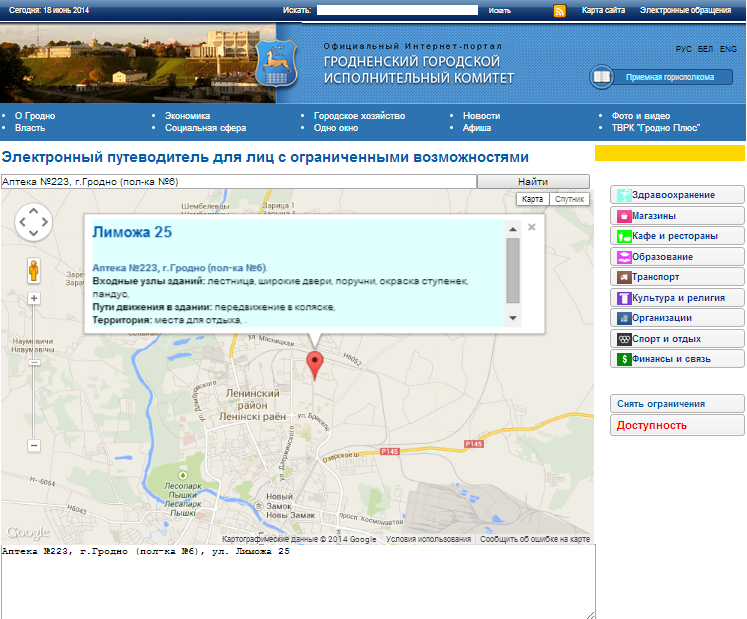
**Рисунок 4 – Найденный объект на карте**

Внизу, в поле «Краткая информация об объекте» отобразилось название и адрес объекта (см. рисунок 5).



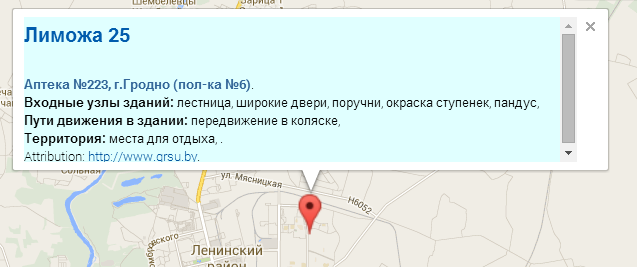
**Рисунок 5 – Краткая информация об объекте**

Подробно описание объекта с наличием имеющейся инфраструктуры можно получить, наведя курсор мышки на маркер и щелкнув по нему (см. рисунок 6).



**Рисунок 6 – Подробное описание объекта**

Информационное окно (см. рисунок 7) содержит: адрес объекта, его полное название, список наличия инфраструктуры, в результате которого можно узнать какая инфраструктура присутствует в выбранном объекте, и ссылки на текущую информацию об объекте.

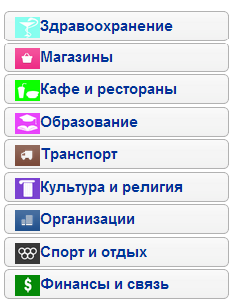


**Рисунок 7 – Информационное окно**

1. Поиск объектов по выбранной подкатегории.

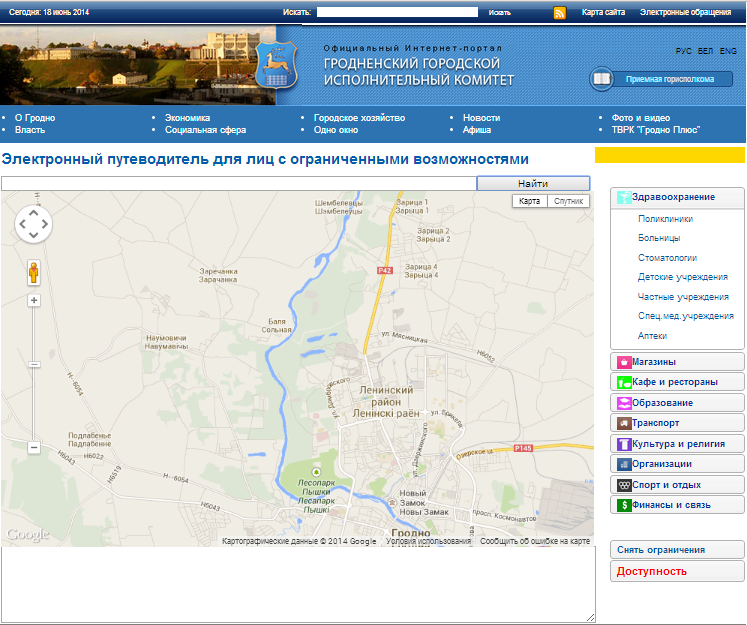
Найдем все объекты подкатегории «Поликлиники» для этого выполним следующие шаги.

В правой части окна находится меню «Категории» (см. рисунок 8).



**Рисунок 8 – меню «Категории»**

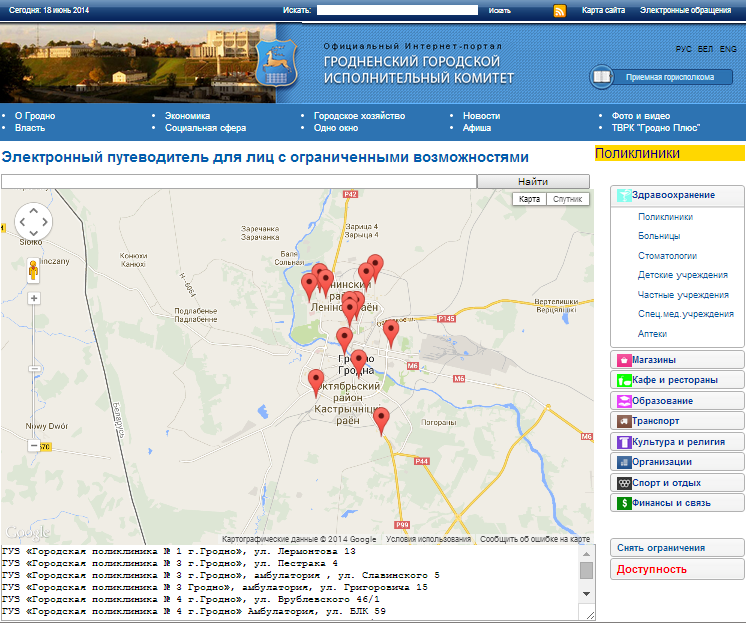
В меню «Категории» выберем категорию «Здравоохранение», где находится нужная нам подкатегория. Наведем курсор мышки на категорию «Здравоохранение» и щелкнем по ней (см. рисунок 9).



**Рисунок 9 – Список подкатегорий категории «Здравоохранение»**

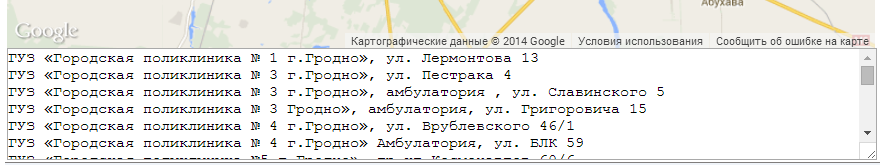
Выберем нужную нам подкатегорию «Поликлиники» в развернутом списке и щелкнем по ней.

На карте появились все объекты подкатегории «Поликлиники», помеченные маркерами. В верхнем правом угле желтым индикатором выделилась выбранная нами подкатегория (см. рисунок 10).



**Рисунок 10 – Найденные объекты на карте**

Внизу, в поле «Краткая информация об объектах» находится список краткого описания все выбранных объектов(см. рисунок 11).



**Рисунок 11 – Краткая информация об объектах**

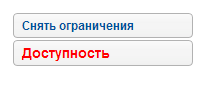
Подробную информацию об объекте можно получить, щелкнув по маркеру нужного объекта (см. рисунок 12).



**Рисунок 12 – Подробное описание выбранного объекта**

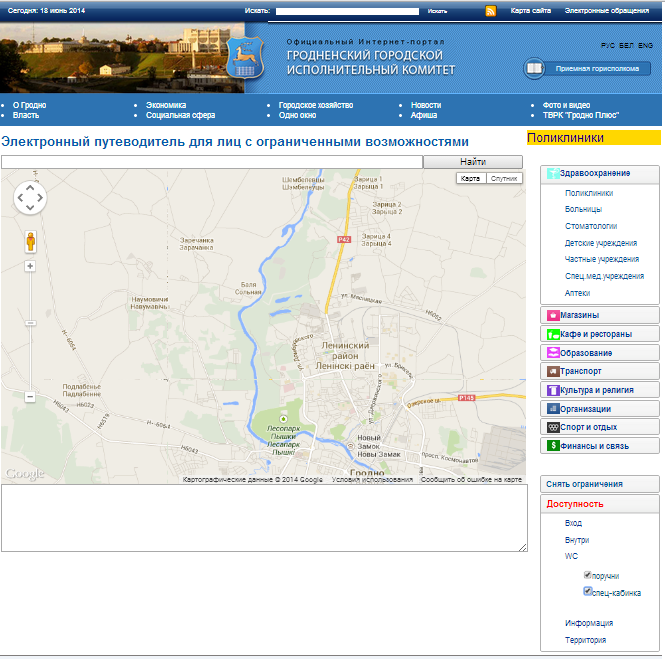
1. Поиск с выбором наличия инфраструктуры.

В случае, когда нам нужно найти объекты с наличием какой-нибудь инфраструктуры, имеется меню «Ограничения» (см. рисунок 13). Мы можем добавить метку нужного нам ограничения и снять все метки.

****

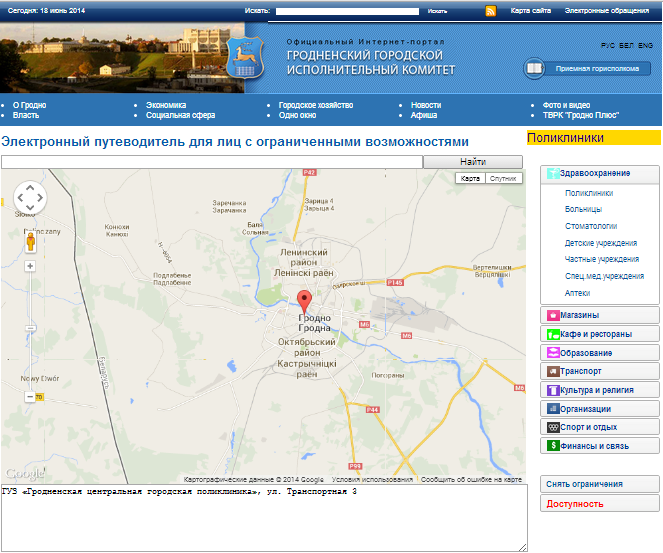
**Рисунок 13 – Меню «Ограничения»**

Добавим метку нужной нам инфраструктуры. Пускай это будет WC поручни и спец-кабинка (см. рисунок 14).



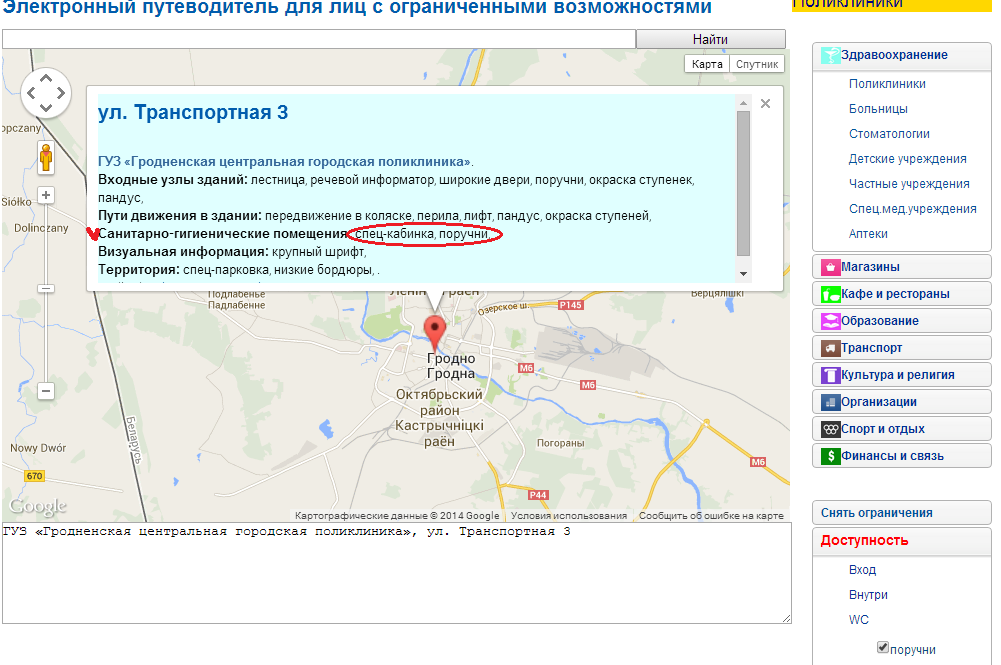
**Рисунок 14 – Добавление доступностей**

Выберем сейчас подкатегорию поликлиники (см. рисунок 15).



**Рисунок 15 – Найденные объекты на карте с доступностями**

Щелкнем по маркеру и убедимся в правильности работы системы (см. рисунок 16).



**Рисунок 16 – Проверка работы поиска с доступностями**