Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

(ВолгГТУ)

Кафедра «Программное обеспечение автоматизированных систем»

**«Проектные решения»**

Выполнили: студенты группы ПрИн-466

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО студента) (подпись студента)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО студента) (подпись студента)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО студента) (подпись студента)

Проверил:

к.т.н.,доц. Жукова И.Г.\_\_\_\_\_\_

Научный консультант ст. преподаватель

Ляпина О. Н.\_\_\_\_\_\_\_

Волгоград, 2017

Содержание

[Введение 3](#_Toc497823932)

[1 Диаграмма вариантов использования 4](#_Toc497823933)

[2 Сценарии работы программы 5](#_Toc497823934)

[3 Диаграмма активности 7](#_Toc497823935)

[4 ER-диаграмма 8](#_Toc497823936)

[5 Диаграмма классов 9](#_Toc497823937)

[6 Описание классов 10](#_Toc497823938)

[7 Макеты экранных форм для первого релиза программы 13](#_Toc497823939)

[8 Тестирование программы 20](#_Toc497823940)

[9 График работ по проекту (распределение работ внутри команды по двухнедельному циклу) 21](#_Toc497823941)

[Приложение А. Журнал тестирования 25](#_Toc497823942)

# Введение

Была поставлена задача разработать программу, которая бы являлась оптимизацией бизнес-процесса в сказке «Теремок». Для достижения данной цели были выбраны наиболее подходящие архитектура программы и технологии разработки (см. документ «Анализ требований»). Таким образом, были разработан первый и второй релизы программы, обладающие указанным в документе «Анализ требований» функционалом.

# 1 Диаграмма вариантов использования

Диаграмма вариантов использования представлена на рисунке 1.

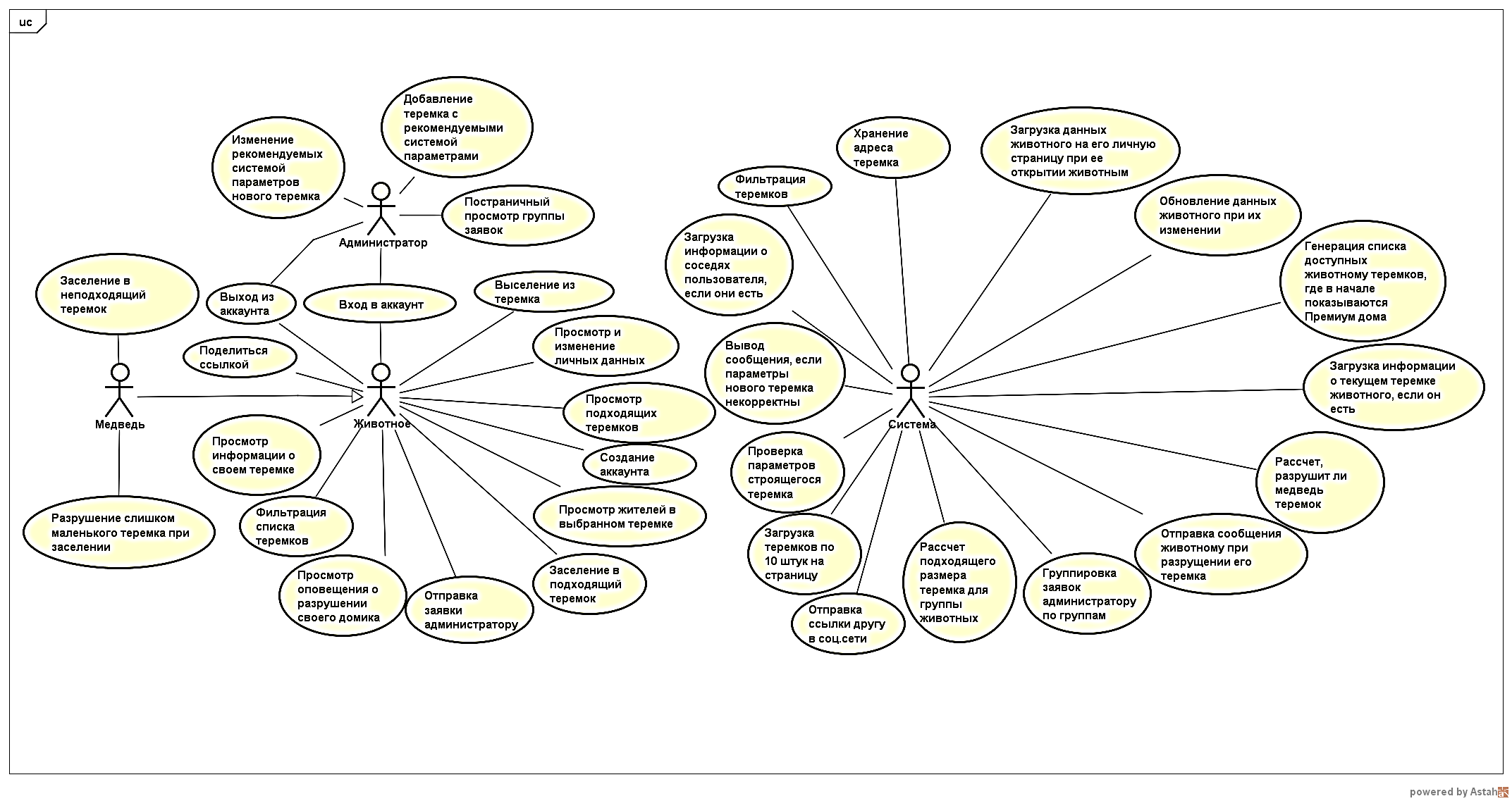


Рисунок 1. Диаграмма вариантов использования

# 2 Сценарии работы программы

Успешные сценарии работы программы представлены ниже.

**I. Регистрация**

Пользователь находится на странице входа на сайт (см. рис. 5).

1. Пользователь нажимает кнопку «Регистрация»
2. Система перенаправляет пользователя на страницу регистрации
3. Пользователь вводит тип в поле «Тип животного»
4. Пользователь вводит логин в поле «Логин»
5. Пользователь вводит пароль в поле «Пароль»
6. Пользователь вводит рост в поле «Рост»
7. Пользователь вводит длину в поле «Длина»
8. Пользователь жмет на кнопку «Зарегистрироваться»
9. Система проверяет логин на уникальность

5.1) Если введенный логин уже существует

5.1.2) Система выдает сообщение о том, что логин уже занят

5.2) Иначе

5.2.1) Система заносит пользователя в базу данных

1. Система перенаправляет пользователя в личный кабинет

**II. Вход**

Пользователь находится на странице входа на сайт (см. рис. 5).

1. Пользователь вводит логин
2. Пользователь вводит пароль
3. Пользователь жмет кнопку «Войти»
4. Система проверяет есть ли в базе данных пользователь с введенными логином и паролем

4.1) Если такого пользователя нет

4.1.2) Система выдает сообщение о том, что логин или пароль введен неправильно

4.2) Иначе

4.2.1) Система перенаправляет пользователя в личный кабинет

**III. Поиск жилья**

Пользователь находится в личном кабинете (см. рис. 8).

1. Пользователь жмет на кнопку «Найти теремок»
2. Система перенаправляет на страницу заселения
3. Система генерирует и отображает постраничный список доступных теремков
4. Пользователь устанавливает фильтры поиска
5. Пользователь применяет фильтры нажатием кнопки «Применить»
6. Система генерирует и отображает постраничный список теремков, проходящих по заданным фильтрам
7. Если в списке есть подходящий теремок

3.1) то пользователь нажимает кнопку «Заселиться»

3.2) Система вносит пользователя в список жителей данного теремка

1. Иначе пользователь завершает поиск, покидая страницу

# 3 Диаграмма активности

На рисунке ниже представлена диаграмма активности.

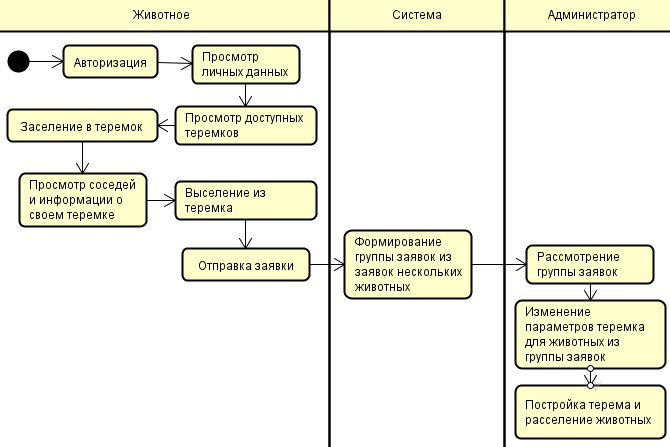


Рис. 2 Диаграмма активности

# 4 ER-диаграмма

Ниже на рисунке представлена ER-диаграмма.

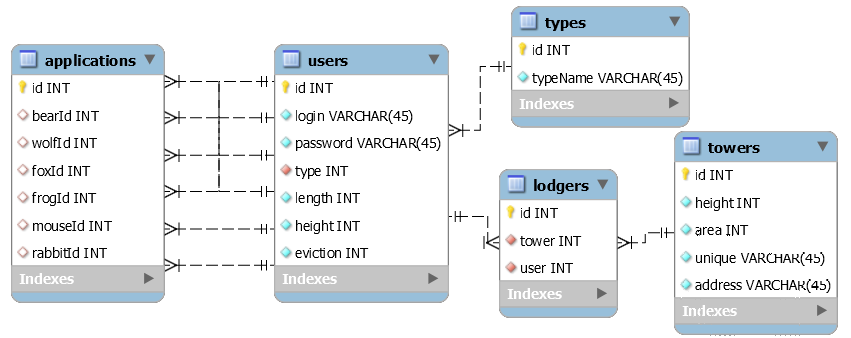


Рис. 3 ER-диаграмма

# 5 Диаграмма классов

На рисунке ниже представлена диаграмма классов.

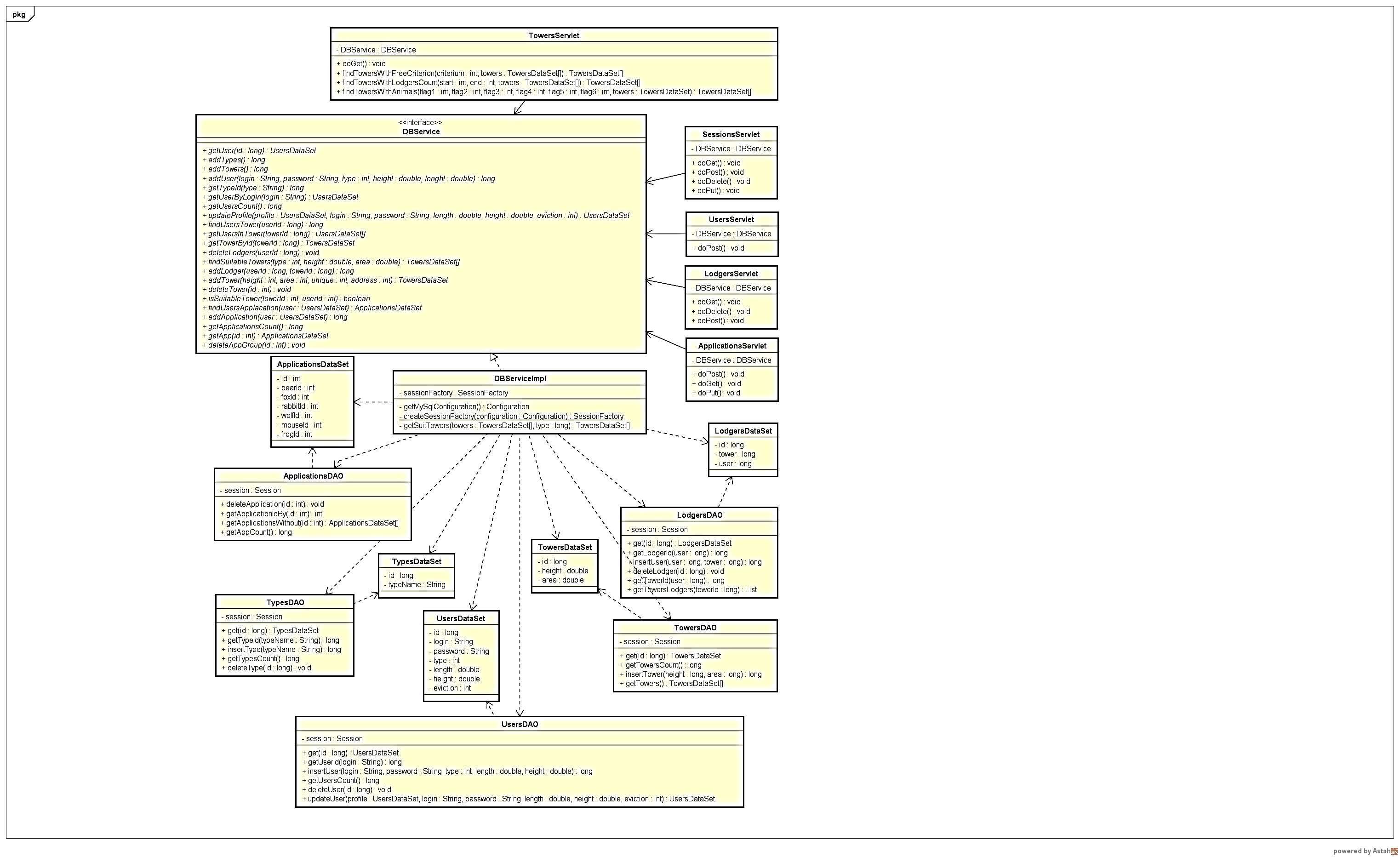


Рис. 4 Диаграмма классов

# 6 Описание классов

Таблица 1. Классы и интерфейсы пакета dbService

|  |  |
| --- | --- |
| Класс/интерфейс | Описание |
| public class DBException  extends [Exception](http://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html?is-external=true) | Класс возвращающий исключение при некорректной работе с БД |
| public interface DBService | Интерфейс работы с базой данных |
| public class DBServiceImpl  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true)  implements [DBService](file:///C:\Users\Anastasia\Desktop\businessprocess\server\target\site\apidocs\dbService\DBService.html) | Класс – имплементация DBServic'а, реализует методы работы с базой данных и соединение с ней |

Таблица 2. Классы пакета DAO

|  |  |
| --- | --- |
| Класс | Описание |
| public class LodgersDAO  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true) | Класс, который предоставляет доступ к таблице lodgers в БД  Поля:  Session [session](file:///C:\Users\Anastasia\Desktop\businessprocess\server\target\site\apidocs\dbService\dao\LodgersDAO.html#session) - сессия для которой совершаем обращение к базе данных |
| public class TowersDAO  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true) | Класс, который предоставляет доступ к таблице towers в БД  Поля:  Session [session](file:///C:\Users\Anastasia\Desktop\businessprocess\server\target\site\apidocs\dbService\dao\LodgersDAO.html#session) - сессия для которой совершаем обращение к базе данных |
| public class TypesDAO  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true) | Класс, который предоставляет доступ к таблице types в БД  Поля:  Session [session](file:///C:\Users\Anastasia\Desktop\businessprocess\server\target\site\apidocs\dbService\dao\LodgersDAO.html#session) - сессия для которой совершаем обращение к базе данных |
| public class UsersDAO  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true) | Класс, который предоставляет доступ к таблице users в БД  Поля:  Session [session](file:///C:\Users\Anastasia\Desktop\businessprocess\server\target\site\apidocs\dbService\dao\LodgersDAO.html#session) - сессия для которой совершаем обращение к базе данных |
| public class ApplicationsDAO  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true) | Класс, который предоставляет доступ к таблице applications в БД  Поля:  Session session - сессия для которой совершаем обращение к базе данных |

Таблица 3. Классы пакета DataSets

|  |  |
| --- | --- |
| Класс | Описание |
| @Entity  public class LodgersDataSet  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true)  implements [Serializable](http://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html?is-external=true) | Класс для хранения данных таблицы lodgers, полученных из базы данных  Поля:  long id – id жильца  long tower – id терема из таблицы towers  long user – id животного из таблицы users |
| @Entity  public class TowersDataSet  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true)  implements [Serializable](http://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html?is-external=true) | Класс для хранения данных таблицы towers, полученных из базы данных  Поля:  long id – id терема  double height – высота терема  double area – общая площадь терема  unique – флаг того, что теремок типа премиум  address – адрес терема |
| @Entity  public class UsersDataSet  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true)  implements [Serializable](http://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html?is-external=true) | Класс для хранения данных таблицы users, полученных из базы данных  Поля:  long id – id животного  String login – логин животного  String password – пароль животного  long type – тип животного  double length – длина животного  double height – рост животного  int eviction – флаг, сломал ли кто-то терем животного |
| @Entity  public class TypesDataSet  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true)  implements [Serializable](http://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html?is-external=true) | Класс для хранения данных таблицы types, полученных из базы данных  Поля:  long id – id типа  String typeName – тип животного |
| @Entity  public class ApplicationsDataSet  extends Object  implements Serializable | Класс для хранения данных таблицы applications, полученных из базы данных  Поля:  Id - Id группы заявок  bearId- Id медведя в группе заявок  wolfId- Id волкав группе заявок  foxId- Id лисы в группе заявок  rabbitId- Id зайца в группе заявок  frogId- Id лягушки в группе заявок  mouseId- Id мышки в группе заявок |

Таблица 4. Классы пакета Servlets

|  |  |
| --- | --- |
| Класс | Описание |
| public class LodgersServlet  extends javax.servlet.http.HttpServlet | Сервлет для обработки запросов с клиента на работу с таблицей жильцов  Поля:  DBService dbService – класс для работы с базой данных |
| public class SessionsServlet  extends javax.servlet.http.HttpServlet | Сервлет для обработки запросов с клиента на работу с сессией и данными из сессии  Поля:  DBService dbService – класс для работы с базой данных |
| public class TowersServlet  extends javax.servlet.http.HttpServlet | Сервлет для обработки запросов с клиента на работу с таблицей теремков  Поля:  DBService dbService – класс для работы с базой данных |
| public class UsersServlet  extends javax.servlet.http.HttpServlet | Сервлет для обработки запросов с клиента на работу с таблицей животных  Поля:  DBService dbService – класс для работы с базой данных |
| public class ApplicationsServlet  extends javax.servlet.http.HttpServlet | Cервлет для обработки запросов связанных с заявками пользователей  Поля:  DBService dbService – класс для работы с базой данных |

Таблица 5. Классы пакета main

|  |  |
| --- | --- |
| Класс | Описание |
| public class Main  extends [Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html?is-external=true) | Главный класс проекта, который запускает сервер |

# 7 Макеты экранных форм для первого релиза программы

Макеты экранных форм представлены на рисунках ниже.

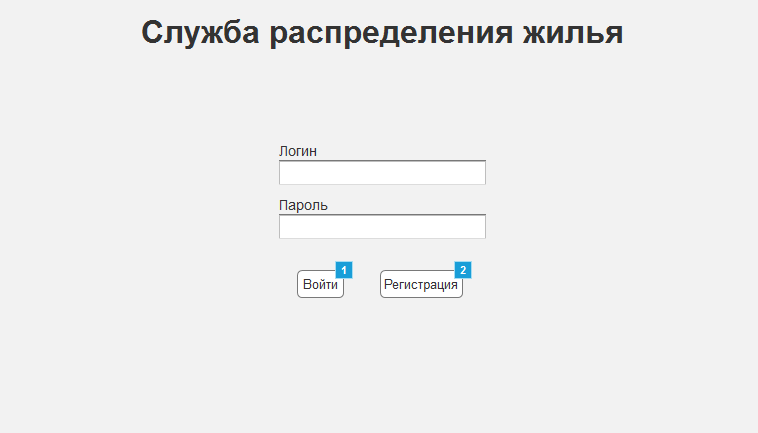


Рис. 5 Вход на сайт

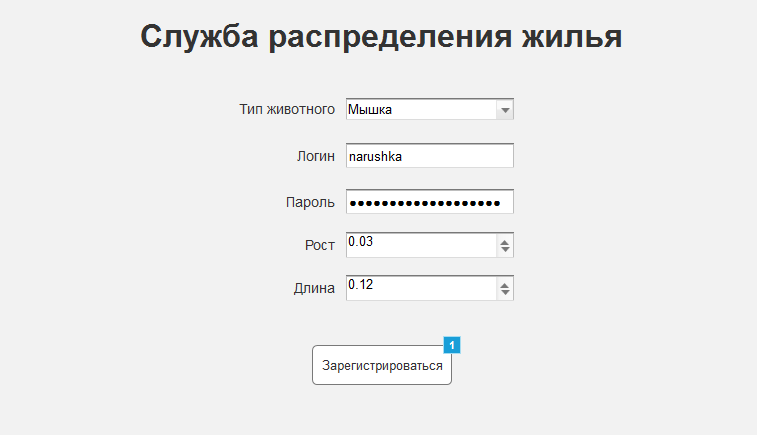


Рис. 6 Регистрация

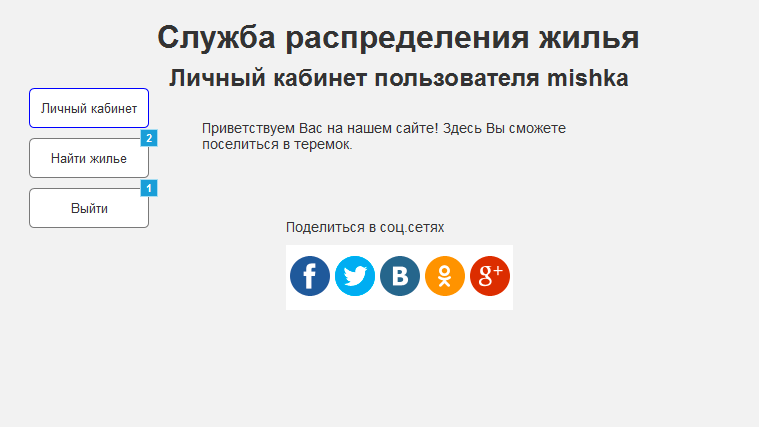


Рис. 7 Главная страница

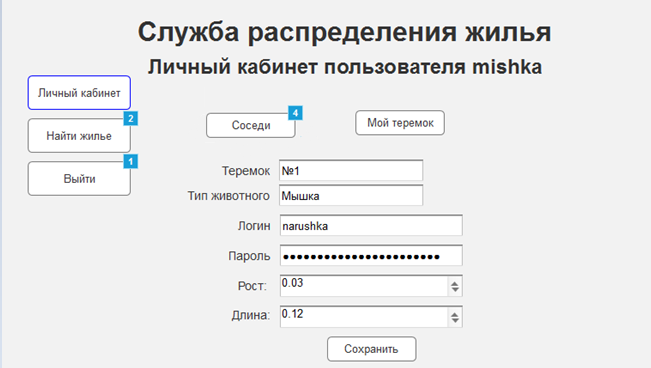


Рис. 8 Личный кабинет

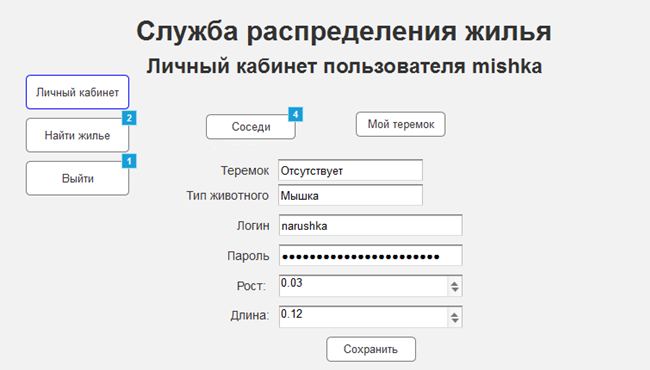


Рис. 9 Личный кабинет (у животного нет теремка)

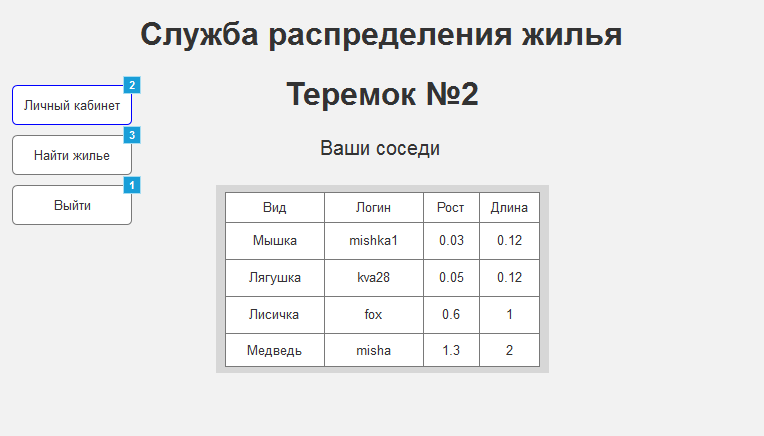


Рис. 10 Мои соседи

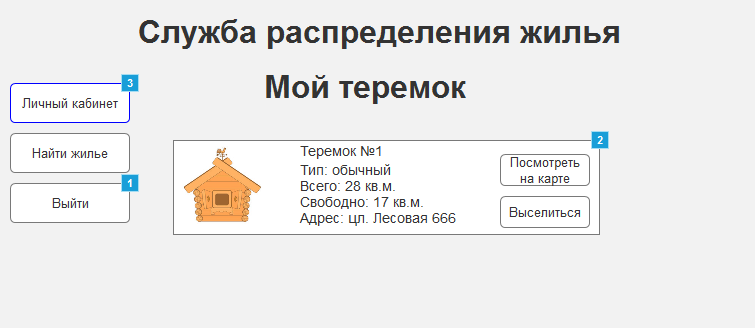


Рис. 11 Мой теремок

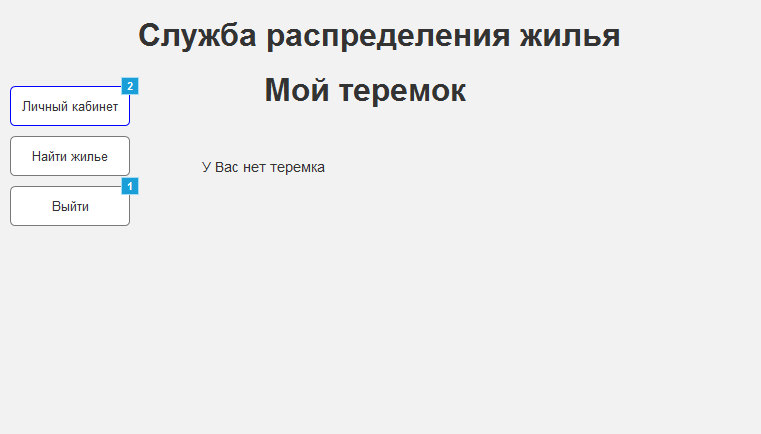


Рис. 12 Мой теремок (в случае, если у животного нет теремка)

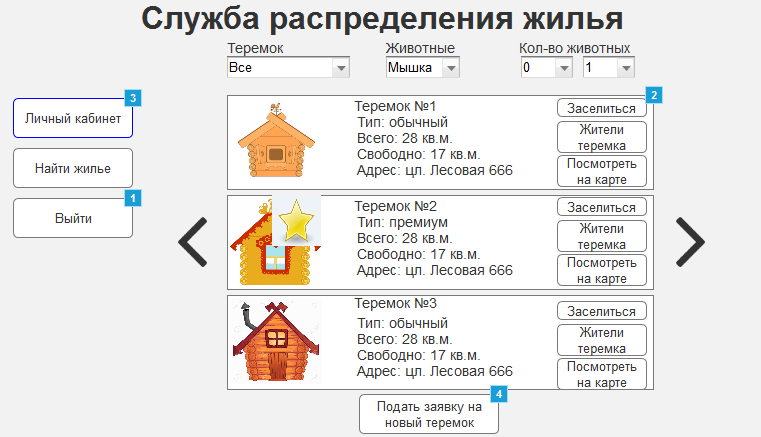


Рис. 13 Страница заселения

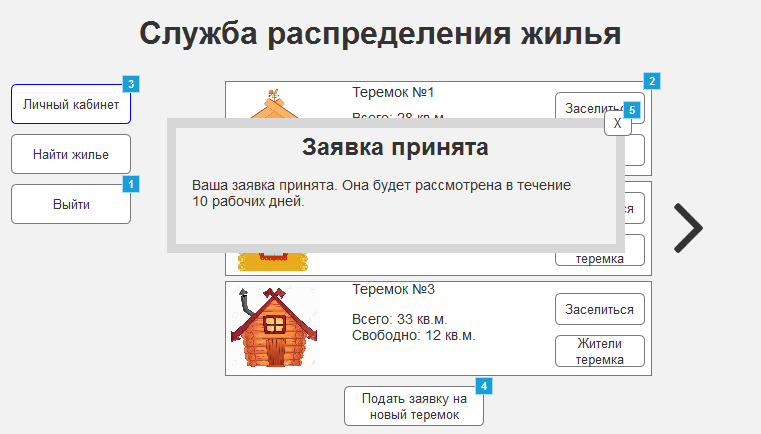


Рис. 14 Нажатие кнопки «Подать заявку на новый теремок»

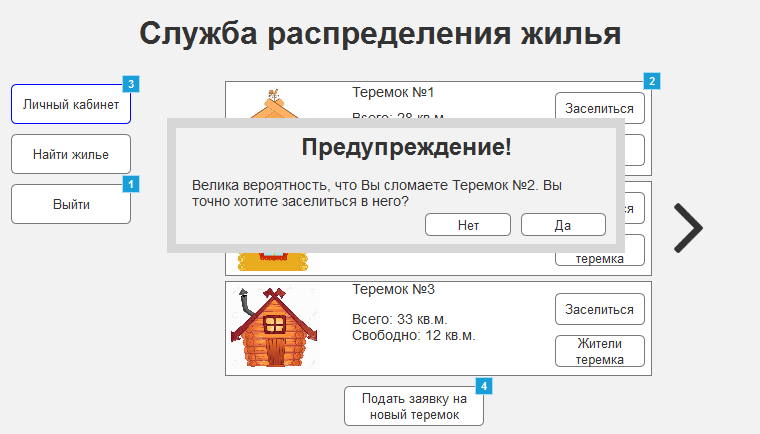


Рис. 15 Нажатие кнопки «Заселиться» медведем

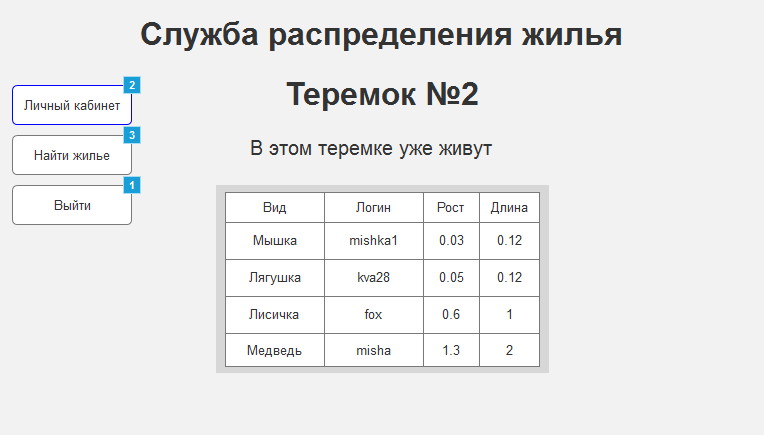


Рис. 16 Жители теремка

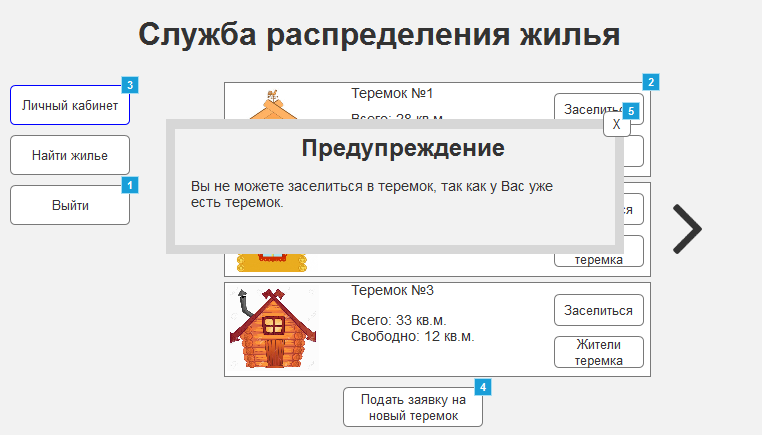


Рис. 17 Нажатие на кнопку «Заселиться» в случае, когда у животного уже есть теремок

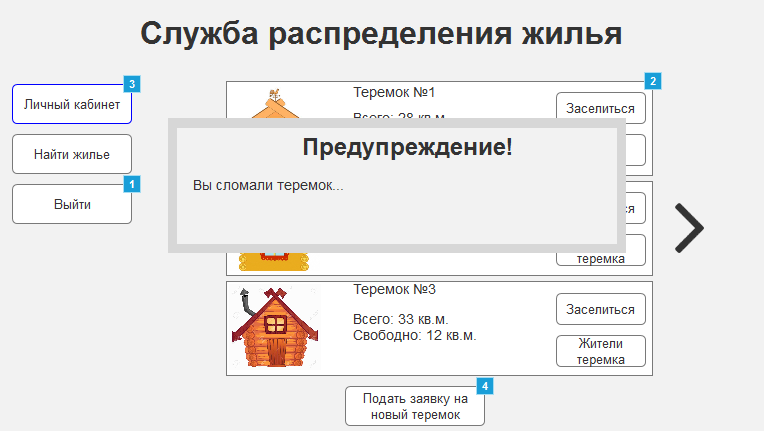


Рис. 18 Медведь сломал теремок

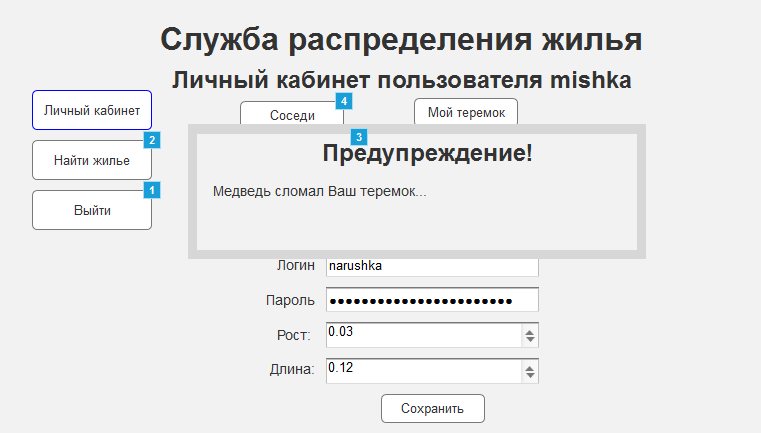


Рис. 19 Сообщение животному о том, что медведь сломал теремок



Рис. 20 Личная страница администратора



Рис. 21 Администратор ввел некорректные параметры теремка при его постройке

# 8 Тестирование программы

Тестирование программы представлено в приложении А данного документа.

# 9 График работ по проекту (распределение работ внутри команды по двухнедельному циклу)

Таблица 6. График работ по проекту 12.09.17 – 26.09.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Работа с заказчиком. Постановка задачи. | Шалиевская А.А.  Островский Д.С.  Тажибова А.В. | 3 ч. | 3 ч. | 100% |
| Распределение задач на 1 период работы | Шалиевская А.А. | 30 мин. | 30 мин. | 100% |
| Предварительное распределение обязанностей между членами команды. Обсуждение. | Шалиевская А.А.  Островский Д.С.  Тажибова А.В. | 1 ч. | 1 ч. | 70% |
| Выбор сказки, по которой будет реализован бизнес-процесс | Шалиевская А.А.  Островский Д.С.  Тажибова А.В. | 6 ч. | 6 ч. | 100% |
| Обсуждение подробностей программы | Шалиевская А.А.  Островский Д.С.  Тажибова А.В. | 4 ч. | 4 ч. | 100% |
| Составление документа с предложениями команды по автоматизации БП | Тажибова А.В. | 1 ч. | 1 ч. | 100% |

Таблица 7. График работ по проекту 26.09.17 – 10.10.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Работа с заказчиком. Уточнение требований.  Постановка задачи (п.1\*). | Шалиевская А.А.  Островский Д.С. | 3 ч. | 3 ч. | 100% |
| Составление графика на 2 цикл работы (п.21).  Распределение ролей между членами команды (п.2). | Шалиевская А.А. | 1.5 ч. | 1.1 ч. | 100% |
| Обсуждение требований заказчика в команде | Шалиевская А.А.  Тажибова А.В.  Островский Д.С. | 2 ч. | 1.2 ч. | 100% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Редактирование п. 3,4,5,6,7,8,11  +  Обсуждение их с командой | Тажибова А.В. | 3 ч. | 3 ч. | 100% |
| Заполнение глоссария (п.22), формулировка нефункциональных требований (п.10) + обсуждение с командой | Островский Д.С. | 1 ч. + 1ч. | 1.5 ч. | 80% |
| Разработка диаграммы вариантов использования (п.13), планирование разработки (п.20) | Шалиевская А.А. | 2 ч. | 2 ч. | 100% |
| Написание сценариев использования программы (п.14) + обсуждение с командой | Тажибова А.В. | 3 ч.+1ч. | 30 мин. | 100% |
| Формулировка функциональных требований (п.9), создание треугольника компромиссов (п.12) + обсуждение с командой | Шалиевская А.А. | 2 ч. | 1 ч. | 90% |
| Создание макетов экранных форм (п. 15), п.16,17,18,19 | Островский Д.С. | 2 ч. | 2 ч. | 90% |
| ER – диаграмма + обсуждение с командой | Островский Д.С. | 2 ч. +1 ч. | - | 70% |
| Изучение теоретических материалов для создания сервера | Шалиевская А.А. | 6 ч. | 4 ч. | 50% |
| Подготовка документа «Описание концепции проекта» к проверке | Тажибова А.В. | 2 ч. | 3 ч. | 90% |

Таблица 8. График работ по проекту 10.10.17 – 24.10.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Работа с заказчиком. Уточнение требований. | Шалиевская А.А.  Островский Д.С.  Тажибова А.В. | 3 ч. | 3 ч. | 100% |
| Составление графика на 3 цикл работы (п.21) | Шалиевская А.А. | 30 мин. | 30 мин. | 100% |
| Исправление замечаний по назначенным пунктам документа | Шалиевская А.А. | 2 ч. | 1.5 ч. | 100% |
| Исправление замечаний по назначенным пунктам документа | Тажибова А.В. | 2 ч. | 3 ч. | 100% |
| Исправление замечаний по назначенным пунктам документа | Островский Д.С. | 2 ч. | 1 ч. | 100% |
| ER – диаграмма + обсуждение с командой | Островский Д.С. | 2 ч. | 1 ч. | 100% |
| Окончательное оформление документа «Описание концепции проекта» | Тажибова А.В. | 2 ч. | 3 ч. | 100% |
| Верстка утвержденных макетов | Тажибова А.В. | 10 ч. | 4 ч. | 20% |
| Изучение теоретических материалов для создания сервера | Шалиевская А.А. | 4 ч. | 4 ч. | 100% |
| Создание сервера | Шалиевская А.А. | 2 ч. | 2 ч. | 100% |

Таблица 9. График работ по проекту 24.10.17 – 7.11.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Работа с заказчиком. Проверка документов | Тажибова А.В. | 3 ч. | 0.5 ч. | 100% |
| Составление графика на 4 цикл работы | Шалиевская А.А. | 30 мин. | 30 мин. | 100% |
| Исправление замечаний по назначенным пунктам документа | Шалиевская А.А. | 2 ч. | 1.5 ч. | 100% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Исправление замечаний по назначенным пунктам документа | Тажибова А.В. | 2 ч. | 2 ч. | 100% |
| Исправление замечаний по назначенным пунктам документа | Островский Д.С. | 1 ч. | 1 ч. | 100% |
| Написание тестов | Островский Д.С. | 2 ч. | 2 ч. | 100% |
| Тестирование программы | Островский Д.С. | 2 ч. | 2 ч. | 100% |
| Связка клиента и сервера | Островский Д.С. | 10 ч. | 18 ч. | 100% |
| Верстка утвержденных макетов | Тажибова А.В. | 30 ч. | 49 ч. | 100% |
| Написание сервера | Шалиевская А.А. | 40 ч. | 35 ч. | 100% |
| Создание ER-диаграммы | Шалиевская А.А. | 1 ч. | 1 ч. | 100% |
| Создание диаграммы классов | Шалиевская А.А. | 2 ч. | 1.5 ч. | 100% |
| Создание презентации | Шалиевская А.А. | 3 ч. | 4 ч. | 100% |
| Диаграмма активности | Шалиевская А.А. | 1 ч. | 1 ч. | 100% |
| Подготовка раздела «Тестирование» в документе проектные решения | Островский Д.С. | 2 ч. | 1 ч. | 100% |
| Исправление ошибок, найденных при тестировании | Шалиевская А.А. | 2 ч. | 3 ч. | 100% |
| Подготовка итогового варианта документа «Анализ требований» | Тажибова А.В. | 4 ч. | 5 ч. | 100% |
| Подготовка итогового варианта документа «Проектные решения» | Тажибова А.В. | 3 ч. | 3 ч. | 100% |

\*п.1,2…- пункты в документе «Анализ требований».

Таблица 10. График работ по проекту 7.11.17 – 21.11.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Исправить ошибки в 1 релизе | Шалиевская | 10ч. | 10 ч. | 100% |
| Исправить ошибки в 1 релизе | Тажибова | 4ч | 3ч | 100% |
| Исправить ошибки в 1 релизе | Островский | 6ч | 4ч | 100% |
| Реализовать возможность поделиться с друзьями | Шалиевская | 2ч | 1ч | 100% |
| Реализация режима админа | Шалиевская | 5ч | 4ч | 100% |
| Реализация режима админа | Островский | 4ч | 3ч | 100% |
| Верстка страницы админа | Тажибова | 3ч | 4ч | 100% |
| Реализация на сервере возможности подавать заявки | Шалиевская | 4ч | 4ч | 100% |

Таблица 11. График работ по проекту 22.11.17 – 15.12.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Реализация возможности подавать заявки | Островский | 4ч. | 5 ч. | 100% |
| Отрисовка информационных окон | Тажибова | 2ч | 2ч | 100% |
| Реализация постраничного просмотра | Островский | 4ч | 4ч | 100% |
| Реализация адреса | Островский | 3ч | 3ч | 100% |
| Реализация премиум домов | Шалиевская | 3ч | 3ч | 100% |
| Подключение к клиенту премиум домов | Островский | 1ч | 1ч | 100% |
| Изменение внешнего вида премиум домов | Тажибова | 0.5ч | 0.5ч | 100% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Исполнитель | Планируемые часы работы | Реально затраченные часы | Процент выполнения |
| Реализация системы фильтров | Шалиевская | 6ч | 8ч | 100% |
| Подключение системы фильтров | Островский | 2ч. | 3ч | 100% |
| Отрисовка фильтров | Тажибова | 3ч | 5ч | 100% |
| Исправление документации | Тажибова | 3ч | 3ч | 100% |
| Редактирование диаграмм, заполнение графика разработки, написание паспортов классов | Шалиевская | 3ч | 3ч | 100% |
| Тестирование программы | Островский | 4ч | 4ч | 100% |
| Исправление ошибок | Шалиевская  Островский  Тажибова | 5ч | 4ч | 100% |
| Создание презентации | Шалиевская | 5ч | 5ч | 100% |

# Приложение А. Журнал тестирования

Тестирование запросов проводилось в программе для автоматического тестирования API – Postman. Тестирование методов сервера проводилось с помощью JUnit.

1. Тестирование запросов

Для всех тестов, начальными предусловиями являются следующие шаги:

– запуск разрабатываемого сервера;

– запуск приложения Postman;

Таблица А.1.1 Тест регистрации

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тест регистрации нового пользователя | | |
| **Функция:** | Регистрация | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку “Регистрация”, выбрать POST запрос | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Введите следующие строчки в поля для входных данных:  key: “login” value: “testlogin”  password: “password” value: “testpass”  key: “type” value: “1”  key: “length” value: “5”  key: “height” value: “3” | | Данные успешно введены |  |
| Нажмите на кнопку Send | | Открылось новое поле с текстом:  {"id":8,  "login":"testlogin",  "password":"testpass",  "type":1,  "length":4.0,  "height":3.0,  "eviction":0,  "terem":-1,  "evictionReal":0}  Статус: 200 OK |  |
| **Постусловие:** | |  | |

Таблица А.1.2 Авторизации

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование авторизации пользователя | | |
| **Функция:** | Вход в систему | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключиться на вкладку “Действия над пользователем”, выбрать POST запрос. | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Введите следующие строчки в поля для входных данных:  key: “login” value: “testlogin”  password: “password” value: “testpass” | | Данные успешно введены |  |
| Нажмите на кнопку Send | | Открылось новое поле с текстом:  {"id":8,  "login":"testlogin",  "password":"testpass",  "type":1,  "length":4.0,  "height":3.0,  "eviction":0,  "terem":-1,  "evictionReal":0}  Статус: 200 OK |  |
| **Постусловие:** | |  | |

Таблица А.1.3 Тест получения данных пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование получения информации о пользователе | | |
| **Функция:** | Получения данных пользователя | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку “Действия над пользователем”, выбрать GET запрос. | | Открылась вкладка для теста |  |
| **Шаги теста:** | |  | |
| Нажмите на кнопку Send | | Полученный ответ:  {"id":8,  "login":"testlogin",  "password":"testpass",  "type":1,  "length":4.0,  "height":3.0,  "eviction":0,  "terem":-1,  "evictionReal":0}  Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.4 Тест выхода из аккаунта

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование выхода из аккаунта | | |
| **Функция:** | Выход из аккаунта | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку  “Действия над пользователем”, выбрать DELETE запрос | | Открылась вкладка для теста |  |
| **Шаги теста:** | |  | |
| Нажмите на кнопку Send | | Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.5 Тест изменения информации о себе

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование изменения информации о пользователе | | |
| **Функция:** | Изменение данных пользователя | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку “Действия над пользователем”, выбрать PUT запрос. | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Введите следующие строчки в поля для входных данных:  key: “login” value: “testlogin”  password: “password” value: “testpass”  key: “length” value: “6”  key: “height” value: “4” | | Данные успешно введены |  |
| Нажмите на кнопку Send | | Полученный ответ:  {"id":8,  "login":"testlogin",  "password":"testpass",  "type":1,  "length":6.0,  "height":4.0,  "eviction":0,  "terem":-1,  "evictionReal":0}  Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.6 Тест заселения пользователя в теремок

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование заселения пользователя | | |
| **Функция:** | Заселение пользователя в теремок | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку  “Действия над теремком”, выбрать POST запрос. | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Введите следующие строчки в поля для входных данных:  key: “id” value: “1”  key: “first” value: “1” | | Данные успешно введены |  |
| Нажмите на кнопку Send | | Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.7 Тест получения информации о теремке

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование ввода товара в систему | | |
| **Функция:** | Ввод товара в систему | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку  “Действия над теремком”, выбрать GET запрос. | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Введите следующие строчки в поля для входных данных:  key: “mine” value: “1” | | Данные успешно введены |  |
| Нажмите на кнопку Send | | Полученный ответ:  {"terem":1,  "area":20000,  "height":200,  "count":0,  "users":[],  "freeArea":19976.0}  Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.8 Тест выселения пользователя из теремка

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование выселения пользователя | | |
| **Функция:** | Выселение пользователя | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку  “Действия над теремком”, выбрать DELETE запрос | | Открылась вкладка для теста |  |
| **Шаги теста:** | |  | |
| Нажмите на кнопку Send | | Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.9 Тест получения подходящих теремков

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование вывода подходящих теремков | | |
| **Функция:** | Вывод подходящих теремков | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку  “Действия над теремками”, выбрать GET запрос. | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Нажмите на кнопку Send | | Полученный ответ:  {"towers":[  {"id":1,"height":200,"area":20000,"freeArea":20000.0},  {"id":3,"height":60,"area":20000,"freeArea":20000.0},  {"id":4,"height":90,"area":7000,"freeArea":7000.0},  {"id":5,"height":80,"area":10000,"freeArea":10000.0},  {"id":7,"height":80,"area":500,"freeArea":500.0},  {"id":8,"height":100,"area":100,"freeArea":100.0},  {"id":9,"height":10,"area":60,"freeArea":60.0},  {"id":10,"height":5,"area":300,"freeArea":300.0}]}  Статус: 200 ОК |  |
| **Постусловие:** | |  | |

Таблица А.1.10 Тест заселения пользователя Медведь в неподходящий теремок

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование заселения пользователя Медведь в неподходящий теремок | | |
| **Функция:** | Заселение пользователя в теремок | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**  Пройден |
| **Предусловие:** | |  | |
| В Postman переключится на вкладку  “Действия над теремком”, выбрать POST запрос. | | Открылась вкладка для теста |  |
| Открыть окно Params | | Открылось поле для ввода параметров |  |
| **Шаги теста:** | |  | |
| Введите следующие строчки в поля для входных данных:  key: “id” value: “1”  key: “first” value: “0” | | Данные успешно введены |  |
| Нажмите на кнопку Send | | Статус: 400 bad request.  Теремок пропал из базы данных. |  |
| **Постусловие:** | |  | |

1. Тестирование JUnit

Для тестирования методов сервера необходимо включить тестирование в проекте и запустить его.

А.2.1 Тест правильности получения пользователя при корректных данных

|  |  |
| --- | --- |
| **Название:** | Тестирование получения пользователя по id при корректных данных |
| **Функция:** | testGetUserTrue() |
| **Ожидается:** | Получен запрашиваемый пользователь |
| **Получено:** | Получен запрашиваемый пользователь |
| **Результат:** | Тест пройден |

А.2.2 Тест корректности вывода ошибки при получении пользователя по некорректному id

|  |  |
| --- | --- |
| **Название:** | Тестирование получения пользователя по id при корректных данных |
| **Функция:** | testGetUserFalse() |
| **Ожидается:** | Возврат null |
| **Получено:** | Возврат null |
| **Результат:** | Тест пройден |

А.2.3 Тест правильности id пользователя при добавлении в базу данных

|  |  |
| --- | --- |
| **Название:** | Тестирование правильности id пользователя при добавлении в базу данных |
| **Функция:** | testAddUser() |
| **Ожидается:** | Добавленный пользователь имеет наибольший по значению id |
| **Получено:** | Добавленный пользователь имеет наибольший по значению id |
| **Результат:** | Тест пройден |

А.2.4 Тест вывода типа животного по его id

|  |  |
| --- | --- |
| **Название:** | Тестирование вывода типа животного по его id |
| **Функция:** | testGetTypeIdTrue() |
| **Ожидается:** | Все введенные id вывели правильный тип животного |
| **Получено:** | Все введенные id вывели правильный тип животного |
| **Результат:** | Тест пройден |

А.2.5 Тест правильности получения пользователя при корректных данных

|  |  |
| --- | --- |
| **Название:** | Тестирование получения пользователя по логину при корректных данных |
| **Функция:** | testGetUserByLoginTrue() |
| **Ожидается:** | Получен запрашиваемый пользователь |
| **Получено:** | Получен запрашиваемый пользователь |
| **Результат:** | Тест пройден |

А.2.6 Тест корректности вывода ошибки при получении пользователя по некорректному логину

|  |  |
| --- | --- |
| **Название:** | Тестирование получения пользователя по id при корректных данных |
| **Функция:** | testGetUserByLoginFalse() |
| **Ожидается:** | Возврат null |
| **Получено:** | Возврат null |
| **Результат:** | Тест пройден |