*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***к курсовому проекту по дисциплине   
"Технологии программирования"***

***на тему***

*Проектирование и разработка программной системы   
информационной системы*

*«Помощник путешественника по Владимиру»*

*Выполнил: Емельянов В. А.*

*Принял: доц. кафедры ИСПИ*

*Вершинин В.В.*

*Владимир, 2022*

**<утвержденный Лист задания нА курсовой проект>**

**Аннотация**

Аннотация составляется на русском и английском языках. Это краткая характеристика работы, ее содержимого. В конце указывается свойства документа (пояснительной записки).

Курсовой проект представлен на 56 страницах, рисунков – 5, таблиц – 11, использованных источников – 4, приложений – 2, иллюстрационный материал на 3 листах формата А1.

**Содержание**

# ВЕДЕНИЕ

В ходе выполнения курсового проекта требовалось разработать программную систему информационной системы «помощник путешественника по Владимиру»

Целью курсового проектирования является изучение процесса разработки сложных систем на примере конкретной программной системы, определяемой тематикой курсового проекта.

Для достижения поставленной цели необходимо решить следующие задачи:

* изучение особенностей конкретной предметной области, относящихся к теме курсового проекта;
* анализ возможных подходов и методов решения с обоснованием выбранного подхода;
* разработка моделей (математической, структурной, информационной, данных) программной системы;
* выбор алгоритмов с учетом их точности и спецификой предметной области;
* разработка программного обеспечения;
* анализ полученных результатов работы ПО.

# Описание предметной области

* 1. Анализ

Основополагающей функцией программной системы - интерактивная динамическая карта, на которой расположены различные объекты для посещения, например, достопримечательности, заведения, интересные места. Управлением каждого объекта занимаются конкретные люди – владельцы или уполномоченные лица. У каждой точки интереса присутствует система отзывов и пожеланий. В веб-приложении помимо динамической карты есть поиск. Реализована система авторизации пользователей. У владельцев заведений разработан функционал добавления точек интереса при помощи заполнения формы с данными и контактами. После модерирования объект добавляют на сайт. Для каждого объекта авторизованный путешественник может создать беседу для совместного посещения данного заведения. Беседа создается автоматически в мессенджере Telegram. Приглашение происходит по ссылке. Для улучшения основного аккаунта до владельца заведения необходимо заполнить форму с данными, которую в последствии рассмотрит и подтвердит администратор системы. У каждого пользователя будет личный кабинет, где в зависимости от типа аккаунта будут свои возможности.

* 1. Функционал системы.

Функции системы

Программная система «Помощник путешественника по Владимиру» выполняет:

* сбор, актуализацию и хранение данных об объектах города Владимир;
* предоставление удобного взаимодействия с динамической картой;
* предоставление системы модерирования собственных объектов, заведений, интересных мест;
* предоставление возможности создание встреч для совместного посещения.
  1. Пользователи разрабатываемой системы

1) Путешественник – человек, авторизовавшийся в программной системе. Присутствует возможность оставлять отзывы к заведениям, создавать встречи для совместного посещения. Есть возможность улучшить аккаунт до владельца.  
2) Гость – неавторизованный путешественник системы. Ограничен функционал взаимодействия с системой. Возможен только просмотр объектов  
3) Владелец – особый пользователь системы, подтвержденный модераторами сервиса. По мимо функций обычного пользователя присутствует возможность добавления и модерирование своих заведений.  
4) Администратор – человек, в чьи обязанности входит отслеживание новых заявок по заведениям, модерирование отзывов, слежение за выполнением правил сервиса

* 1. Словарь предметной области  
     1) Личный кабинет – особый раздел на сайте, где пользователи могут просматривать, редактировать и в зависимости от типа аккаунта выполнять различные действия.  
     2) Точка интереса (объект, заведение) – место, которое можно посетить одному или с участниками группы, ранее созданной в мессенджере Telegram. У каждого объекта есть свой рейтинг и отзывы для более информативного ознакомления.

3) Встреча – мероприятие для совместного посещения объекта или заведения. Привязывается к конкретной дате и к конкретному месту. Для удобства общения, организации и сбора людей создается автоматически беседа в мессенджере Telegram и выдается ссылка для вступления.

**4)Интерактивная карта –** динамическая карта, на которой будут расположены различные точки для посещения. По нажатию на точку можно будет перейти в ее детальное описание.

**5) Отзыв –** словесное описание от посещенного объекта или заведения. Может быть удален администратором.

# проектирование системы

* 1. **ДИАГРАММЫ UML**
     1. **Диаграмма прецедентов**

В ходе проектирования программной системы была разработана диаграмма прецедентов, представленная на рисунке 1.

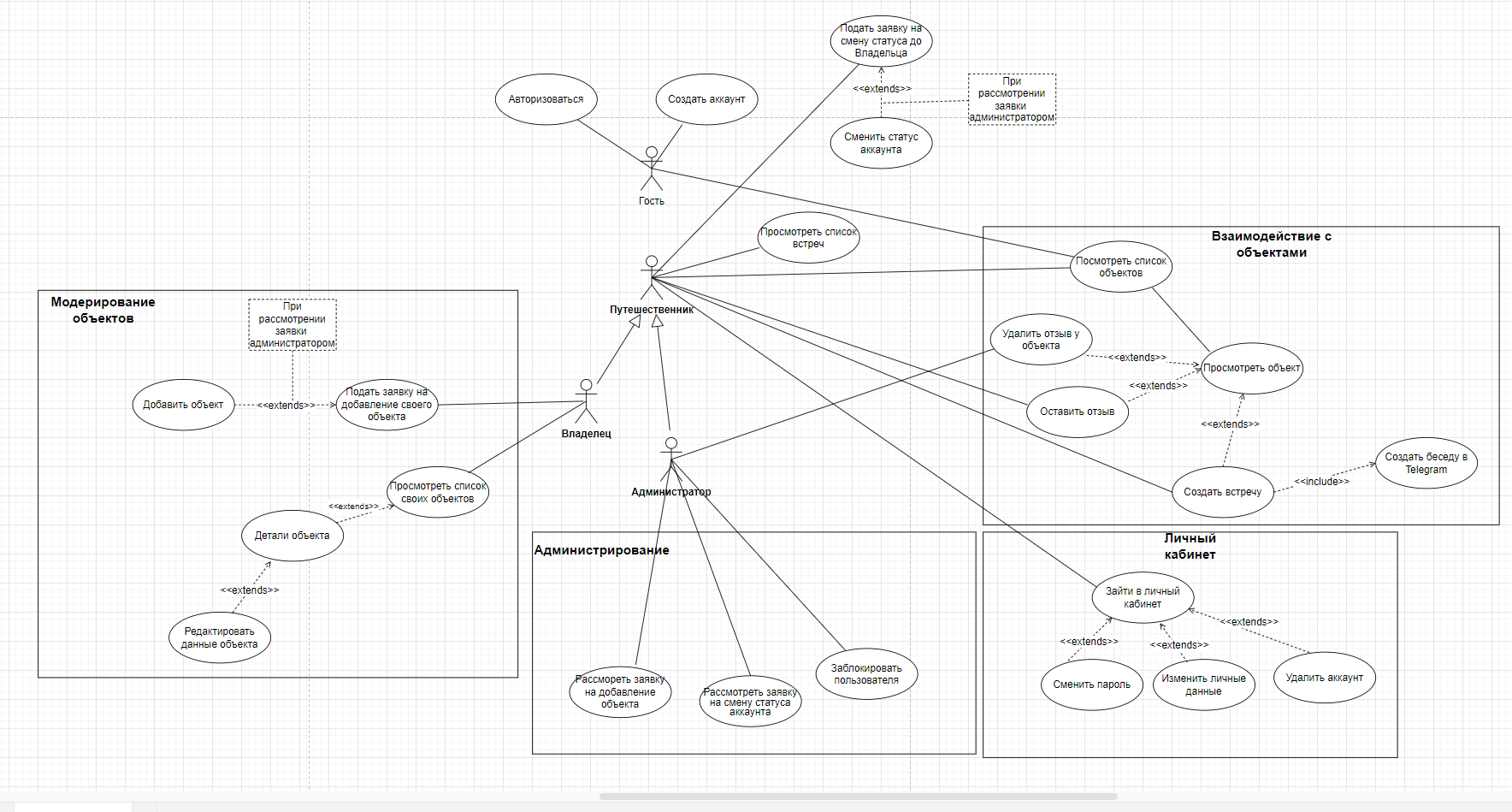


Рисунок №1. Диаграмма прецедентов.

* + 1. **Описание прецедентов**

*Название:* Создать встречу

*Актер:* Путешественник или владелец

*Предусловие:* Человек авторизовался в системе и выбрал объект из общего списка

*Основной поток:* Создание сущности встреча

Выполняется создание встречи и производится привязывание этой сущности к создавшему пользователю.

*Альтернативный поток:* Ошибка обработки запроса сервером.

*Постусловие:* Создание беседы в мессенджере Телеграм и предоставлении ссылки на вступление.

*Название*: Оставить отзыв

*Актер:* Путешественник или Владелец

*Предусловие:* Участник системы авторизовался и выбрал конкретный объект или заведение и перешел в его детали.

*Основной поток:* Написание отзыва.

Путешественник выражает свои мысли, эмоции и оценку от посещенного мероприятия.

*Альтернативный поток:* Ошибка обработки запроса сервером.

*Постусловие:* Обновление текущей страницы сайта

*Название*: Подать заявку на смену статуса до владельца

*Актер:* Путешественник

*Предусловие:* Участник системы авторизован

*Основной поток:* Подача заявки на смену статуса.

Путешественник, захотевший сменить свой статус аккаунта до владельца для добавления своих объектов или заведений, подает необходимую юридическую информацию и отправляет заявку на рассмотрение администратором.

*Альтернативный поток:*

Ошибка обработки запроса сервером;

Отклонение заявки.

*Постусловие:* Переадресация человека на главную страницу сайта.

*Название*: Подать заявку на добавление своего объекта.

*Актер:* Владелец

*Предусловие:* Участник системы авторизован

*Основной поток:* Подача заявки на добавление объекта.

Владелец, решивший добавить свой объект или заведение, заполняет особую форму с данными и посылает заявку на рассмотрение администратором

*Альтернативный поток:*

Ошибка обработки запроса сервером;

Отклонение заявки.

*Постусловие:* Переадресация человека на главную страницу сайта.

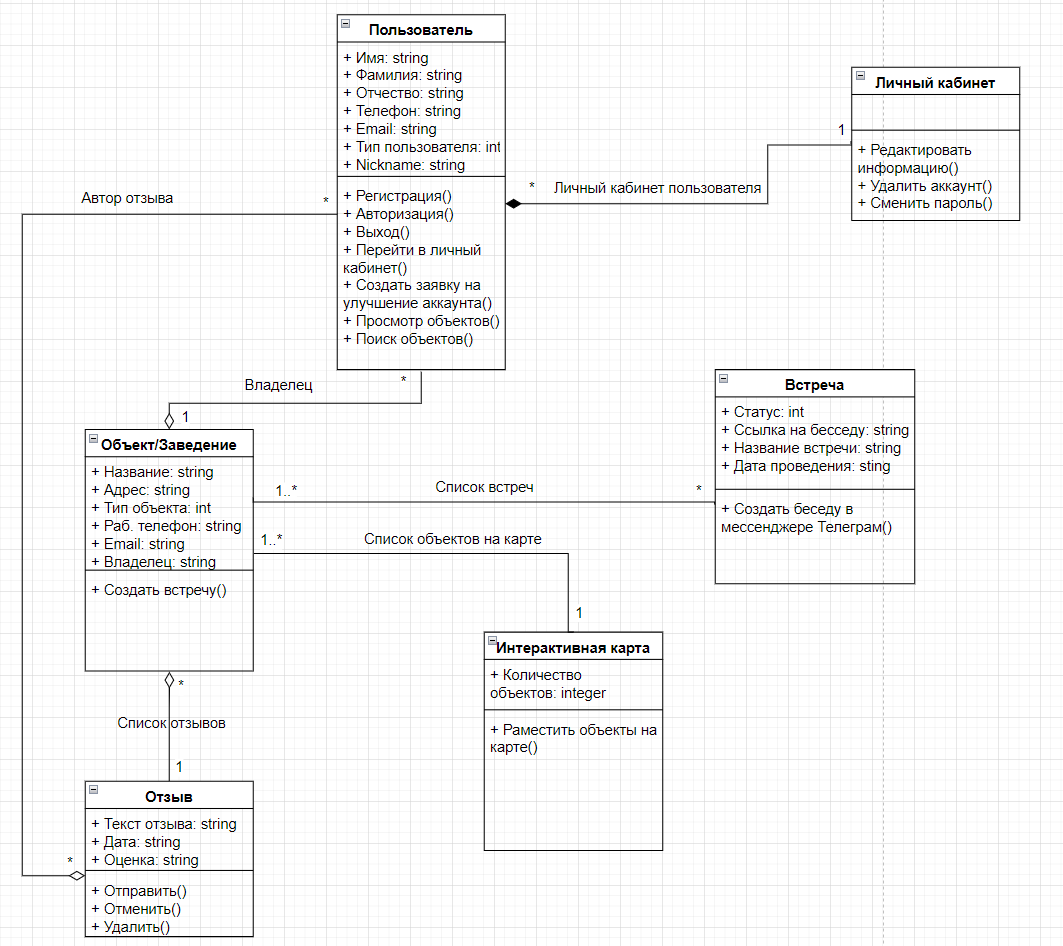
* 1. ** Диаграмма классов**В ходе проектирования системы была разработана диаграмма классов на рисунке 2.

Рисунок №2. Диаграмма классов.

* 1. **Диаграмма последовательностей.**

В ходе проектирование системы были разработаны основные диаграммы последовательности, описывающие основные взаимодействия между объектами системы и актерами. Данные диаграммы представлены на рисунках 3, 4, 5.

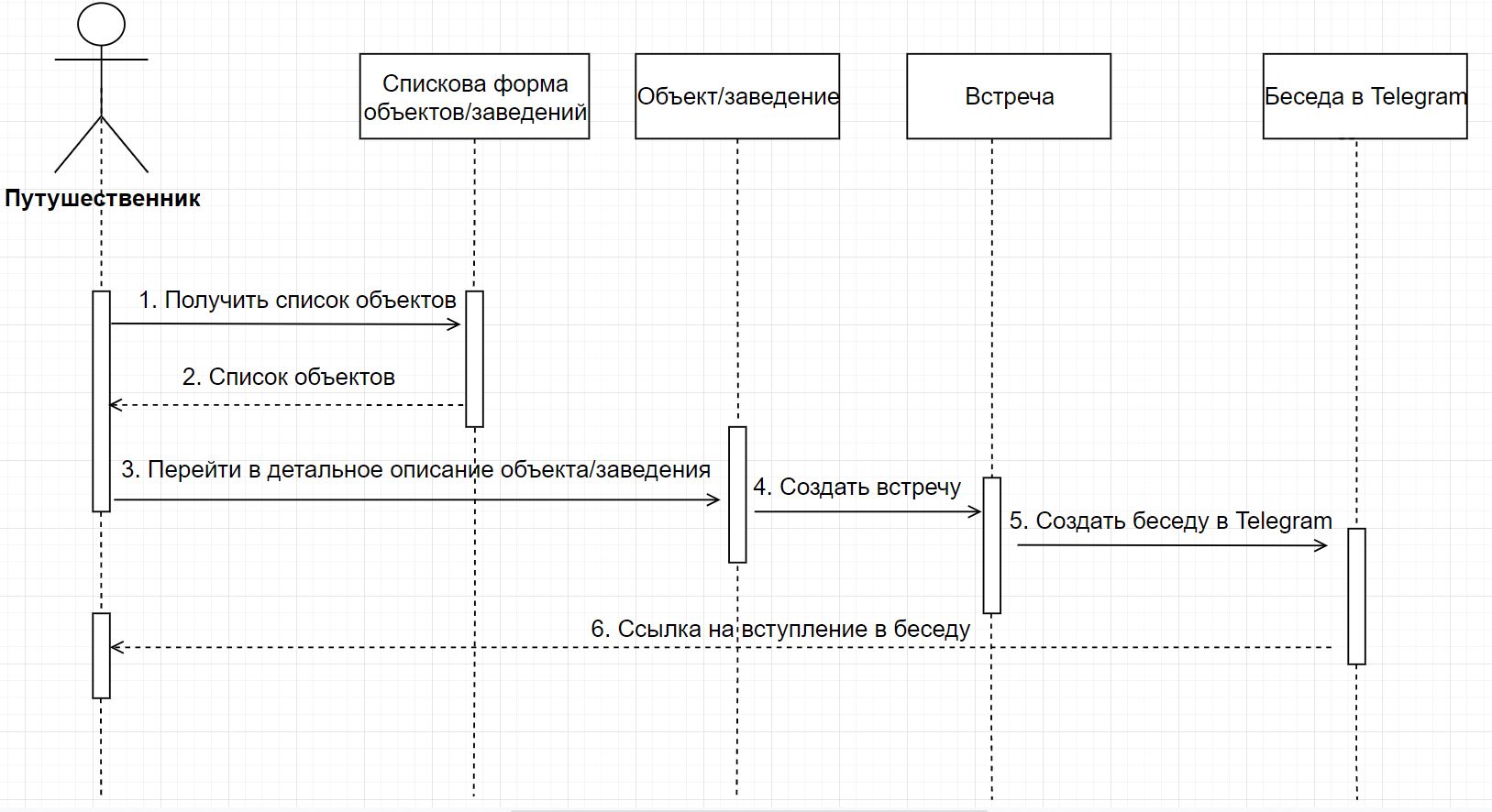


Рисунок №3. Диаграмма последовательности для прецедента

«Создать встречу».

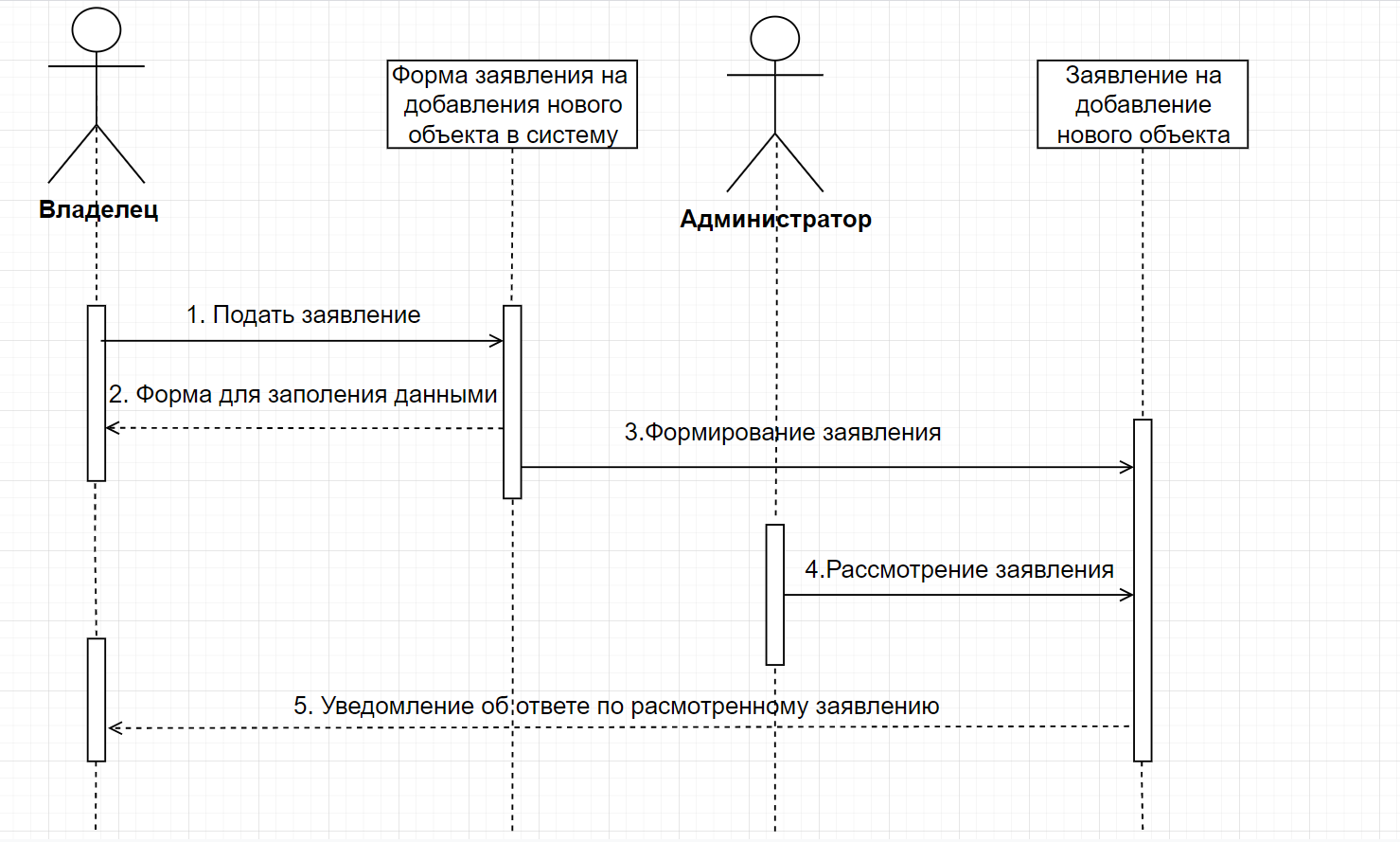
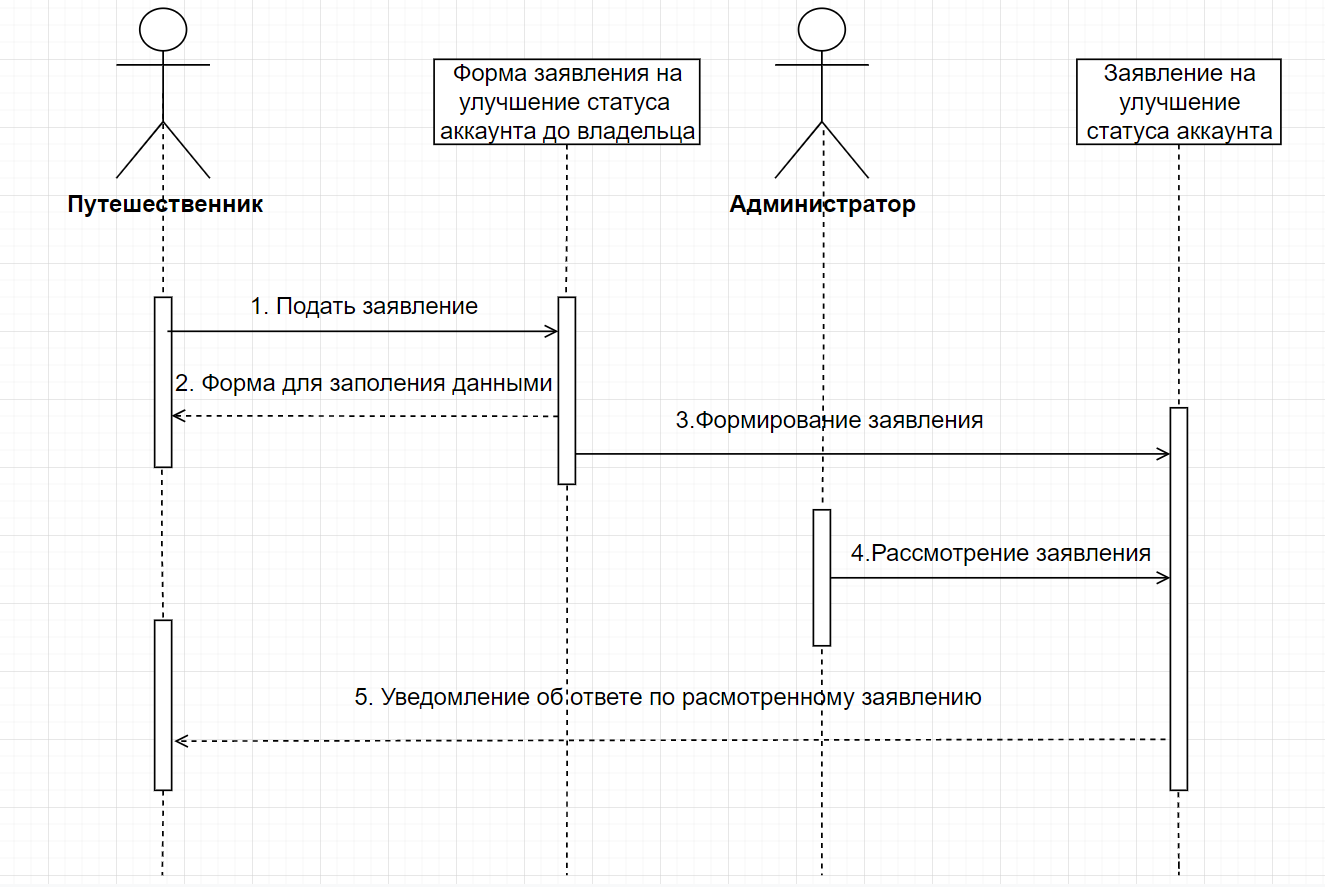


Рисунок №4. Диаграмма последовательности для прецедента

«Подать заявку на добавление своего объекта».

Рисунок №5. Диаграмма последовательности для прецедента

«Подать заявку на смену статуса до Владельца».

* 1. **Диаграмма состояний.**

В ходе проектирование системы были разработаны диаграммы состояний, описывающие жизненные циклы объектов системы. Данные диаграммы представлены на рисунках 6, 7.

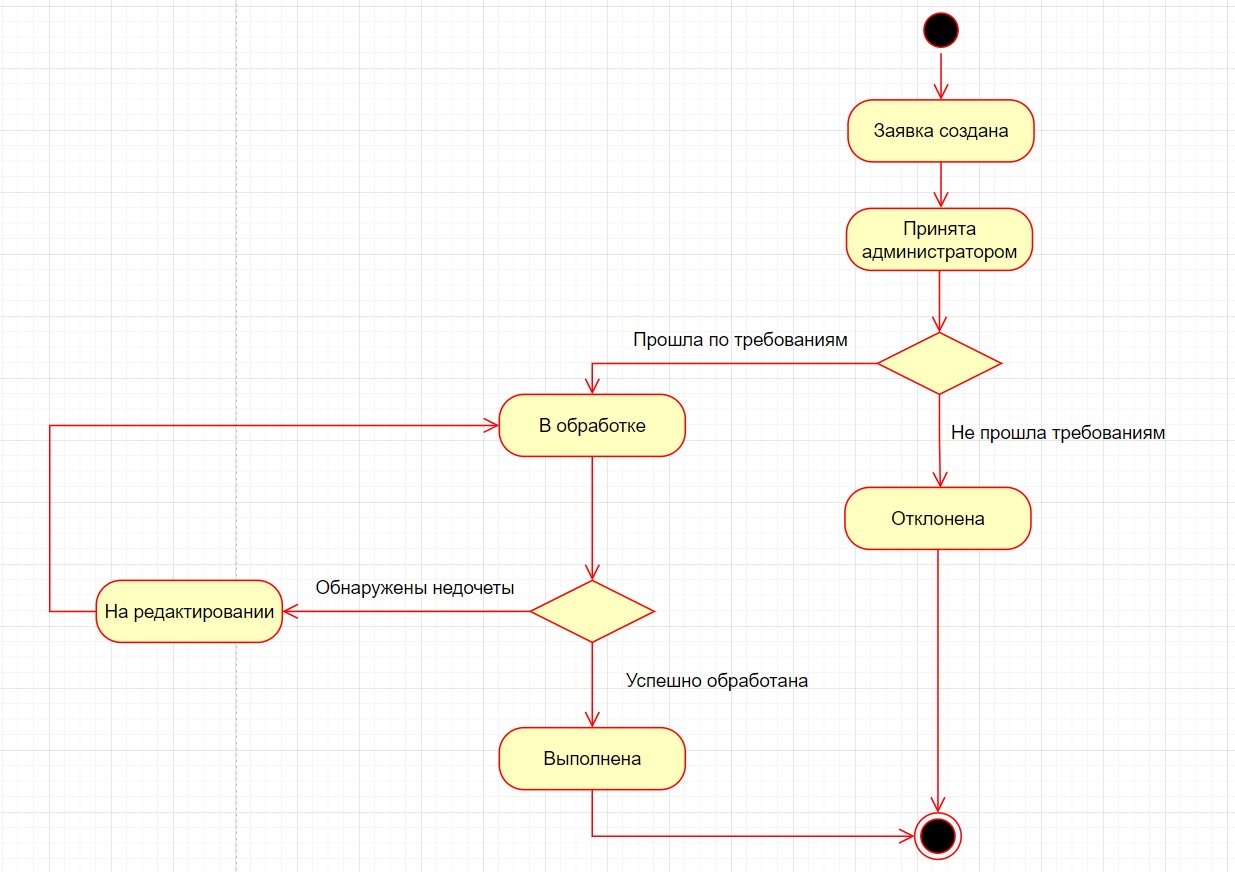


Рисунок №6. Диаграмма состояний, описывающая жизненный цикл

объекта «Заявка на добавление нового объекта в систему» и

объекта «Заявка на улучшение статуса аккаунта до Владельца».

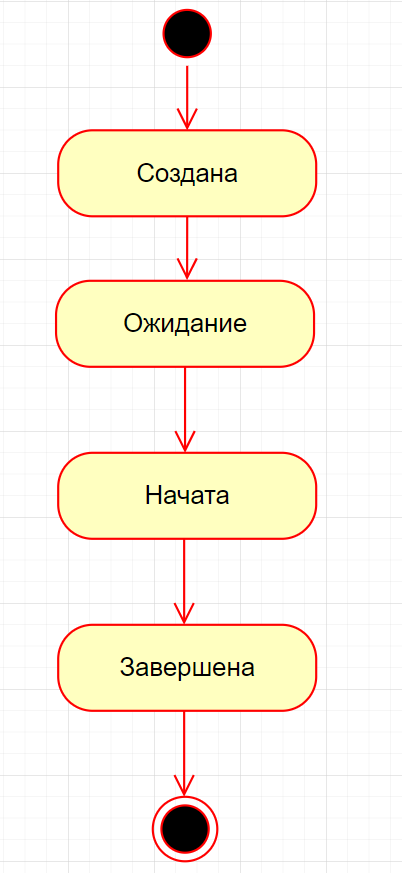


Рисунок №7. Диаграмма состояний, описывающая жизненный цикл

объекта «Встреча».

* 1. **ER-диаграммы.**

В ходе проектирования системы реализована ER-диаграмма, представленная на рисунке 8.

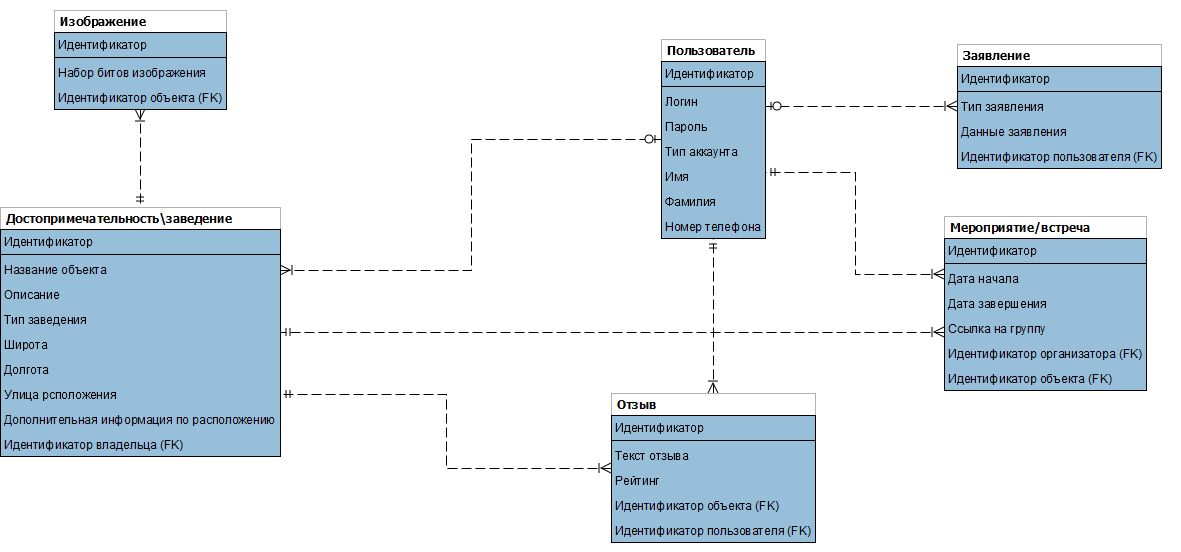
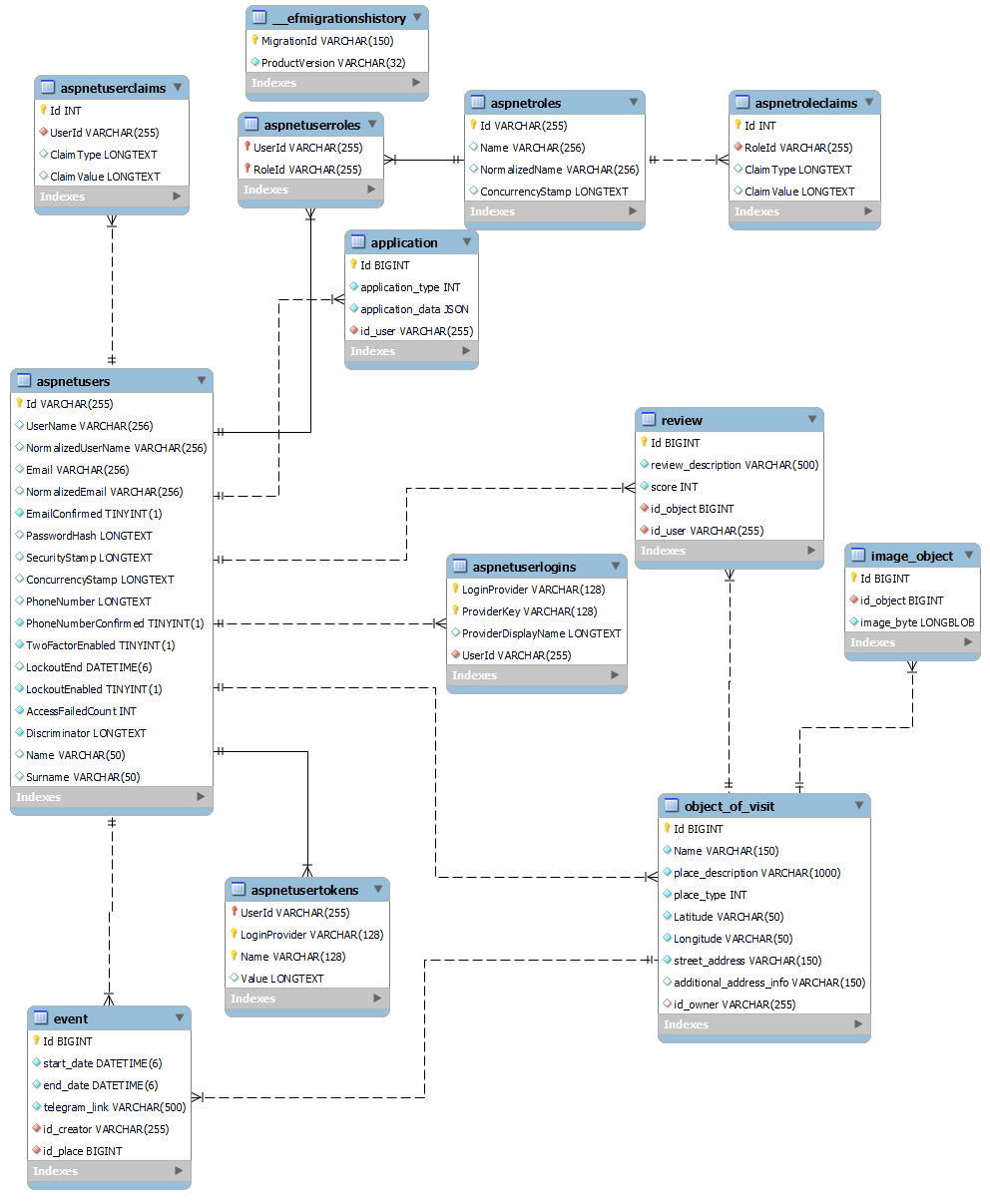


Рисунок №8. ER-диаграмма логического уровня.

Также была разработана предварительная ER-диаграмма физического уровня, представленная на рисунке 9.

Рисунок №9. ER-диаграмма физического уровня

* 1. **Скриншоты программной системы.**

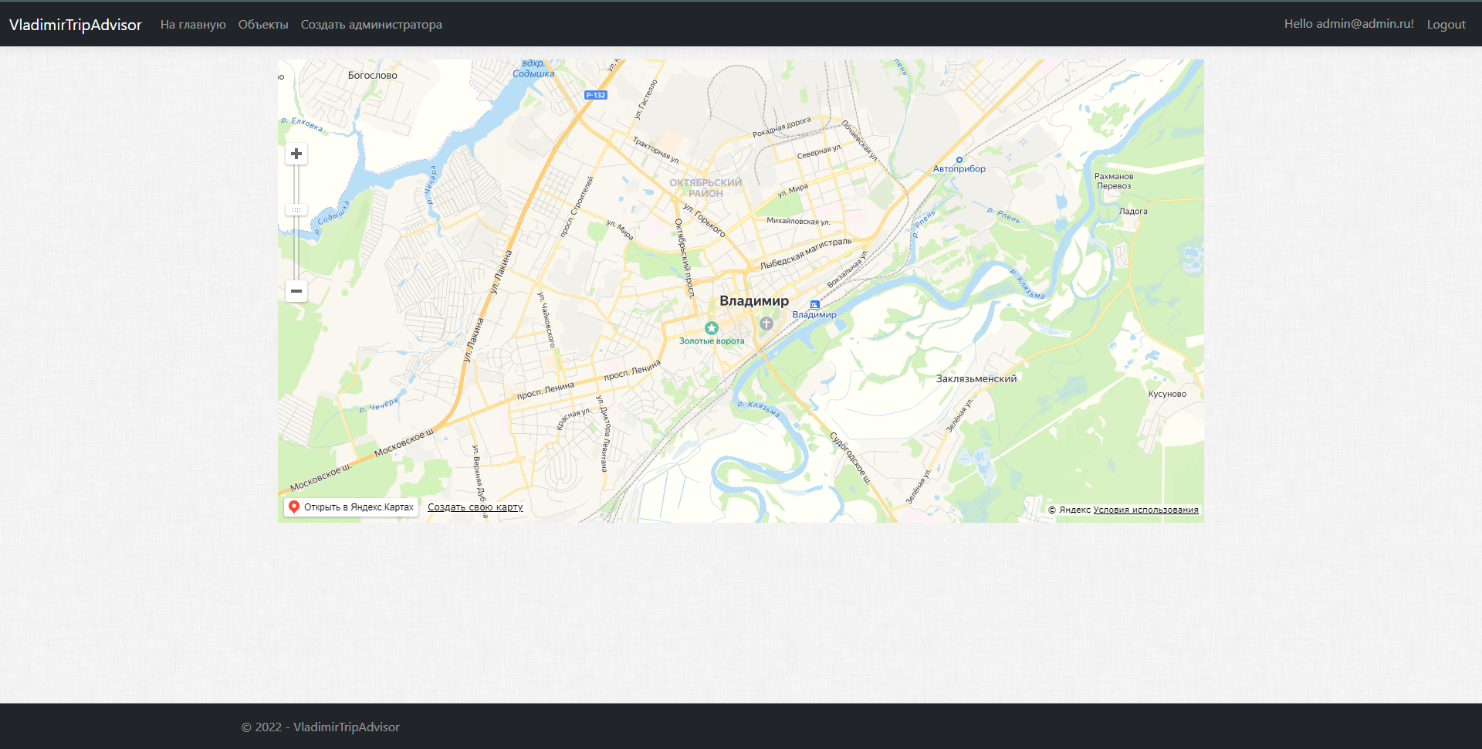
На рисунках 10 – представлены прототипы пользовательского интерфейса сайта программной системы.

Рисунок №.10 Главная страница.

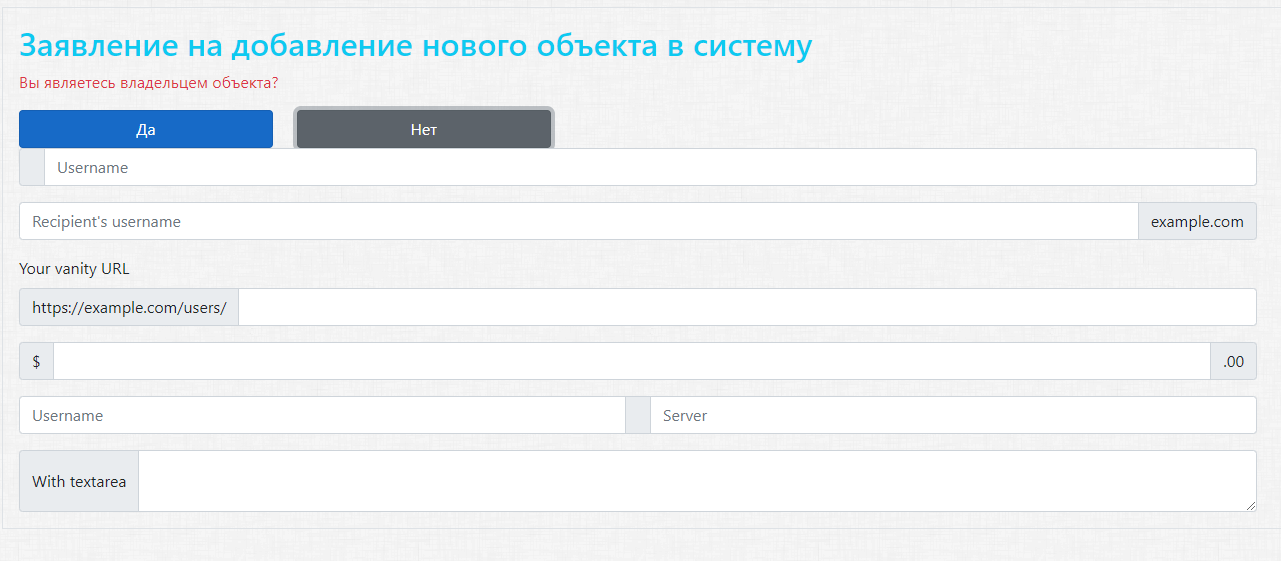


Рисунок №11. Тестовая форма с заявкой на добавление нового объекта.

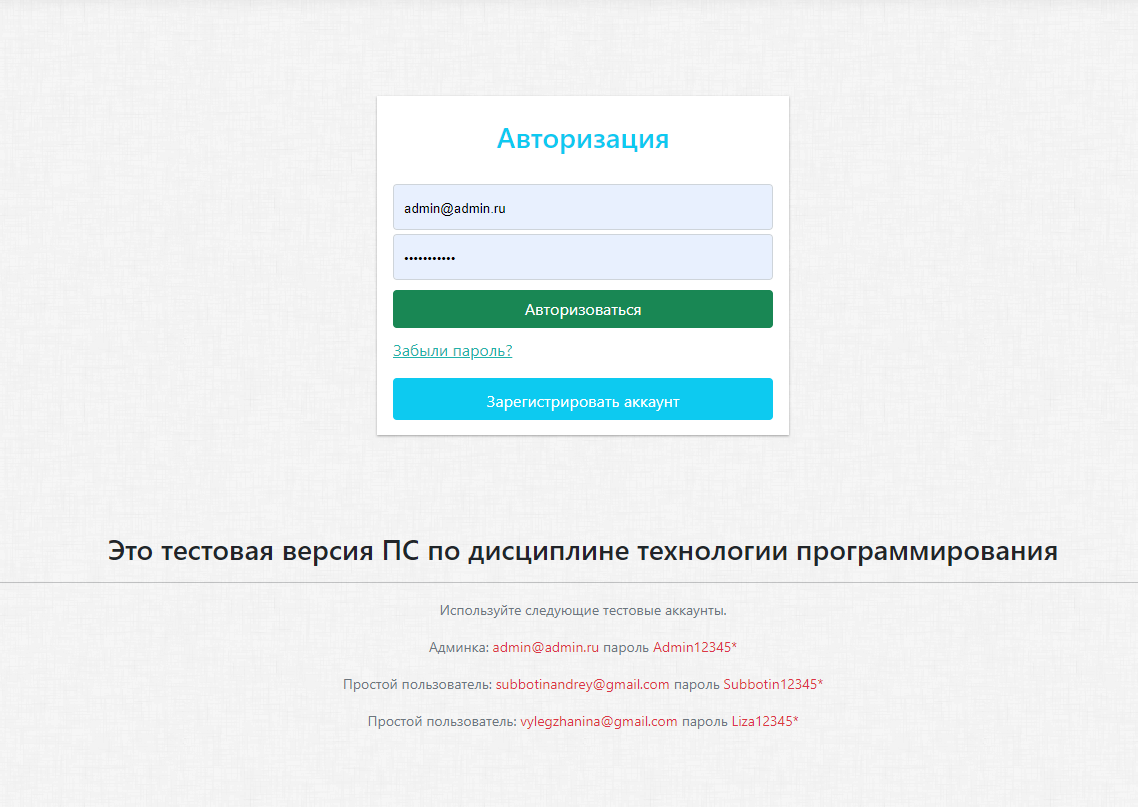


Рисунок №11. Окно авторизации.

* 1. **Фрагмент кода программной системы.**

ymaps.ready(init);

function init() {

var myMap = new ymaps.Map('map', {

center: [56.13, 40.39],

zoom: 14

}),

objectManager = new ymaps.ObjectManager({

// Чтобы метки начали кластеризоваться, выставляем опцию.

clusterize: true,

// ObjectManager принимает те же опции, что и кластеризатор.

gridSize: 32,

clusterDisableClickZoom: true

});

// Чтобы задать опции одиночным объектам и кластерам,

// обратимся к дочерним коллекциям ObjectManager.

objectManager.objects.options.set('preset', 'islands#greenDotIcon');

objectManager.clusters.options.set('preset', 'islands#greenClusterIcons');

myMap.geoObjects.add(objectManager)

myMap.controls.remove('searchControl');

myMap.controls.remove('geolocationControl');

myMap.controls.remove('trafficControl');

myMap.controls.remove('fullscreenControl');

myMap.controls.remove('rulerControl');

myMap.controls.remove('typeSelector');

$.ajax({

url: "/ymaps/objects.json"

}).done(function (data) {

objectManager.add(data);

});

}

# заключение

Основные выводы по работе и достигнутые результаты. Достаточно трех четырех абзацев текста.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Липаев В.В. Проектирование программных систем. М.: Высш. шк, 1990.

2. Буч Г. Объектно-ориентированное проектирование / Пер. с анг. Конкорд, 1996.

3. Майерс Г. Надежность программного обеспечения. М.: Мир, 1980

4. <название статьи/книги/материала> [Электронный ресурс]: <автор>. – Режим доступа: <полный URL>

ПРИЛОЖЕНИЕ А Исходный код программной системы

package ru.home.lt.cf;

import lrapi.lr;

import org.apache.commons.configuration.ConfigurationException;

import org.apache.commons.configuration.XMLConfiguration;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class UC01CreateApplication extends UseCase{

final static private Logger logger = LoggerFactory.getLogger(UC01CreateApplication.class);

public static Double createILSFPercentage = 82.82;

public static Double createCWFPPercentage = 10.80;

public static Double createLLQ1Percentage = 5.21;

public static Double createCORNPercentage = 0.78;

public static Double createCCLPPercentage = 0.39;

private UC01CreateILSF ilsf;

private UC01CreateCWFP cwfp;

private UC01CreateLLQ1 llq1;

private UC01CreateCORN corn;

private UC01CreateCCLP cclp;

static {

XMLConfiguration config = null;

try {

config = new XMLConfiguration("../load-test.xml");

} catch (ConfigurationException e) {

logger.error(e.getLocalizedMessage());

}

createILSFPercentage = config.getDouble(LoadTestConfiguration.CREATE\_ILSF\_APPLICATION\_PERCENTAGE, 82.82);

createCWFPPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CWFP\_APPLICATION\_PERCENTAGE, 10.80);

createLLQ1Percentage = config.getDouble(LoadTestConfiguration.CREATE\_LLQ1\_APPLICATION\_PERCENTAGE, 5.21);

createCORNPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CORN\_APPLICATION\_PERCENTAGE, 0.78);

createCCLPPercentage = config.getDouble(LoadTestConfiguration.CREATE\_CCLP\_APPLICATION\_PERCENTAGE, 0.39);

LogProductPercentages();

}

ПРИЛОЖЕНИЕ Б Структура таблиц БД



Рисунок Б.1. Структура таблицы "Пользователи"

**карман с диском на котором записаны материалы курсового проектирования:**

1. пояснительная записка

2. проект системы с исходными кодами

3. дополнительные компоненты системы, необходимые для ее развертывания

4. прочие вспомогательные источники

5. презентация по которой будет проходить защита работы

**Папка(файлик) для графического материала к пояснительной записке и сам графический материал в ней**