|  |  |
| --- | --- |
| Институт (факультет) | Институт Информационных Технологий |
| Кафедра | Математического и Программного Обеспечения ЭВМ |

**КУРСОВОЙ ПРОЕКТ**

|  |
| --- |
| по дисциплине Технология разработки программного обеспечения |
| по модулю Управление программными проектами |

|  |  |
| --- | --- |
| на тему | «Разработка программного обеспечения информационного портала |
| "Социополис 35": "Туристический и досуговый сервисы"» | | |

|  |
| --- |
| Выполнил студент группы |
| 1ПИб-01-41оп |
| направление подготовки (специальности) |
| 09.03.04, Программная инженерия |
| *шифр, наименование* |
| Гришкин Дмитрий Евгеньевич |
| *фамилия, имя, отчество* |

|  |
| --- |
| Руководитель |
| Ершов Евгений Валентинович |
| *фамилия, имя, отчество* |
| Доктор технических наук, профессор |
| *должность* |

|  |
| --- |
| Дата представления работы |
| «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г. |
|  |
| Заключение о допуске к защите |
|  |
|  |
|  |
|  |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| количество баллов |
| Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Оглавление

[Введение 4](#_Toc512188591)

[1 Основная часть 5](#_Toc512188592)

[1.1 Сравнительный анализ известных технических решений в области туристических и досуговых сервисов 5](#_Toc512188593)

[1.2 Выбор технологий, сред и языков программирования 10](#_Toc512188594)

[1.2.1 Выбор модели жизненного цикла информационного портала социальных сервисов и услуг «Социополис 35» 10](#_Toc512188598)

[1.2.2 Выбор подхода к разработке 11](#_Toc512188599)

[1.2.3 Выбор инструментальных средств 12](#_Toc512188600)

[1.2.4 Выбор базы данных 14](#_Toc512188601)

[1.3 Анализ и уточнение требований к программному обеспечению 15](#_Toc512188602)

[1.3.1 Анализ процесса обработки информации и выбор структур данных для её хранения 15](#_Toc512188603)

[1.4 Разработка спецификаций проектируемой системы 19](#_Toc512188604)

[1.4.1 Построение диаграмм вариантов использования 19](#_Toc512188605)

[1.4.2 Построение контекстных диаграмм классов 25](#_Toc512188606)

[1.4.3 Разработка модели процессов туристических и досуговых сервисов 26](#_Toc512188607)

[1.4.4 Построение диаграмм деятельностей сценариев вариантов использования 33](#_Toc512188608)

[1.5 Проектирование системы 34](#_Toc512188609)

[1.5.1 Проектирование структуры системы и построение диаграмм пакетов 34](#_Toc512188610)

[1.5.2 Проектирование классов пакетов 35](#_Toc512188611)

[1.5.2.1 Проектирование классов пакета “ГИС” 35](#_Toc512188612)

[1.5.3 Построение диаграмм компонентов 41](#_Toc512188613)

[1.5.4 Построение диаграмм размещения 42](#_Toc512188614)

[1.6 Проектирование интерфейса пользователя 43](#_Toc512188615)

[1.6.1 Построение графа диалога 43](#_Toc512188616)

[1.7 Выбор стратегии тестирования, разработка тестов, программа и методика испытаний 44](#_Toc512188617)

[1.7.1 Объект испытаний 45](#_Toc512188618)

[1.7.2 Цель испытаний 45](#_Toc512188619)

[1.7.3 Требования к информационному, аппаратно-программному обеспечению и документации 45](#_Toc512188620)

[1.7.3.1 Требования к функциональным характеристикам 45](#_Toc512188621)

[1.7.3.2 Требования к надежности 46](#_Toc512188622)

[1.7.3.3 Требования к аппаратному и информационному обеспечению 46](#_Toc512188623)

[1.7.3.4 Требования к программной документации 46](#_Toc512188624)

[1.7.4 Состав, порядок и методы испытаний 46](#_Toc512188625)

[1.7.4.1 Состав испытаний 47](#_Toc512188626)

[1.7.4.2 Методы испытаний 48](#_Toc512188627)

[1.7.5 Результаты проведения испытаний 49](#_Toc512188628)

[2 Технико-экономическое обоснование 51](#_Toc512188629)

[2.1 Трудозатраты на разработку и отладку программы 51](#_Toc512188630)

[2.2 Расчет себестоимости программного продукта 55](#_Toc512188631)

[2.3 Расчет цены программного продукта 59](#_Toc512188632)

[2.4 Расчет экономической эффективности 59](#_Toc512188633)

[Заключение 63](#_Toc512188634)

[Список литературы 65](#_Toc512188635)

[Приложение 1. Техническое задание 66](#_Toc512188636)

[Приложение 2 71](#_Toc512188637)

[Приложение 3 72](#_Toc512188638)

# Введение

В рамках программы опорного ВУЗа Череповецким государственным университетом принято решение о реализации проекта «Социополис.35», в рамках которого осуществляется:

* исследование образа города Череповца и его внутренних закономерностей развития;
* создание привлекательного внешнего облика города;
* проведение ряда конференций социологов, предпринимателей, экономистов, экологов, архитекторов по обмену знаниями и опытом в сфере развития города;
* продвижение города как объекта туризма.

В рамках данного проекта принято решение о создании и поддержке информационного портала социальных сервисов и услуг «Социополис.35», который призван обеспечить жителей города Череповца востребованными, удобными и уникальными для жителей города социальные сервисами и услугами.

# Основная часть

## Сравнительный анализ известных технических решений в области туристических и досуговых сервисов

В ходе проведенного анализа имеющихся аналогов туристических и досуговых сервисов представлены следующие информационные сайты и порталы, включающие рассматриваемые сервисы, c указанием имеющихся достоинств и недостатков:

1) Информационный портал в франшизе «ГИС 4geo» (cherepovets.4geo.ru).

Страница «Места отдыха» информационного портала в франшизе «ГИС 4geo» представлена на рис. 1.

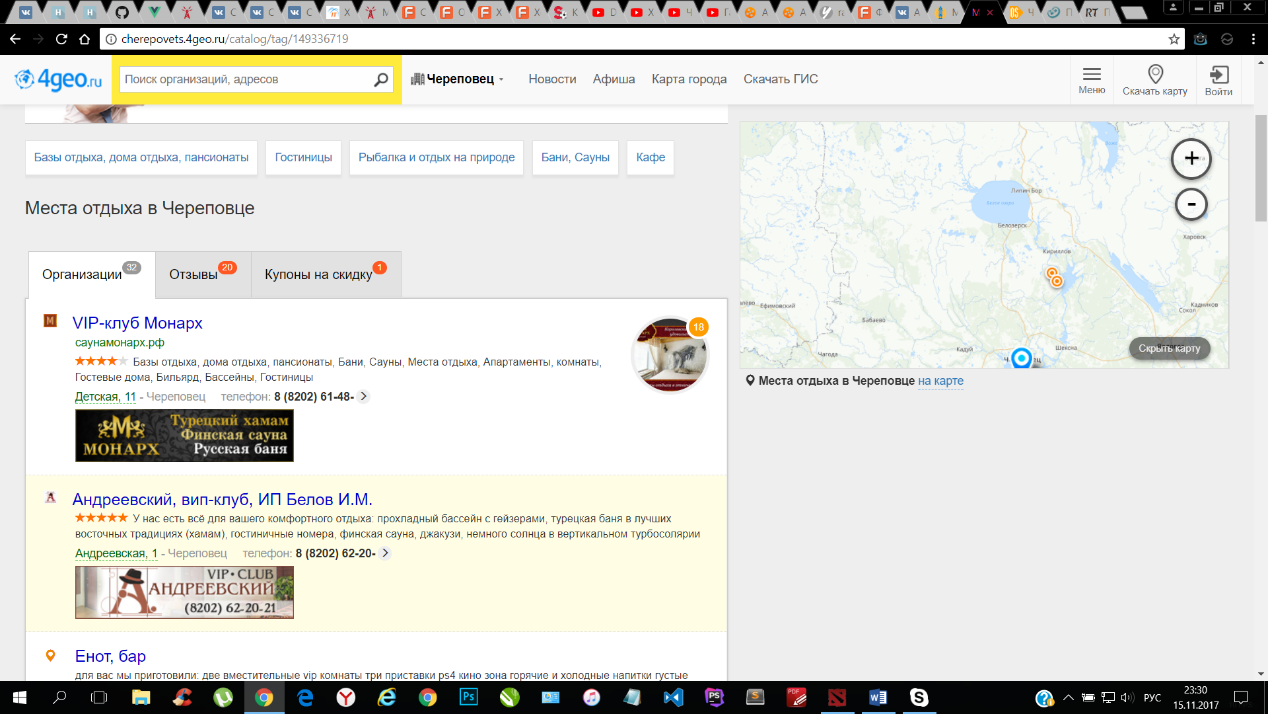


Рис. 1. Страница сайта «cherepovets.4geo.ru»

Достоинства сервиса:

* разделение мероприятий на подгруппы для удобного поиска развлекательных комплексов;
* наличие контактной информации;
* указание местоположения на карте;
* возможность пользователю оставить заявку (рис.2).

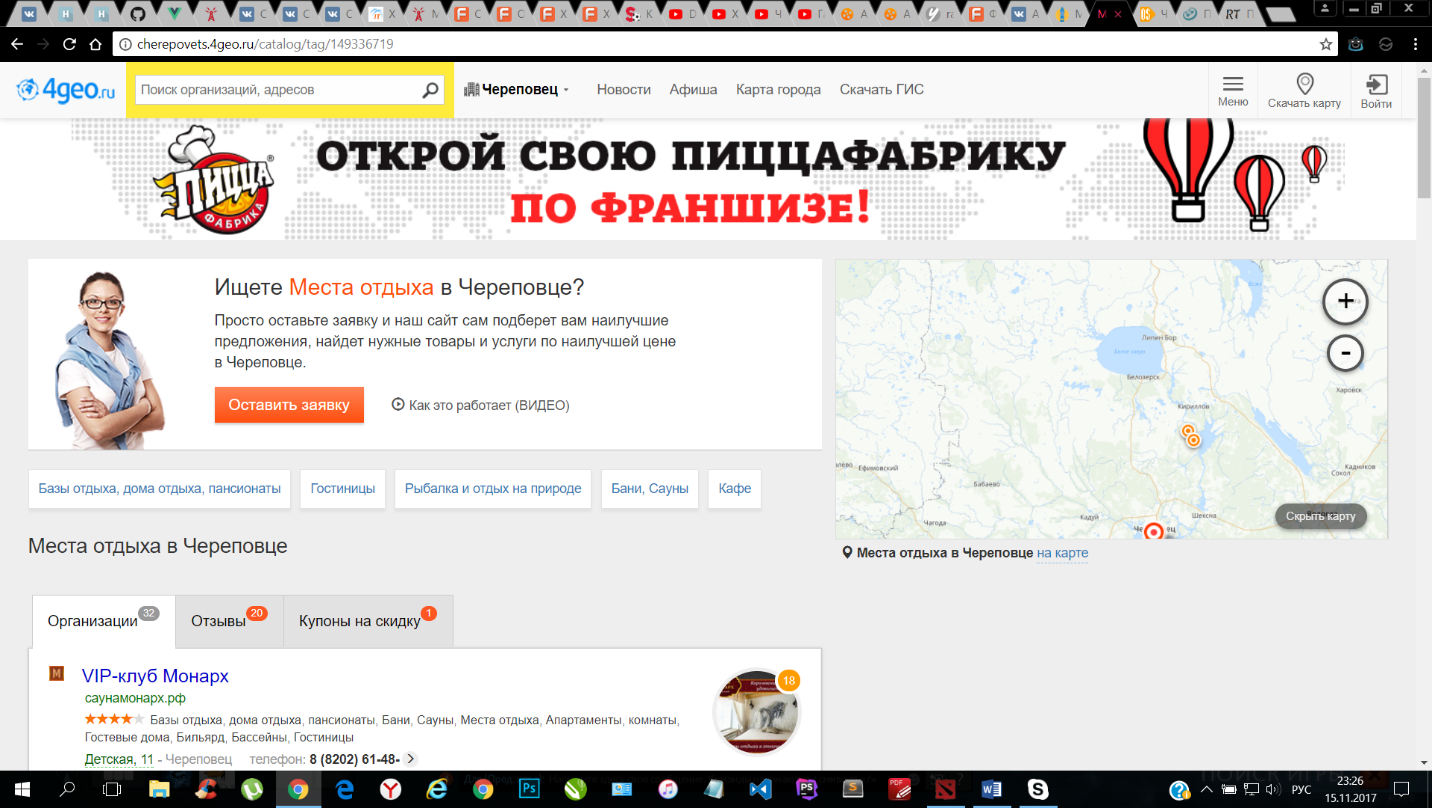


Рис.2. Окно предложения заявки

Недостатки сервиса:

* отсутствие путеводителей;
* отсутствие музеев.

2) cherinfo - официальный сайт г. Череповца (gid.cherinfo.ru).

Страница «Места отдыха» официального сайта г. Череповца - cherinfo представлена на рис. 3.

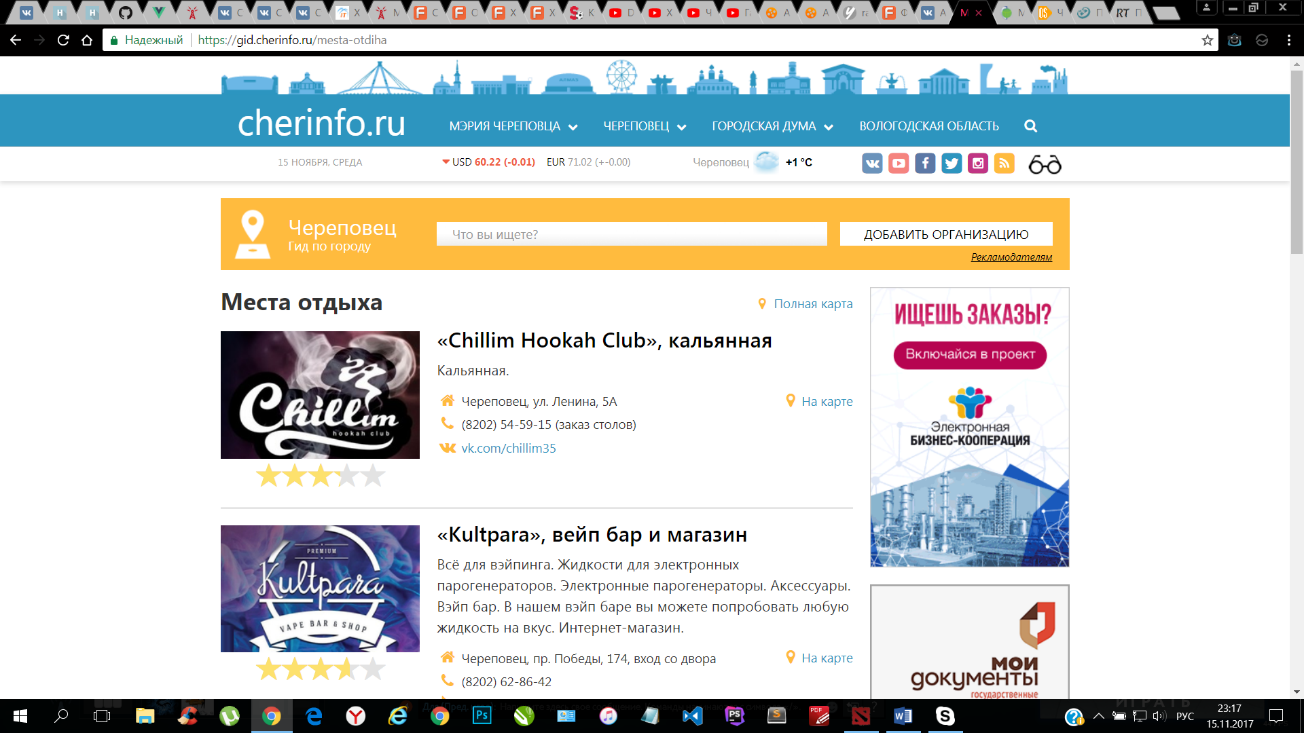


Рис.3. Страница сайта gid.cherinfo.ru

Достоинства сервиса:

* разделение на подгруппы для удобного поиска;
* удобный и приятный интерфейс;
* наличие музеев и развлекательных комплексов;
* наличие контактной информации;
* указание местоположения на карте.

Недостатки сервиса:

* отсутствие путеводителей.

3) Журнал путешествия по России (rutrip.net).

Страница туристического сервиса сайта rutrip.net представлена на рис. 4.

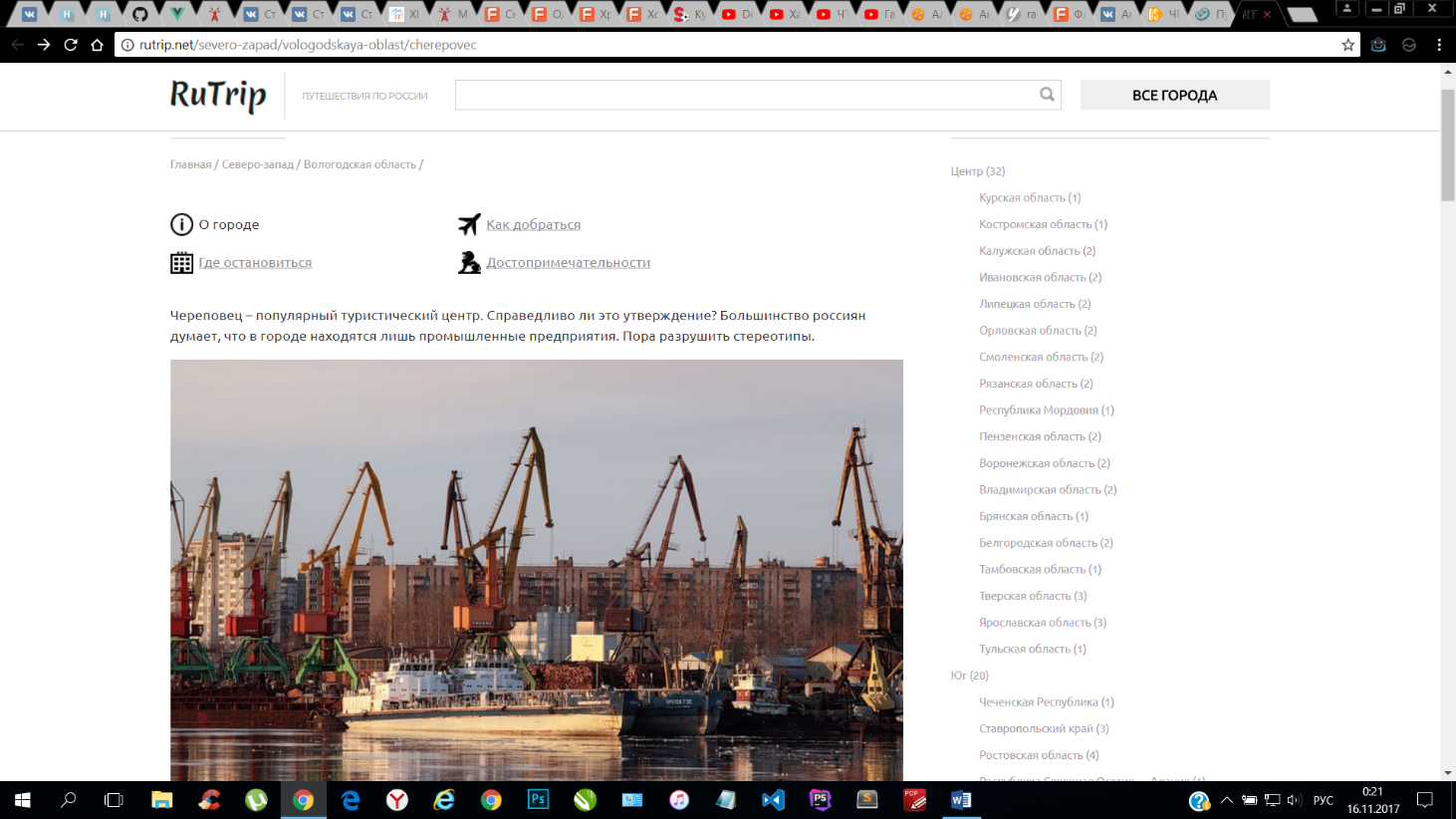


Рис. 4. Страница сайта rutrip.net

Достоинства сервиса:

* фотографии и местонахождение достопримечательностей города.

Недостатки сервиса:

* нет музеев и развлекательных комплексов;
* нет контактной информации;
* отсутствие путеводителей.

4) Крупнейший в мире сайт о путешествиях (tripadvisor.ru).

Страница туристического сервиса сайта tripadvisor.ru представлена на рис. 5.

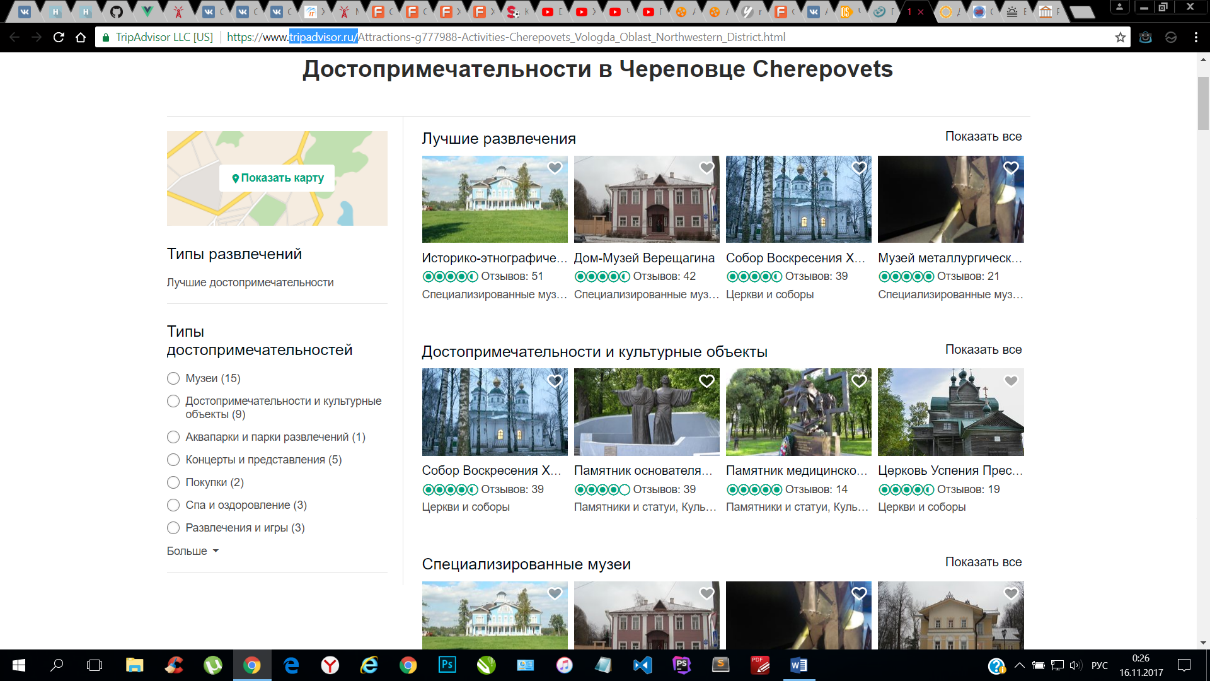


Рис. 5. Страница сайта tripadvisor.ru

Достоинства сервиса:

* фотографии и местонахождение достопримечательностей города;
* удобный и приятный интерфейс.

Недостатки сервиса:

* нет музеев и развлекательных комплексов;
* нет контактной информации;
* отсутствие путеводителей.

5) Бизнес-каталог Вологодской области (reestr35.ru).

Страница досугового сервиса сайта reestr35.ru представлена на рис. 6.

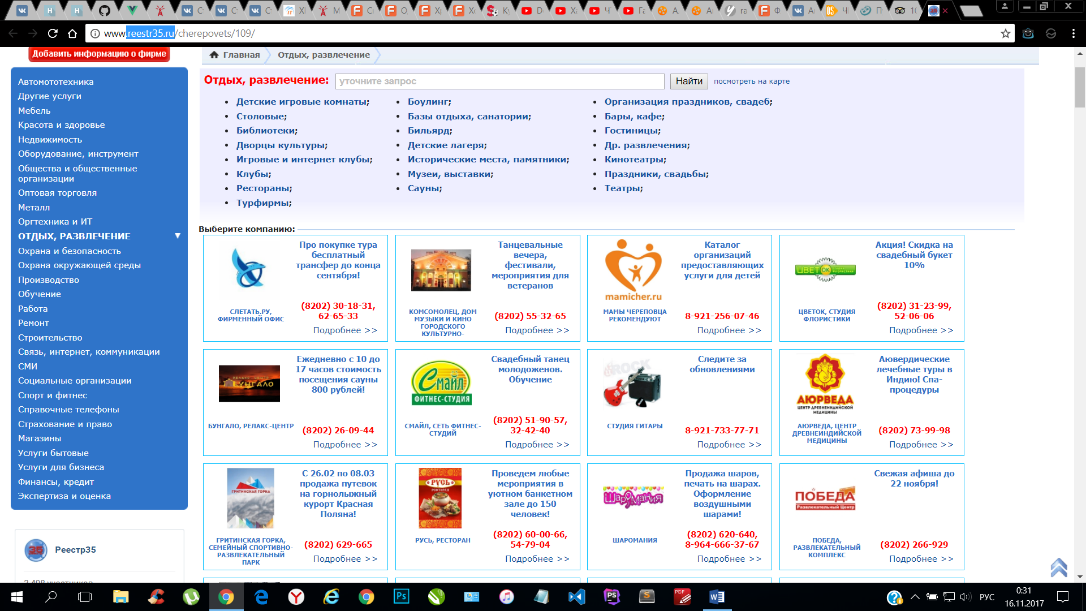


Рис. 6. Страница афиши «reestr35.ru»

Достоинства сервиса:

* разделение на подгруппы для удобного поиска;
* наличие театров, музеев и развлекательных комплексов;
* наличие контактной информации;
* указание местоположения на карте.

Недостатки сервиса:

* отсутствие путеводителей.

Вывод: в ходе проведенного анализа выявлено, что разработка удобного туристического и досугового веб-сервиса для жителей г. Череповца является востребованной задачей.

Из всех рассмотренных вариантов не один сайт не обладает наличием путеводителей, так же во многих случаях нет театров и музеев.

У части сайтов очень неудобный интерфейс, который отталкивает посетителей. Некоторые сайты не адаптированы под мобильные устройства и просматривать страницу со смартфона крайне неудобно.

## Выбор технологий, сред и языков программирования

1. 2. 1. Выбор модели жизненного цикла информационного портала социальных сервисов и услуг «Социополис 35»

Жизненный цикл программного обеспечения (ЖЦ ПО) – это период от момента возникновения идеи создания программного обеспечения до момента завершения поддержки программного обеспечения фирмой-разработчиком или фирмой, выполняющей сопровождение.

Под моделью ЖЦ понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ.

Для разработки информационного портала «Социополис.35» проведен анализ моделей ЖЦ и выбрана каскадная модель ЖЦ с промежуточным контролем (рис. 7), которая предусматривает последовательное выполнение этапов ЖЦ с возможностью возврата на предыдущие этапы.

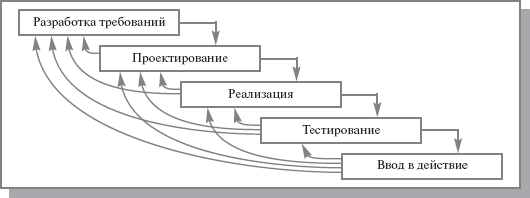


Рис. 7. Каскадная модель с промежуточным контролем

Каскадная модель ЖЦ с промежуточным контролем имеет следующие преимущества для разрабатываемого информационного портала:

* данная модель ЖЦ позволяет сформулировать достаточно полно сроки и этапы разработки на ранних этапах, в отличие от спиральной и инкрементной моделей, что является немаловажным для проектируемой системы, поскольку разработка будет вестись в ограниченные сроки;
* явным преимуществом каскадной модели ЖЦ с промежуточным контролем является то, что модель подходит для систем, у которых требования удается выявить на ранних этапах проектирования. Это справедливо для информационного портала «Социополис.35», поскольку его разработка преследует решение конкретной проблемы обеспечения жителей города Череповца удобными и востребованными социальными сервисами, что позволяет достаточно сформулировать требования к системе ещё на этапе анализа;
* недостатки указанной модели связаны прежде всего с разработкой систем, для которых требования не удается выявить на ранних этапах проектирования, что представляется маловероятным для информационного портала «Социополис.35».
  + 1. Выбор подхода к разработке

Для разработки программного обеспечения в настоящее время применяют два подхода:

* структурный подход;
* объектный подход.

В основе структурного подхода лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм.

В основе объектного подхода лежит объектная декомпозиция, т. е. представление разрабатываемого программного обеспечения в виде совокупности объектов, в процессе взаимодействия которых через передачу сообщений происходит выполнение требуемых функций.

Достоинства объектного подхода по сравнению со структурным:

* более естественная декомпозиция предметной области, которая существенно облегчает разработку;
* независимая разработка отдельных частей (объектов) программы;
* существенное увеличение степени повторного использования кода за счет способов организации программ, основанных на механизмах наследования, полиморфизма, композиции, наполнения;
* наличие стандартного средства описания разработки программных продуктов с использованием объектного подхода - языка UML (unified modeling language – унифицированный язык моделирования).

Поскольку для предметной области информационного портала «Социополис.35» естественной декомпозицией является объектная декомпозиция и на основании приведенных преимуществ объектного подхода для разрабатываемого ПО выбран объектный подход к разработке.

* + 1. Выбор инструментальных средств

Разрабатываемый информационный портал «Социополис.35» имеет клиент-серверную архитектуру.

После проведения сравнительной характеристики (табл. 1), в качестве языка программирования, для серверной части сервиса Социополис.35, был выбран язык php на основании приведенных преимуществ:

* у языка низкий порог вхождения, и он легок в изучении;
* встраиваемый язык, это обеспечивает большую гибкость;
* сценарии выполняются с большой скоростью, что делает его эффективнее.

Таблица 1

Сравнение серверных языков

|  |  |  |  |
| --- | --- | --- | --- |
|  | PHP | JAVA | PYTHON |
| Порог вхождения | Низкий порог вхождения | Высокий порог вхождения | Низкий порог вхождения |
| Изучение | Простота изучения | Сложность  изучения | Простота изучения |
| Работа с сайтом | Работа из “коробки” | Работа из Java-машины | Использование дополнительных библиотек |
| ООП | Объектно-ориентированный  язык | Объектно-ориентированный  язык | Объектно-ориентированный  язык |
| Безопасность | Высокая безопасность | Высокая безопасность | Высокая безопасность |
| Встраиваемость | Встраиваемый язык | Не встраиваемый | Не встраиваемый |

Для разработки клиентской части были выбраны препроцессор Html Pug и препроцессор Css Sass, а также jQuery, так как они упрощают и ускоряют разработку путем сокращения и облегчения синтаксиса.

Так же для сборки проекта будет использоваться система сборки Gulp, которая позволит быстро собирать проект.

Кратко опишем используемые выбранные клиентские технологии:

* jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX;
* Jade — это препроцессор HTML, а также шаблонизатор, написанный на JS для Node.js;
* Sass (Syntactically Awesome Stylesheets) — модуль, включенный в Haml. Sass — это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS кода и упрощения файлов каскадных таблиц стилей;
* Gulp — это инструмент сборки веб-приложения, позволяющий автоматизировать повторяющиеся задачи, такие как сборка и минификация CSS- и JS-файлов, запуск тестов, перезагрузка браузера и т.д. Тем самым Gulp ускоряет и оптимизирует процесс веб-разработки. В статье мы рассмотрим основы использования этого инструмента.

Для работы с PHP, jQuery, Sass и Pug была выбрана программа PhpStorm.

PhpStorm - кроссплатформенная интегрированная среда разработки для PHP. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA. PhpStorm представляет собой интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Имеется полноценный SQL-редактор.

* + 1. Выбор базы данных

Для хранения данных была выбрана база данных – MySQL, так как язык php поддерживает работу с этой базой данных. Так же для упрощения работы с базой данных и php была выбрана библиотека RedBean.

ORM (Object-Relational Mapping) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных». Существуют как проприетарные, так и свободные реализации этой технологии.

RedBeanPHP — ORM-библиотека. Основное ее отличие от коллег, типа Propel или Doctrine, в отсутствии необходимости ручного конфигурирования объектов, т.е. никаких xml, yml или ini-файлов. RedBenPHP на лету создает таблицы, поля и индексы. Любой объект можно связать с другим. Из БД поддерживаются MySQL, SQLite и Postgres.

Для работы с базой данных было выбрано веб-приложение PhpMyAdmin.

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PHPMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

## Анализ и уточнение требований к программному обеспечению

### Анализ процесса обработки информации и выбор структур данных для её хранения

Процесс обработки информации, который осуществляется на сайте «Социополис.35» сервиса «Туристических и досуговых сервисы» включает следующие пункты:

1. Отображение информации о реализованных сервисах

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующих сервисов. Выходная информация включает в себя:

* описание сервиса;
* изображение сервиса;
* наименование сервиса.

1. Просмотр путеводителей

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующих путеводителей. Выходная информация включает в себя:

* описание путеводителя;
* изображение путеводителя;
* наименование путеводителя.

1. Отображение маршрута путеводителя

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующих маршрута. Выходная информация представляет собой группу метку, объединённых в единый маршрут, отображенный на карте.

1. Просмотр мест для занятия спортом

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующего места. Выходная информация включает в себя:

* описание места;
* изображение места;
* наименование места;
* адрес места.

1. Просмотр мест для отдыха

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующего места. Выходная информация включает в себя:

* описание места;
* изображение места;
* наименование места;
* адрес места.

1. Просмотр достопримечательности

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующей достопримечательности. Выходная информация включает в себя:

* описание достопримечательности;
* изображение достопримечательности;
* наименование достопримечательности;
* адрес достопримечательности.

1. Просмотр моделей компьютерной реконструкции

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующей модели. Выходная информация включает в себя:

* описание модели;
* изображение модели;
* наименование модели;

1. Отображение места или достопримечательности на карте

Клиентской частью информационным порталом делается запрос к базе данных на сервере, и выполняется загрузка интересующих места или достопримечательности. Выходная информация представляет собой метку, отображенную на карте.

Ключевым элементом разрабатываемого информационного портала «Социополис.35» сервиса «Туристические и досуговые сервисы» является база данных, размещаемая на удаленном сервере, которая содержит сведения об общественном транспорте, его координаты, также сведения об предстоящих и прошедших мероприятиях.

ER-диаграмма представлена базы данных представлена на рис.3.1.

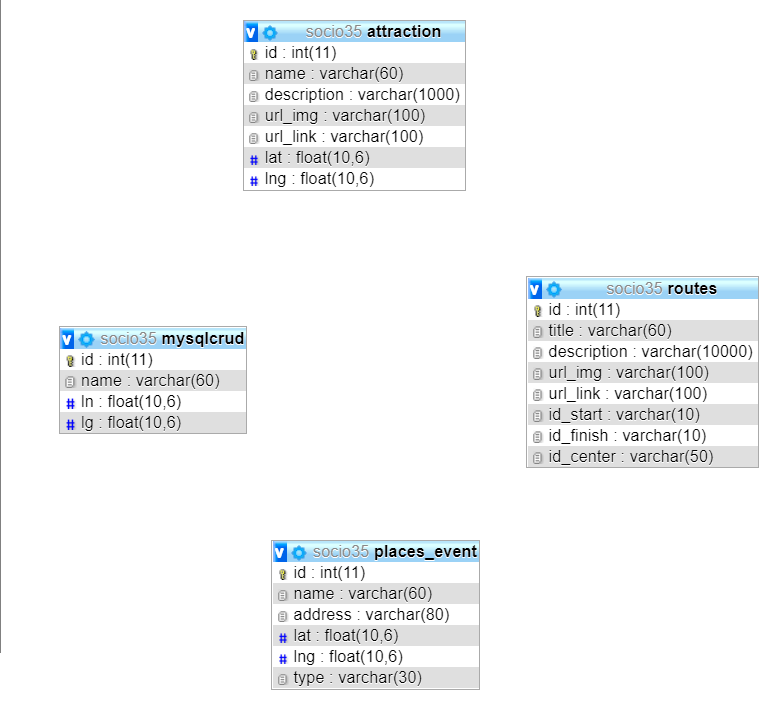


Рис. 3.1. ER-диаграмма

## Разработка спецификаций проектируемой системы

### Построение диаграмм вариантов использования

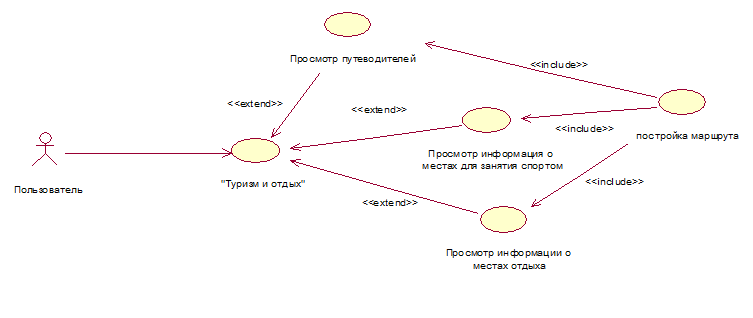


Рис.8. Диаграмма вариантов использования системы «Туристические и досуговые сервисы»

Описание вариантов использования туристических и досуговых сервисов представлено в табл. 2 – 10.

Таблица 2

Описание варианта использования «Туризм и отдых»

|  |  |
| --- | --- |
| Наименование | Туризм и отдых |
| Цель и назначение | Предоставить пользователю список возможных сервисов (услуг), которыми он может воспользоваться |
| Действующее лицо | Пользователь |
| Тип | Основной |

Таблица 3

Описание варианта использования «Туризм и отдых» сценария «Поиск объекта»

|  |  |
| --- | --- |
| Наименование | Поиск объекта |
| Цель и назначение | Пользователь может вводить данные, получать необходимую информацию |
| Действующее лицо | Пользователь |
| Тип | Дополнительный |

Таблица 4

Описание варианта использования «Путеводители»

|  |  |
| --- | --- |
| Наименование | Путеводители |
| Цель и назначение | Пользователь может просмотреть путеводители по городу |
| Действующее лицо | Пользователь |
| Тип | Основной |

Таблица 5

Описание варианта использования «Путеводители» сценария «Прокладка маршрута путеводителя»

|  |  |
| --- | --- |
| Наименование | Прокладка маршрута путеводителя |
| Цель и назначение | Пользователь может вводить данные, получать изображенный маршрут путеводителя на карте |
| Действующее лицо | Пользователь |
| Тип | Дополнительный |

Таблица 6

Описание варианта использования «Путеводители» сценария «Просмотр адресов и описание достопримечательностей»

|  |  |
| --- | --- |
| Наименование | Просмотр адресов и описание достопримечательностей |
| Цель и назначение | Пользователь может выбрать нужную ему достопримечательность, увидеть ее фотографию, описание и путь до неё на карте |
| Действующее лицо | Пользователь |
| Тип | Дополнительный |

Таблица 7

Описание варианта использования «Места для занятия спортом»

|  |  |
| --- | --- |
| Наименование | Места для занятия спортом |
| Цель и назначение | Пользователь может просмотреть места для занятия спортом |
| Действующее лицо | Пользователь |
| Тип | Основной |

Таблица 8

Описание варианта использования «Места для занятия спортом» сценария «Просмотр адресов и описание мест для занятия спортом»

|  |  |
| --- | --- |
| Наименование | Просмотр адресов и описание мест для занятия спортом |
| Цель и назначение | Пользователь может выбрать нужную ему место, увидеть ее фотографию, описание и путь до неё на карте |
| Действующее лицо | Пользователь |
| Тип | Дополнительный |

Таблица 9

Описание варианта использования «Места для отдыха и развлечений»

|  |  |
| --- | --- |
| Наименование | Места для отдыха и развлечений |
| Цель и назначение | Пользователь может просмотреть места для отдыха и развлечений |
| Действующее лицо | Пользователь |
| Тип | Основной |

Таблица 10

Описание варианта использования «Места для отдыха и развлечений» сценария «Просмотр адресов и описание мест для отдыха и развлечений»

|  |  |
| --- | --- |
| Наименование | Просмотр адресов и описание мест для отдыха и развлечений |
| Цель и назначение | Пользователь может выбрать нужную ему место, увидеть ее фотографию, описание и путь до неё на карте |
| Действующее лицо | Пользователь |
| Тип | Дополнительный |

Типичный ход событий основных вариантов использования представлен в табл. 11-13.

Таблица 11

Типичный ход событий варианта использования «Путеводители»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| 1. Пользователь выбирает путеводитель 2. Ввод данных, выбранный элемент из списка | 1. Отображается список путеводителей 2. Обработка данных, отображение нужного путеводителя |

Таблица 12

Типичный ход событий варианта использования

«Места для занятия спортом»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| 1. Пользователь выбирает место для занятия спортом 2. Просмотр маршрута до места | 1. Обработка данных, отображается адрес и информация о месте для занятия спортом 2. Обработка данных, вывод маршрута до места |

Таблица 13

Типичный ход событий варианта использования «Места для отдыха и развлечений»

|  |  |
| --- | --- |
| Действия пользователя | Отклик системы |
| 1. Пользователь выбирает место для отдыха и развлечений 2. Просмотр маршрута до места | 1. Обработка данных, отображается адрес и информация о месте для отдыха и развлечений 2. Обработка данных, вывод маршрута до места |

### Построение контекстных диаграмм классов

Контекстная диаграмма классов туристических и досуговых сервисов представлена на рис. 9.

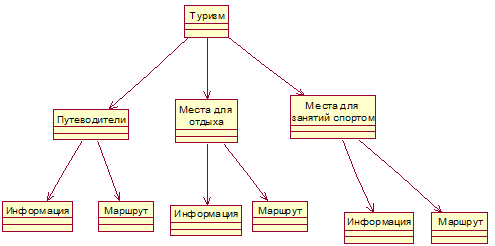


Рис. 9. Контекстная диаграмма классов туристических и досуговых сервисов

Основные классы предметной области пространственно-распределённых навигационных сервисов представлены в табл. 14.

Таблица 14

Основные классы-понятия предметной области туристических и досуговых сервисов

|  |  |
| --- | --- |
| Класс | Описание |
| Туризм | Главная страница сервиса. Находится историческая информация и кнопки для перехода на под сервисы |
| Путеводители | Страница под сервиса, которая содержит путеводители и достопримечательности города |
| Места для отдыха | Страница под сервиса, которая содержит места для отдыха |
| Места для занятия спортом | Страница под сервиса, которая содержит места для занятия спортом |
| Информация | Страница сайта, содержит информацию о под сервисе |
| Маршрут | Страница сайта, содержит алгоритм построение и вывода маршрута |

### Разработка модели процессов туристических и досуговых сервисов

На рис. 10 показана диаграмма последовательностей системы для сценария варианта использования «Просмотр мест для занятия спортом» в процессе взаимодействия пользователя с системой. Операции представлены в табл. 15 – 17.

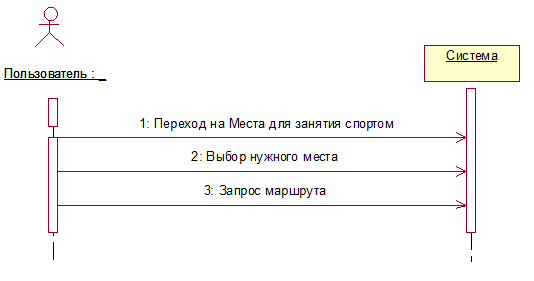


Рис. 10. Диаграмма последовательностей системы для сценария варианта использования «Просмотр мест для занятия спортом»

Таблица 15

Переход на места для занятия спортом

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия  Вывод | Переход на места для занятия спортом  Переход на места для занятия спортом  Системная  Вариант использования «Просмотр мест для занятия спортом»  -  -  Осуществлено переход на места для занятия спортом |

Таблица 16

Выбор нужного места

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия | Выбор нужного места  Выбор нужного места  Системная  Вариант использования «Просмотр мест для занятия спортом»  -  Осуществлен выбор нужного места |

Таблица 17

Запрос маршрута

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия  Вывод | Запрос маршрута  Запрос маршрута  Системная  Вариант использования «Просмотр мест для занятия спортом»  -  Осуществлен запрос маршрута  Осуществлен вывод маршрута |

На рис. 11 показана диаграмма последовательностей системы для сценария варианта использования «Просмотр мест для отдыха» в процессе взаимодействия пользователя с системой. Операции представлены в табл. 18 – 20.

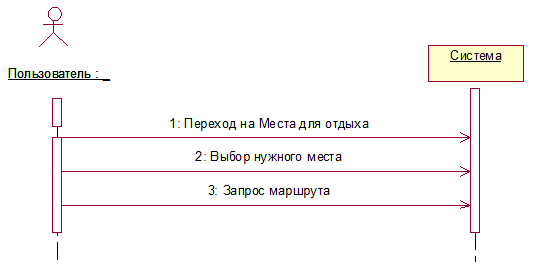


Рис. 11. Диаграмма последовательностей системы для сценария варианта использования «Просмотр мест для отдыха»

Таблица 18

Переход на места для отдыха

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия  Вывод | Переход на места для отдыха  Переход на места для отдыха  Системная  Вариант использования «Просмотр мест для отдыха»  -  -  Осуществлено переход на места для занятия спортом |

Таблица 19

Выбор нужного места

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия | Выбор нужного места  Выбор нужного места  Системная  Вариант использования «Просмотр мест для отдыха»  -  Осуществлен выбор нужного места |

Таблица 20

Запрос маршрута

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия  Вывод | Запрос маршрута  Запрос маршрута  Системная  Вариант использования «Просмотр мест для отдыха»  -  Осуществлен запрос маршрута  Осуществлен вывод маршрута |

На рис. 12 показана диаграмма последовательностей системы для сценария варианта использования «Путеводители» в процессе взаимодействия пользователя с системой. Операции представлены в табл. 21 – 24.

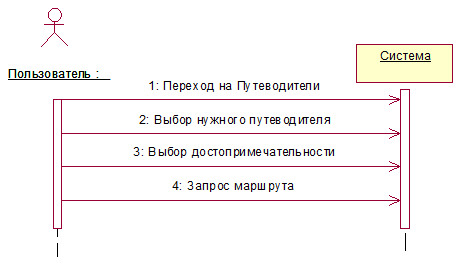


Рис. 12. Диаграмма последовательностей системы для сценария вариантов использования «Путеводители»

Таблица 21

Переход на путеводители

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия  Вывод | Переход на путеводители  Переход на путеводители  Системная  Вариант использования «Путеводители»  -  -  Осуществлено переход на путеводители |

Таблица 22

Выбор нужного путеводителя

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия | Выбор нужного путеводителя  Выбор нужного путеводителя  Системная  Вариант использования «Путеводители»  -  Осуществлен выбор нужного путеводителя |

Таблица 23

Выбор достопримечательности

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия | Выбор достопримечательности  Выбор нужной достопримечательности  Системная  Вариант использования «Путеводители»  Осуществлено переход на путеводители  Осуществлен выбор нужной достопримечательности |

Таблица 24

Запрос маршрута

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя  Назначение  Тип  Ссылки  Предусловия  Постусловия  Вывод | Запрос маршрута  Запрос маршрута  Системная  Вариант использования «Путеводители»  -  Осуществлен запрос маршрута  Осуществлен вывод маршрута |

### Построение диаграмм деятельностей сценариев вариантов использования

Диаграмма деятельности сценария варианта использования приведен на рис. 13.

Для перехода на страницу с путеводителями пользователю необходимо выбрать на сайте «Путеводители», после чего он попадет на новую страницу. Далее выбрать из списка интересующий его путеводитель, система обратиться к API карт и в случае успешной синхронизации выдаст карту с транспортом.

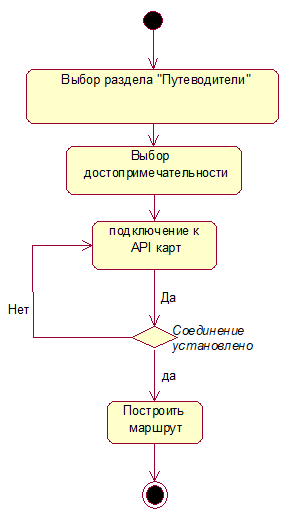


Рис.13. Диаграмма деятельности для ВИ «Путеводители»

Для просмотра маршрута пользователю необходимо выбрать путеводитель, затем достопримечательность. Система проверит доступ к картам и если соединение будет установлено, то выдаст маршрут.

## Проектирование системы

### Проектирование структуры системы и построение диаграмм пакетов

Диаграммы пакетов унифицированного языка моделирования отображают зависимости между пакетами, составляющими модель.

Диаграммы пакетов могут использовать пакеты, содержащие прецеденты для иллюстрации функциональности программного обеспечения системы. Диаграммы могут использовать пакеты, которые представляют различные слои программного комплекса для иллюстрации его слоистой архитектуры. Зависимости между этими пакетами могут быть снабжены метками / стереотипами, чтобы указать механизм связи между слоями [1].

Для разрабатываемой системы была построена диаграмма пакетов (рис.14).

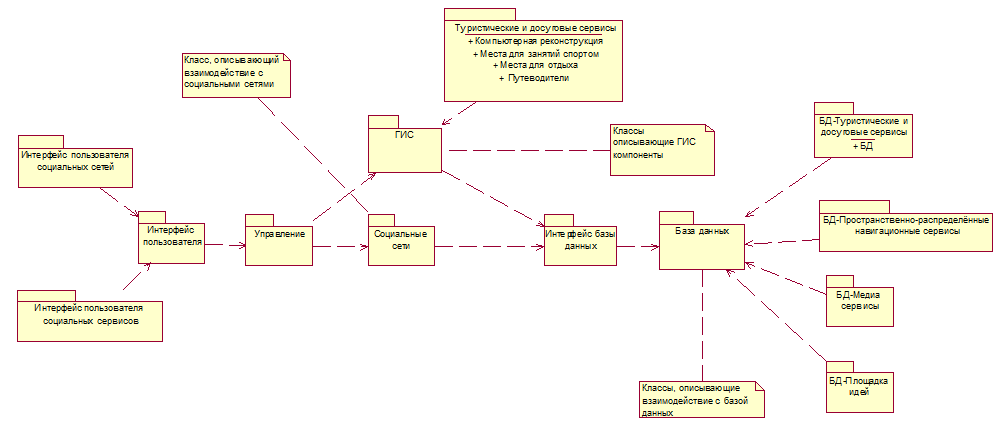


Рис.14. Диаграмма пакетов системы «Туристические и досуговые сервисы»

Интерфейс пользователя – классы, реализующие объекты интерфейса пользователя.

Управление – классы, реализующие связь между интерфейсом и классами, обеспечивающими функциональное выполнение действий.

ГИС – классы, описывающие ГИС компоненты

Интерфейс базы данных– классы, реализующие интерфейс для обращения к базе данных.

База данных – классы, описывающие взаимодействия с базой данных.

### Проектирование классов пакетов

#### Проектирование классов пакета “Туристические и досуговые сервисы”

##### Исходная диаграмма

Исходная диаграмма классов для функционального пакета приведена на рис.15.

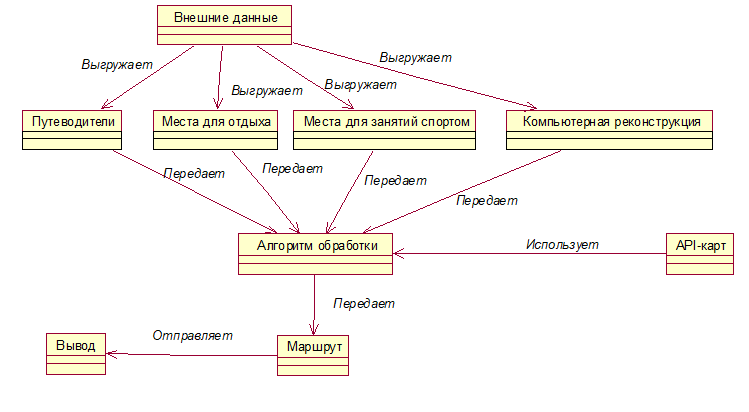


Рис.15. Исходная диаграмма классов

Описание классов:

* Класс «Внешние данные». Содержит данные которые передает пользователь системе;
* Класс «Путеводители». Содержит данные для обработки, которые были выгружены из внешних данных;
* Класс «Места для занятия спортом». Содержит данные для обработки, которые были выгружены из внешних данных;
* Класс «Места для отдыха». Содержит данные для обработки, которые были выгружены из внешних данных;
* Класс «Компьютерная реконструкция». Содержит данные для обработки, которые были выгружены из внешних данных;
* Класс «API-карт». Содержит открытые функции постройки карт;
* Класс «Алгоритм обработки». Содержит построения маршрута и анализа ГЕО данных, так же содержит обработчики API карт;
* Класс «Маршрут». Содержит строит маршрут по данным из класса «Алгоритм обработки»;
* Класс «Вывод». Содержит информацию для вывода на экран на основе данных маршрута.

##### Построение диаграмм последовательностей взаимодействий объектов класса

Диаграмма последовательности взаимодействий объектов класса для нормального хода событий приведена на рис.16.

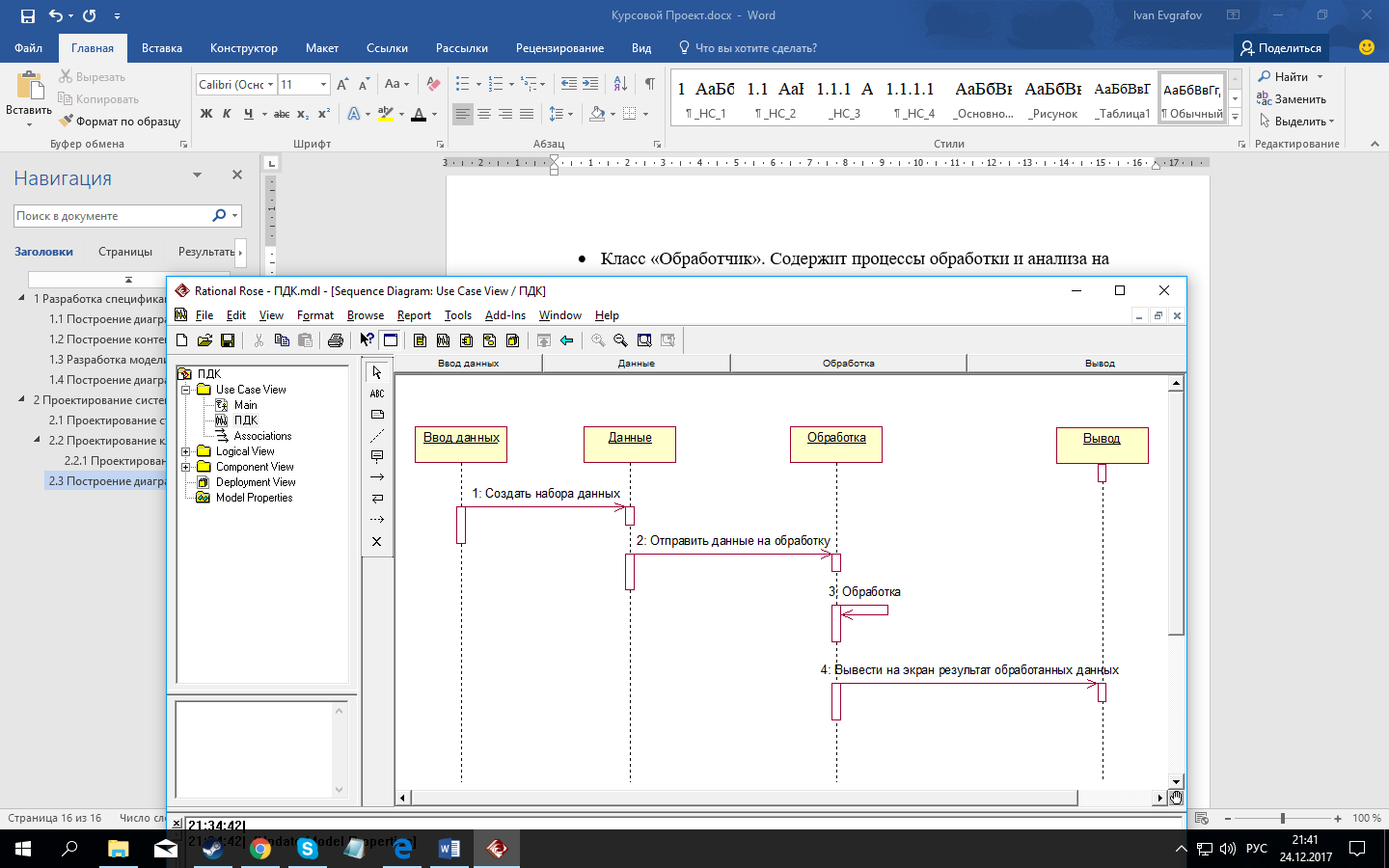


Рис.16. Диаграмма последовательности взаимодействий объектов класса для нормального хода событий

На основе данных, которые загружены в систему, происходит обработка и анализ данных. При нормальном ходе событий эти данные обрабатываются, выводятся на экран, и система не выдает никаких ошибок.

##### Построение диаграммы кооперации

Диаграммы кооперации предназначены для описания динамических аспектов моделируемой системы [2]. Обычно они применяются для того, чтобы:

* показать набор взаимодействующих объектов в реальном окружении;
* распределить функциональность между классами, основываясь на результатах изучения динамических аспектов системы;
* описать логику выполнения сложных операций, особенно в тех случаях, когда один объект взаимодействует еще с несколькими объектами;
* изучить роли, выполняемые объектами внутри системы, а также отношения между объектами, в которые они вовлекаются, выполняя эти роли.

Диаграмма кооперации показана на рис.17.

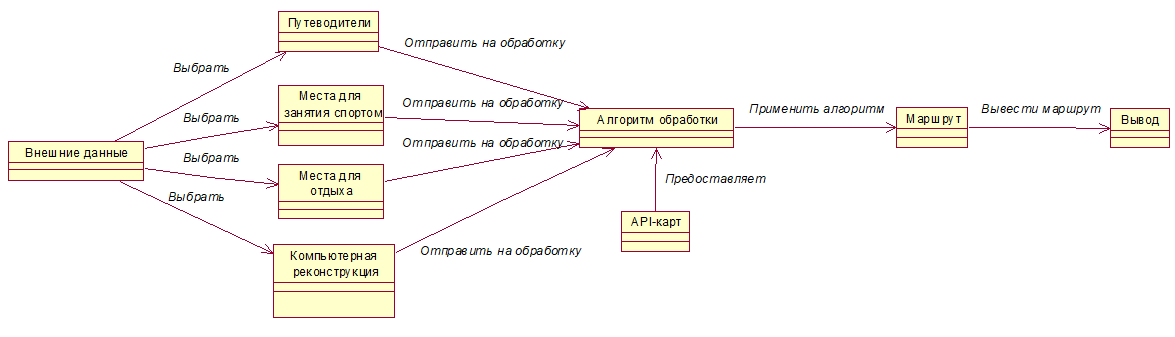


Рис.17.Диаграмма кооперации

Пользователя передает данные в классы «Путеводители», «Места для занятия спортом», «Места для отдыха», «Компьютерная реконструкция». При обработке, данные берутся из классов «Путеводители», «Места для занятия спортом», «Места для отдыха», «Компьютерная реконструкция», а алгоритмы обработки из класса «API-карт». После обработки и вовремя её данные записываются в «Маршрут». На основе данных класса «Маршрут» формируются и выводится конечный маршрут.

##### Построение уточненной диаграммы классов

Уточненная диаграмма классов, представленная на рис. 18

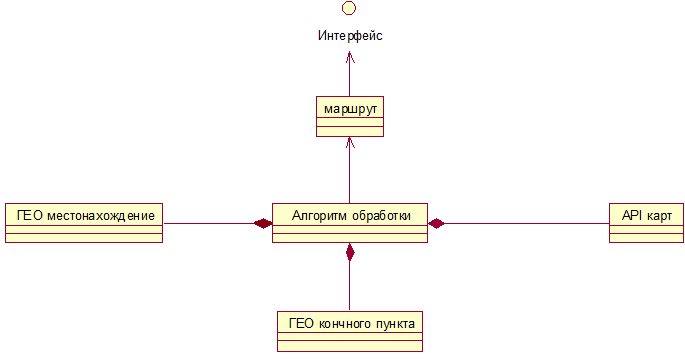


Рис.18. Уточненная диаграмма классов

Из диаграммы видно, что классы «API карт», «ГЕО местонахождения», «ГЕО конечного пункта» входят в состав класса «Алгоритм обработки».

##### Разработка детальной диаграммы классов

Детальная структура класса, представленная на рис. 19

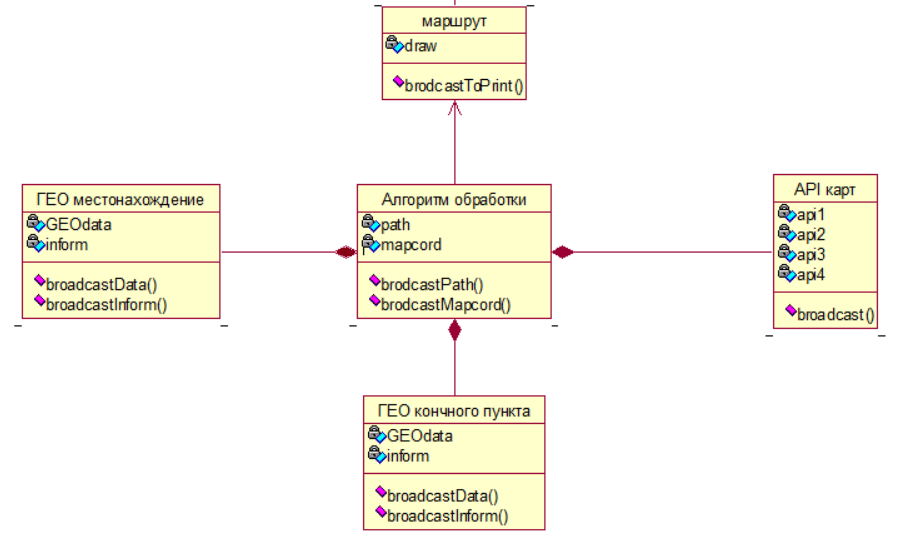


Рис.19. Детальная структура класса

* Класс «Маршрут». Содержит процессы для отображения маршрута и данные координат самого маршрута.
* Класс «Алгоритм обработки». Содержит данные для хранения пути и координат с карт, процессы для передачи пути и координат.
* Класс «API карт». Содержит открыты функции для работы с картами, и процесс их передачи.
* Класс «ГЕО местонахождения». Содержит ГЕО данные местонахождения, и процесс их передачи
* Класс «ГЕО конечного пункта». Содержит ГЕО данные конечного пункта, и процесс их передачи

### Построение диаграмм компонентов

Физическое представление системы показано при помощи диаграммы компонентов рис.20.

Краткие спецификации компонентов приведены в табл.42.

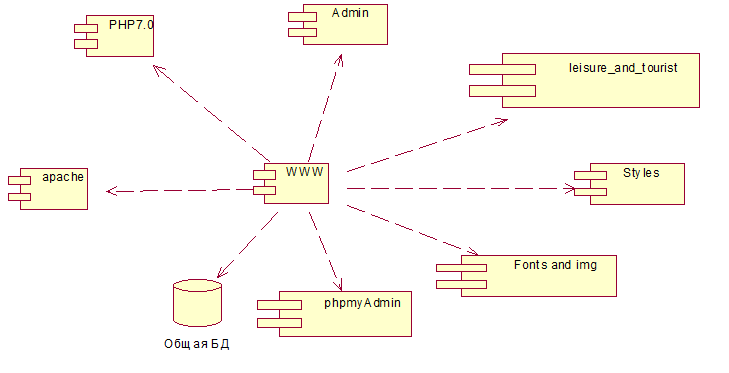


Рис.20. Диаграмма компонентов

Таблица 42

Краткие спецификации компонентов

|  |  |
| --- | --- |
| Название компонента | Описание |
| 1 | 2 |
| WWW | Коревой компонент сервиса, где хранятся все его компоненты |
| phpmyAdmin | Компонент веб интерфейса для работы с базой данных на сервисе |
| Fonts and img | Компонент хранящий шрифты и изображения |
| Styles | Компонент хранящий стили сервиса |
| leisure\_and\_tourist | Компонент хранящий основные классы и скрипты сервиса |

Таблица 42. Продолжение

|  |  |
| --- | --- |
| 1 | 2 |
| Общая БД | представляет собой хранилище, в котором находится все данные используемые для отображения на информационном портале. |
| Apache | веб сервер |
| PHP 7.0 | представляет собой компилятор 7 версии языка php |
| Libs | представляет собой сборник всех сторонних библиотек, которые использовались в разработке |
| Admin | представляет собой сборник классов для администрирования сервиса |

### Построение диаграмм размещения

Диаграмма размещения представлена на рис.21

Для организации работы системы необходим управляющий сервер, выполняющие устройства и сеть для их взаимодействия.

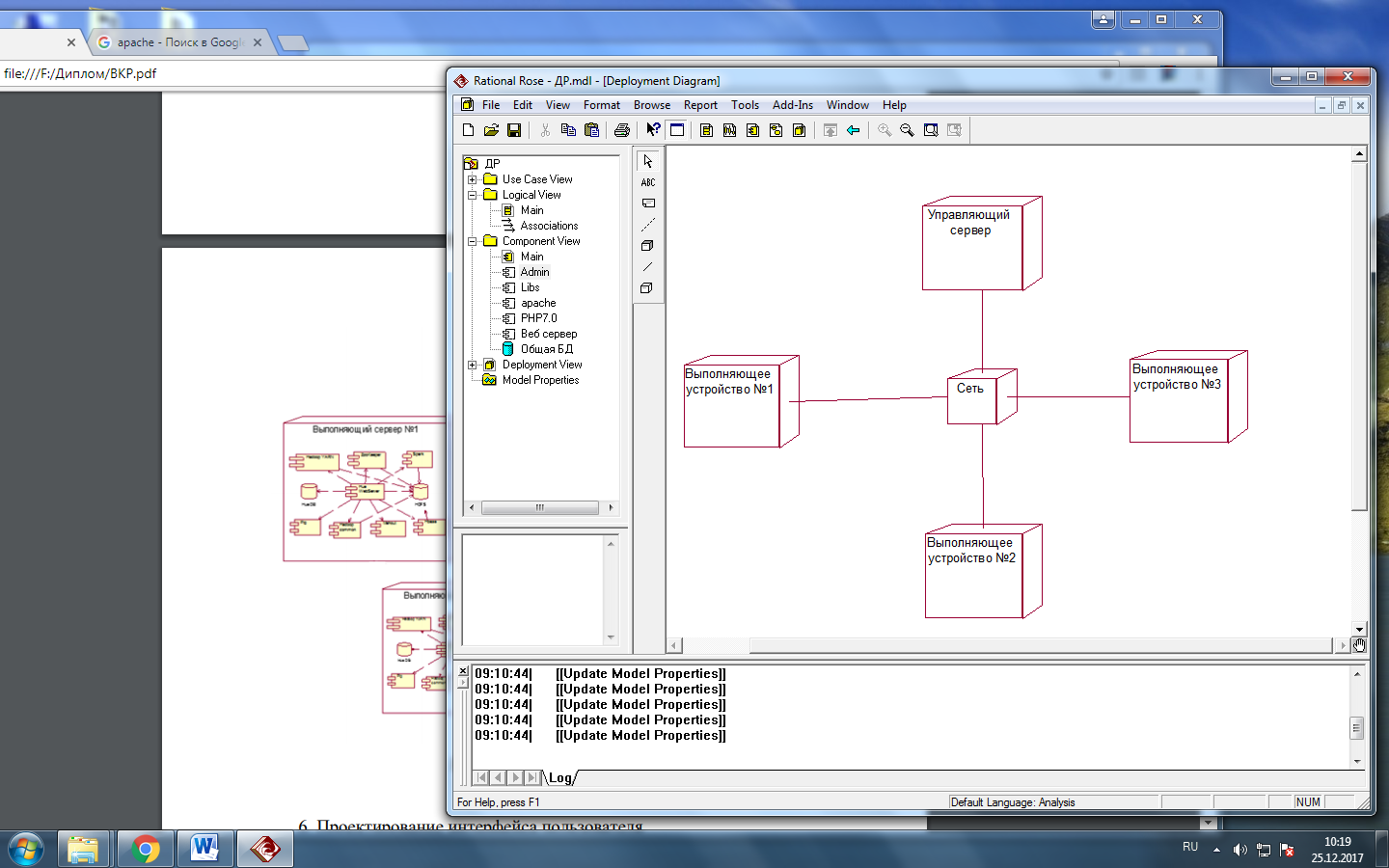


Рис.21. Диаграмма размещения

## Проектирование интерфейса пользователя

### Построение графа диалога

Граф диалога - ориентированный взвешенный граф, каждой вершине которого соответствует определенное состояние диалога, характеризующееся набором доступных пользователю действий. Дуги, исходящие из вершин, показывают возможные изменения состояний при выполнении пользователем указанных действий [5].

Интерфейс клиентской части информационного портала «Социополис 35» «Туристических и досуговых сервисы» содержит диалоговые окна, которые соответствуют вершинам графа диалога (рис. 6.1), описание которых представлено в табл. 6.1.

Таблица 6.1

Вершины графа диалога

|  |  |
| --- | --- |
| Вершина графа | Описание |
| Прибытие на главную страницу сервиса | Страница, содержащая описание главной страницы сервиса |
| Просмотр путеводителей (достопримечательностей) | Страница с описанием путеводителей и достопримечательностей |
| Просмотр мест для отдыха | Страница с описанием мест для отдыха |
| Просмотр мест для занятия спортом | Страница с описанием мест для занятия спортом |
| Просмотр маршрута | Страница, предоставляющая просмотр маршрута путеводителя |



Рис.6.1. Граф диалога «Социополис 35» «Туристических и досуговых сервисы»

## Выбор стратегии тестирования, разработка тестов, программа и методика испытаний

Под тестированием программного обеспечения понимается процесс выявления ошибок в программном обеспечении (ПО). В настоящее время существует несколько методов тестирования ПО, однако, даже они не позволяют однозначно и полностью установить корректность функционирования анализируемой программы. Исходя из этого, все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого ПО [3].

### Объект испытаний

Объектом тестирования является информационный портал «Социополис 35» подсистема «Туристических и досуговых сервисы».

### Цель испытаний

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий две различные цели [2]:

* продемонстрировать заказчику, что программное обеспечение

соответствует требованиям, которые заявлены в техническом задании;

* выявить ситуации, в которых поведение программы является

неправильным, нежелательным или несоответствующим спецификации;

* произвести проверку работы модулей и всей программы на различных

наборах входных данных.

### Требования к информационному, аппаратно-программному обеспечению и документации

#### Требования к функциональным характеристикам

На основании анализа требований к программному обеспечению составлено техническое задание на разработку и разработаны спецификации.

Программное обеспечение должно полностью удовлетворять функциональным и эксплуатационным требованиям, заявленным в техническом задании (прил.1).

#### Требования к надежности

Программное обеспечение должно удовлетворять всем заявленным в техническом задании требованиям к надежности.

В процессе работы программного обеспечения не допустима ситуация полного отказа и не допустима ситуация искажения обрабатываемой информации в процессе работы.

#### Требования к аппаратному и информационному обеспечению

Программа должна функционировать на устройствах с минимальными аппаратными требованиями, заявленными в техническом задании (прил.1) на разработку.

#### Требования к программной документации

В состав программной документации входит: расчетно-пояснительная записка с приложениями (Приложение 1 - Техническое задание, Приложение 2 - Спецификация, Приложение 3 - Текст программы, Приложение 4 - Руководство пользователя).

### Состав, порядок и методы испытаний

Программное обеспечение тестируется на устройствах с версией операционной системы и минимальными техническими требованиями, заявленными в техническом задании (прил.1).

- процессор с тактовой частотой 1400 MHz или более мощный;

- оперативная память 512 Мб или больше;

- архитектура с разрядностью 32 бит или 64 бит (x86 или x64);

- операционная система Windows 7, Windows 8, Windows 10.

- клавиатура;

- мышь;

- доступ в интернет.

#### Состав испытаний

Порядок проведения испытаний:

* проверка запуска программы на тестируемом устройстве;
* тестирование заявленных в техническом задании функциональных

требований на полноту и логику выполнения;

* тестирование интерфейса пользователя на удобство и понятность

использования;

* тестирование надежности;
* выявление ошибок на каждом из этапов;
* отладка программного обеспечения;
* контрольное тестирование.

#### Методы испытаний

Различают два подхода к формированию тестов: структурный и функциональный. Каждый из указанных подходов имеет свои особенности и области применения [5].

Структурный подход базируется на том, что известна структура тестируемого программного обеспечения, в том числе его алгоритмы («стеклянный ящик»). В этом случае тесты строят так, чтобы проверить правильность реализации заданной логики в коде программы.

Функциональный подход основывается на том, что структура программного обеспечения неизвестна («черный ящик»). В этом случае тесты строят, опираясь на функциональные спецификации. Этот подход называют также подходом, управляемым данными, так как при его использовании тесты строят на базе различных способов декомпозиции множества данных.

Наборы тестов, полученные в соответствии с методами этих подходов, обычно объединяют, обеспечивая всестороннее тестирование программного обеспечения.

Также на ранних стадиях разработки программного применяют методы ручного тестирования. Ручной контроль как правило позволяет обнаружить 30-70% ошибок. Исходные данные для таких проверок: техническое задание, спецификации, диаграммы классов, диаграммы компонентов, схемы отдельных компонентов, а на более поздних этапах- алгоритмы и тексты программ [5].

Для тестирования клиентской части приложения выбран метод ручного тестирования, для тестирования правильности выполнения скриптов серверной части приложения выбран метод граничных значений функционального тестирования [5].

### Результаты проведения испытаний

Результаты проведения испытаний, выполненных на основании указанных подходов, представлены в табл. 7.1-7.2.

Все выявленные в процессе тестирования неточности, ошибки работы, несоответствия спецификациям и требованиям технического задания (прил.1) программного обеспечения устранены.

Таблица 7.1

Результаты ручного тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата | Тестолог | Наименование  теста | Описание  теста | Результат тестирования |
| 10.01.2018 | Ершов Е.В. | Соответствие ТЗ | Соответствие выполненной работы к требованиям технического задания | 70% выполнения требований ТЗ |
| 15.01.2018 | Гришкина Ю.Е | Соответствие ТЗ | Соответствие выполненной работы к требованиям технического задания | 90% выполнения требований ТЗ |
| 16.01.2018 | Гришкин Д.Е | Соответствие ТЗ | Соответствие выполненной работы к требованиям технического задания | 100% выполнения требований ТЗ |

Таблица 7.2

Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата и время тестирования | Тестируемый модуль или подпрограмма | Кто проводил тестирование | Описание теста | Результаты тестирования |
| 1 | 2 | 3 | 4 | 5 |
| 13.02.18 | BD.php | Разработчик | Подключение к базе данных | Успех |
| 13.02.18 | BD.php | Разработчик | Создание конструктора | Успех |
| 15.02.18 | BD.php | Разработчик | Работа деструктора | Успех |
| 17.02.18 | guidebooks.php | Разработчик | Вывод списка путеводителей | Успех |
| 17.02.18 | guidebooks.php | Разработчик | Сортировка списка | Успех |
| 20.02.18 | guidebooks.php | Разработчик | Слияние стилей путеводителей и главной страницы | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Вывод путеводителя из базы данных | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Проверка полученных данных на ошибки | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Установка меток маршрута | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Объединение меток единый путь | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Отображение маршрута на карте | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Вывод информации в список | Успех |
| 20.02.18 | item-guidebooks.php | Разработчик | Перестройка маршрута | Успех |

# Технико-экономическое обоснование

## Трудозатраты на разработку и отладку программы

Исходные данные для расчета определяются с учетом сложности разработки, наличия аналогов, квалификации разработчиков:

* предполагаемое число операторов q = 10 000;
* коэффициент сложности разработки с = 1,6;
* коэффициент коррекции программы в ходе ее разработки р = 0,08;
* коэффициент квалификации разработчиков k = 0,8;
* коэффициент увеличения затрат труда вследствие недостатков описания задачи B = 1,3.

Условное число операторов Q в программе определяется по формуле:

где: – предполагаемое число операторов,

– коэффициент сложности разработки,

– коэффициент коррекции программы в ходе ее разработки.

Q= 100001,6 (1 + 0,07) = 17120

Затраты труда на изучение описания задачи (с учетом квалификации программиста) определяются по формуле:

где: – условное число операторов,

– коэффициент квалификации разработчиков,

– коэффициент увеличения затрат труда вследствие недостатков описания задачи.

tи =17120\*1,3/(76\*0,8) = 366 чел./ч.

Затраты труда на разработку алгоритма решения задачи определяются по формуле:

где: – условное число операторов,

– коэффициент квалификации разработчиков.

ta = 17120/(22\*0,8) = 972,7 чел./ч.

Затраты труда на программирование по готовой блок-схеме определяются по формуле:

где: – условное число операторов,

– коэффициент квалификации разработчиков.

tп = 17120/(23\*0,8) = 930,5 чел./ч.

Затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи определяются по формуле:

где: – условное число операторов,

– коэффициент квалификации разработчиков.

17120/(4\*0,8) = 5350

При комплексной отладке задачи затраты труда на отладку программы на ЭВМ определяются по формуле:

где: – затраты труда на отладку программы на ЭВМ при автономной отладке.

tкотл = 5350\*1,5 = 8025 чел./ч.

Затраты труда на подготовку материалов в рукописи определяются по формуле:

где: – условное число операторов,

– коэффициент квалификации разработчиков.

tдр = 17120/(17\*0,8) = 1258,8 чел./ч.

Затраты труда на редактирование, печать и оформление документации определяются по формуле:

где: – затраты труда на подготовку материалов в рукописи.

tдо = 0,75\*1258,8 = 944,1 чел./ч

Затраты труда на подготовку документации по задаче определяются по формуле:

где: – затраты труда на отладку программы на ЭВМ при автономной отладке.

tд = 1258,8 + 944,1 = 2202,9. чел./ч.

Трудоемкость разработки программного обеспечения решения задачи рассчитывается по формуле:

где:– затраты труда на изучение описания задачи,

– затраты труда на разработку алгоритма решения задачи,

– затраты труда на программирование по готовой блок-схеме,

– затраты труда на подготовку документации по задаче.

t = 366 + 972,7 + 930,5 + 5350+ 2202,9 = 9822,1 чел./ч.

## Расчет себестоимости программного продукта

Состав разработчиков: программист-дипломник, руководитель ВКР.

Затраты на оплату труда при разработке программного продукта вычисляются по формуле:

где – общая зарплата работника за час;

– отчисления с зарплаты, %;

– время написания программы

Заработная плата дополнительная определяется по следующей формуле:

где – заработная плата;

– норма отчислений на дополнительную зарплату (10%).

Зарплата общая вычисляется по следующей формуле:

где – заработная плата;

– заработная плата дополнительная.

Отчисления на соцстрах, фонд занятости и пенсионный фонд вычисляются по следующей формуле:

где – отчисления на соцстрах (0,5% от ),

– отчисления в фонд занятости (0,5% от ),

– отчисления в пенсионный фонд (2% от ).

Затраты на оплату труда руководитель ВКР

111 руб.

111\*11,1/100=13 руб.

111+13=124 руб.

124\*0.01/2+124\*0.01/2+124\*0.02=4 руб.

=8\*66=528 ч.

=(124+4)\*528=67 584 руб.

Затраты на оплату труда программист-диплом­ник

150 руб.

150\*15/100=23 руб.

150+23=173 руб.

173\*0.01/2+173\*0.01/2+173\*0.02=5,2 руб.

=8\*66=528 ч.

=(173+5,2)\*528=94090 руб.

Стоимость одного часа машинного времени рассчитывается по формуле:

где – стоимость одного часа машинного времени,

– покупная цена компьютера, руб.,

– срок службы компьютера, год,

– количество рабочих дней в году,

– время работы компьютера в течение суток,

– стоимость одного кВт\*ч электроэнергии,

– мощность вычислительной системы.

(50000/3\*247\*8)+4,25\*0,25=9,49 руб/ч.

Время использования вычислительной техники рассчитывается по следующей формуле:

где – время использования вычислительной техники,

– количество дней разработки ПО.

Затраты на использование машинного времени вычисляются по формуле:

где – затраты на использование машинного времени,

– стоимость одного часа машинного времени,

– время использования вычислительной техники.

=9,49\*528=5011 руб.

Затраты на носители информации принимаются в размере 2 % от цены вычислительной техники:

Затраты на текущий и профилактический ремонт принимаются в размере 4 % от цены вычислительной техники. ,

Себестоимость разработки рассчитывается по формуле:

где – затраты на оплату труда при разработке программного продукта,

– затраты на использование машинного времени,

– затраты на носители информации,

– Затраты на текущий ремонт,

– затраты на текущий и профилактический ремонт.

(67 584+94090) +5011\*2+1000\*2+2000\*2+5000\*2=187696 руб.

## Расчет цены программного продукта

Цена программного продукта рассчитывается по формуле:

где – себестоимость разработки,

– норматив прибыли.

=187696\*(1+0,2) = 225235 руб.

## Расчет экономической эффективности

Расходы потребителя, связанные с эксплуатацией программы, определяются по следующей формуле:

где – эксплуатационные расходы потребителя,

– объем машинного времени в течение года, необходимый для решения данной задачи с использованием программы,

– стоимость одного часа машинного времени,

– цена программного продукта,

– срок службы программного продукта.

=1235\*9,49+225235/2=124337 руб.

Капитальные затраты на вычислительную технику рассчитываются по формуле:

где – цена вычислительной техники, руб.;

– прочие расходы потребителя

=50000+5000=55000 руб.

Капитальные затраты рассчитываются по формуле:

где – объем машинного времени в течение года, необходимый для решения данной задачи с использованием программы,

– капитальные затраты на вычислительную технику,

– полезный годовой фонд времени работы вычислительной техники,

– цена программного продукта.

=(1235\*55000/2000)+ 225235=259198 руб.

Эксплуатационные затраты вычисляются по формуле:

где – фонд заработной платы персонала, обслуживающего решение задачи вручную.

=1,21\*20000\*12=290400 руб.

Годовая экономия эксплуатационных расходов у одного потребителя рассчитывается по формуле:

где – Эксплуатационные затраты,

– расходы потребителя, связанные с эксплуатацией программы.

290400-124337=166063 руб.

Срок окупаемости программного продукта рассчитывается по формуле:

где – капитальные затраты,

– годовая экономия эксплуатационных расходов у одного потребителя.

259198/166063=1,56 г.

Годовой экономический эффект, получаемый одним пот­ре­бителем, рассчитывается по формуле:

где – годовая экономия эксплуатационных расходов у одного потребителя,

– нормативный коэффициент эффективности дополнительных капитальных вложений,

– капитальные затраты.

ЭЭ=166063-0,15\*259198=127184 руб.

# Заключение

В ходе выполнения данной курсовой работы были разработаны спецификации и спроектирована подсистема «Туристических и досуговых сервисов» информационного портала «Социополис35». Были разработаны модель использования, концептуальная модель предметной области, модель процессов, модель реализации, модель развертывания.

Данная разработка создана чтобы помочь людям искать информацию о путеводителях, досуговых и культурно-массовых мероприятиях, достопримечательностях, местах для занятия спортом и туризмом. Так же помочь людям с составлением маршрутов по местам и путеводителям, тем самым помогая экономить их время.

В процессе выполнения задания были сформулированы следующие компетенции:

* + - 1. Владение навыками использования различных технологий разработки программного обеспечения (ПК-3);
      2. Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе, роли людей, процессов, методов, инструментов и технологий обеспечения качества (ПК-4);
      3. Владение стандартами и моделями жизненного цикла (ПК-5);
      4. Владение классическими концепциями и моделями менеджмента в управлении проектами (ПК-6);
      5. Владение основными концепциями и моделями эволюции и сопровождения программного обеспечения (ПК-10);
      6. Владение особенностями эволюционной деятельности, как с технической точки зрения, так и с точки зрения бизнеса (работа с унаследованными системами, возвратное проектирование, реинжиниринг, миграция и рефакторинг) (ПК-11);
      7. Способность к формализации в своей предметной области с учетом ограничений используемых методов исследования (ПК-12);
      8. Готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности (ПК-13);
      9. Готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности (ПК-14);
      10. Способность формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта (ПК-16);
      11. Способность выполнить начальную оценку степени трудности, рисков, затрат и сформировать рабочий график (ПК-17).

# Список литературы

1. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. / А.М. Вендров. - М.: Финансы и статистика, 2009 г. – 758 с.
2. Гома, Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений: Пер. с англ. – М.: ДМК Пресс.– 704с.
3. Бахтизин, В. В. Технология разработки программного обеспечения : учеб. пособие [Текст] / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2010. – 267 с. : ил.
4. Диаграмма компонентов [Электронный ресурс] – Электрон. текстовые дан. –Режим доступа: www.wikipedia.org/Диаграмма\_компонентов, свободный;

# Приложение 1. Техническое задание

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО Череповецкий государственный университет

Институт информационных технологий

Кафедра математического и программного обеспечения ЭВМ

Дисциплина «Технология разработки программного обеспечения»

УТВЕРЖДАЮ

Зав. кафедрой МПО ЭВМ,

д. т.н., профессор \_\_\_\_\_\_\_\_\_\_ Ершов Е.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**«**Разработка программного обеспечения информаци-

онного портала "Социополис 35": "Туристический и досуговый сервисы"**»**

Техническое задание на курсовой проект

Листов \_\_\_

Руководитель:

Ершов Е.В.,

директор института

Выполнил:

студент гр. 1ПИб-01-41оп

Гришкин Д.Е.

Череповец,

2018 г.

Введение

Содержанием курсового проекта является разработка спецификаций и проектирование информационной системы «Социополис35» подсистемы «Туристических и досуговых сервисов».

Область применения системы – использование в информационных порталах для сервисов, связанных с туризмом и отдыхом.

Основание для разработки

Основанием для разработки служит задание на курсовую работу по модулю: "Спецификация, архитектура и проектирование программного обеспечения".

Назначение разработки

Разрабатываемое программное обеспечение необходимо информационному порталу города, для предоставления информации о путеводителях, мест для занятия спортом, мест для отдыха и туризма, а также для предоставления маршрутов путеводителей.

Требования к программе

1. Требования к функциональным характеристикам.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1. Составление маршрутов для путешествий

2. Составление путеводителей по городу

3. Предоставление информации о спортивных мероприятиях

4. Предоставление информации о культурно-массовых мероприятиях

1. Требования к надежности

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

1. Выделение помещения под оборудование.

2. Организация бесперебойного питания оборудования.

3. Обеспечение взаимодействие GPS и ГЛОНАСС.

4. Обеспечение защиты от SQL-инъекции.

5. Обеспечение защиты от CSRF-уязвимости.

6. Обеспечение защиты от XSS или «Межсайтовый скриптинг».

7. Обеспечение защиты незащищенных прямых ссылок.

1. Требования к составу и параметрам технических средств

- Процессор с тактовой частотой 1400 MHz или более мощный;

- Оперативная память 512 Мб или больше;

- Архитектура с разрядностью 32 бит или 64 бит (x86 или x64);

- Операционная система Windows 7, Windows 8, Windows 10.

- Клавиатура

- Мышь

- Доступ в интернет

1. Требования к информационной и программной совместимости

Для разработки клиентской используется Html Pug и препроцессор Css Sass, а также jQuery.

Для сборки проекта будет использоваться система сборки Gulp.

Для работы с PHP, jQuery, Sass и Pug используется программа PhpStorm.

1. Требования к программной документации.

В состав программной документации входит: расчетно-пояснительная записка с приложениями (Приложение 1 - Техническое задание, Приложение 2 - Спецификация, Приложение 3 - Текст программы, Приложение 4 - Руководство пользователя).

Стадии и этапы разработки

Таблица П1.1.

Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапа разработки ПО | Сроки разработки | Результат выполнения | Отметка о выполнении |
| 1 | 2 | 3 | 4 |
| Разработка технического задания | 12.01.18 – 22.01.18 | Техническое задание | Выполнил  20.01.2018 |
| Осуществление сравнительного анализа отечественных и зарубежных аналогов проектируемой системы | 22.01.18 – 6.02.18 | Сравнительный анализ аналогов проектируемой системы | Выполнил  04.02.2018 |
| Выбор технологии, среды и языка программирования | 6.02.18 – 12.02.18 | Выбраны технологии, среды и язык программирования | Выполнил  12.02.2018 |
| Анализ процесса обработки информации, выбор структур данных для её хранения | 12.02.18 – 20.02.18 | Осуществлен анализ процесса обработки информации, выбрана структуры данных для её хранения | Выполнил  20.02.2018 |

Таблица П1.1. Продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Разработка спецификаций проектируемой системы | 20.02.18 – 1.03.18 | Разработаны спецификации проектируемой системы | Выполнил  29.02.2018 |
| Проектирование системы | 1.03.18 – 20.03.18 | Спроектирована система | Выполнил  20.03.2018 |
| Проектирование интерфейса пользователя | 20.03.18 – 1.04.18 | Спроектированный пользовательский интерфейс | Выполнил  29.03.2018 |
| Выбор стратегии тестирования, разработка тестов, прохождение тестов | 1.04.18 – 10.04.18 | Выбрана стратегия тестирования, разработаны тесты | Выполнил  10.04.2018 |
| Написание руководства пользователя | 10.04.18 – 18.04.18 | Руководство пользователя | Выполнил  18.04.2018 |
| Защита курсового проекта | 19.04.18 | Защищенный курсовой проект | Выполнил  22.04.2018 |

Порядок контроля и приемки

Порядок контроля и приемки производится в соответствии со стадиями и

этапами разработки (см. табл. П1.1).

# Приложение 2

Спецификация

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Примечание |
| Расчетно-пояснительная  записка | «Разработка программного обеспечения информаци-  онного портала "Социополис 35": "Туристический и досуговый сервисы"» |  |
| Приложение 3 | Текст программы |  |
| Компоненты | | |
| Index.html | Файл структуры главной страницы |  |
| main\_global.css | Файл стилей главной страницы |  |
| leisure\_and\_tourist.css | Файл стилей сервиса |  |
| leisure\_and\_tourist.php | Файл серверных скриптов сервиса |  |

# Приложение 3

Текст программы

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8"/>

<base href="/"/>

<title>Home</title>

<meta name="viewport" content="width=device-width"/>

<link rel="stylesheet" media="all" href="/css/main\_global.css"/>

</head>

<body>

<header>

<div class="header-content">

<p>#социополис<span>35</span></p>

<div class="weather">

<p>Череповец, -4&#8451;</p>

<p>30 ноября, 19:15</p>

</div>

</div>

<div class="menu-content">

<ul>

<li> <a href="/">Главная</a></li>

<li> <a href="/page/page2.html">О проекте</a></li>

<li> <a href="/page/page2.html">Контакты</a></li>

<li> <a href="/page/page2.html">Поиск</a></li>

</ul>

<div class="mobile-menu-content">

<div aria-hidden="true" class="menu-icon fa fa-bars fa-2x"></div>

<ul class="m\_menu-content">

<li> <a href="/">Главная</a></li>

<li> <a href="/page/page2.html">О проекте</a></li>

<li> <a href="/page/page2.html">Контакты</a></li>

<li> <a href="/page/page2.html">Поиск</a></li>

</ul>

</div>

<p class="logo">#социополис<span>35</span></p>

</div>

</header>

<div class="scrolltotop"> </div>

<section class="slider">

<div class="slider-content">

<div id="goToPrevSlide" class="prev"></div>

<div id="goToNextSlide" class="next"></div>

<ul id="lightSlider">

<li><img src="img/slide1.png"></li>

<li><img src="img/slide1.png"></li>

<li><img src="img/slide1.png"></li>

</ul>

<div class="slider-inf">

<div class="left"><span>Череповцу 240 лет</span>

<p>куда сходить на День города</p>

</div>

</div>

</div>

</section>

<section class="platform\_ideas">

Рис.П3.1. Модуль «index.html»

П3.1. Продолжение

<div class="platform\_ideas-content"><a href="/page/page2.html" class="ideas-title"><span>Площадка идей</span></a>

<div class="ideas-content"><a href="/page/page2.html" class="content-items"><img src="img/1.png">

<p>Проекты</p></a><a href="/page/page2.html" class="content-items"><img src="img/2.png">

<p>Голосование</p></a><a href="/page/page2.html" class="content-items last"><img src="img/3.png">

<p>Доброволь-<br>чество</p></a></div>

</div>

</section>

<section class="sections">

<div class="sections-content">

<div class="sections-title"> Всё самое нужное на одном портале</div>

<div class="sections-slider">

<div id="goToPrevSlide2" class="prev"></div>

<div id="goToNextSlide2" class="next"></div>

<ul id="lightSlider2">

<li>

<div class="slid-content"><a href="/page/navigation/mainNavig.html" class="content-items"><img src="img/5.png">

<p>Транспорт</p></a><a href="/page/page2.html" class="content-items"><img src="img/2.png">

<p>Новости</p></a><a href="/page/page2.html" class="content-items"><img src="img/3.png">

<p>Афиши</p></a><a href="/page/page2.html" class="content-items"><img src="img/6.png">

<p>Юридическая помощь</p></a><a href="/page/page2.html" class="content-items"><img src="img/7.png">

<p>Доступная среда</p></a><a href="/page/leisure\_and\_tourist/home-service.html" class="content-items three"><img src="img/1.png">

<p>Туризм</p></a></div>

</li>

<li>

<div class="slid-content"><a href="/page/page2.html" class="content-items"><img src="img/8.png">

<p>Экология</p></a><a href="/page/page2.html" class="content-items"><img src="img/9.png">

<p>Компьютерная реконструкция</p></a><a href="/page/page2.html" class="content-items"><img src="img/4.png">

<p>Медицина</p></a></div>

</li><li>

<div class="slid-content"><a href="/page/page2.html" class="content-items"><img src="img/5.png">

<p>Транспорт</p></a></div>

</li></ul></div></div></section>

<footer class="footer">

<div class="footer-content">

<div class="soc"><a href="https://vk.com/sociopolis35" aria-hidden="true" class="fa fa-vk fa-2x vk"></a><a href="#" aria-hidden="true" class="fa fa-facebook fa-2x facebook"></a><a href="http://www.thepictaram.club/instagram/chsu\_media" aria-hidden="true" class="fa fa-instagram fa-2x instagram"></a><a href="#" aria-hidden="true" class="fa fa-twitter fa-2x twitter"></a></div>

<div class="footer-menu">

<ul>

<li><a href="/">Главная</a></li>

<li><a href="/page/page2.html">О проекте</a></li>

<li><a href="/page/page2.html">Контакты</a></li>

</ul></div> </div>

</footer>

<script src="/js/all.js"></script>

<script src="/js/main.js"></script>

</body>

</html>

<?php

class Database

{

// private $db\_host = 'localhost';

// private $db\_user = 'o9052991\_socio';

// private $db\_pass = 'ptq9g57y7';

// private $db\_name = 'o9052991\_socio';

private $db\_host = 'localhost';

private $db\_user = 'root';

private $db\_pass = '';

private $db\_name = 'socio35';

private $con = false;

private $result0 = array();

public function connect()

{

if(!$this->con)

{

$myconn = mysql\_connect($this->db\_host,$this->db\_user,$this->db\_pass);

if($myconn)

{

$seldb = mysql\_select\_db($this->db\_name,$myconn);

if($seldb)

{

$this->con = true;

return true;

}

else

{

return false;

}

}

else

{

return false;

}

}

else

{

return true;

}

}

/\*public function close\_connect(){

if(mysql\_close())

{

$this->con = false;

$this->results0 = null;

}

}\*/

public function setDatabase($name)

{

if($this->con)

{

Рис.П3.2. Модуль «db.php»

П3.2. Продолжение

if(mysql\_close())

{

$this->con = false;

$this->results0 = null;

$this->db\_name = $name;

$this->connect();

}

}

}

private function tableExists($table)

{

$tablesInDb = mysql\_query('SHOW TABLES FROM '.$this->db\_name.' LIKE "'.$table.'"');

if($tablesInDb)

{

if(mysql\_num\_rows($tablesInDb)==1)

{

return true;

}

else

{

return false;

}

}

}

public function select($table, $rows = '\*', $where = null, $order = null)

{

$q = 'SELECT '.$rows.' FROM '.$table;

if($where != null)

$q .= ' WHERE '.$where;

if($order != null)

$q .= ' ORDER BY '.$order;

$query = mysql\_query($q);

if($query)

{

$this->numResults = mysql\_num\_rows($query);

for($i = 0; $i < $this->numResults; $i++)

{

$r = mysql\_fetch\_array($query);

$key = array\_keys($r);

for($x = 0; $x < count($key); $x++)

{

if(!is\_int($key[$x]))

{

if(mysql\_num\_rows($query) > 1)

$this->result0[$i][$key[$x]] = $r[$key[$x]];

else if(mysql\_num\_rows($query) < 1)

$this->result0 = null;

else

$this->result0[$key[$x]] = $r[$key[$x]];

}

П3.2. Продолжение

}

}

return true;

}

else

{

return false;

}

}

public function insert($table,$values,$rows = null)

{

if($this->tableExists($table))

{

$insert = 'INSERT INTO '.$table;

if($rows != null)

{

$insert .= ' ('.$rows.')';

}

for($i = 0; $i < count($values); $i++)

{

if(is\_string($values[$i]))

$values[$i] = '"'.$values[$i].'"';

}

$values = implode(',',$values);

$insert .= ' VALUES ('.$values.')';

$ins = mysql\_query($insert);

if($ins)

{

return true;

}

else

{

return false;

}

}

}

public function delete($table,$where = null)

{

if($this->tableExists($table))

{

if($where == null)

{

$delete = 'DELETE '.$table;

}

else

{

$delete = 'DELETE FROM '.$table.' WHERE '.$where;

}

$del = mysql\_query($delete);

if($del)

{

return true;

П3.2. Продолжение

}

else

{

return false;

}

}

else

{return false; }}

public function update($table,$rows,$where,$condition)

{

if($this->tableExists($table))

{

for($i = 0; $i < count($where); $i++)

{

if($i%2 != 0)

{

if(is\_string($where[$i]))

{

if(($i+1) != null)

$where[$i] = '"'.$where[$i].'" AND ';

else

$where[$i] = '"'.$where[$i].'"';

}

}

}

$where = implode($condition,$where);

$update = 'UPDATE '.$table.' SET ';

$keys = array\_keys($rows);

for($i = 0; $i < count($rows); $i++)

{

if(is\_string($rows[$keys[$i]]))

{$update .= $keys[$i].'="'.$rows[$keys[$i]].'"'; }

else

{ $update .= $keys[$i].'='.$rows[$keys[$i]]; }

if($i != count($rows)-1)

{

$update .= ',';

}

}

$update .= ' WHERE '.$where;

print $update ;

$query = mysql\_query($update);

if($query)

{

return true;

}

else

{return false; }}

else

{

return false;

}

}

public function getconn()

{return $this->con; }

public function getResult()

{return $this->result0;

}

}