



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Шаблоны программных платформ языка Джава
по профилю: Разработка программных продуктов и проектирование информационных систем
направления профессиональной подготовки: 09.03.04 Программная инженерия

Тема: «Приложение «Магазин компьютерных комплектующих»»»

Студент: Мурадов Натик Намигович

Группа: ИКБО-16-19

Работа представлена к защите 29.11.2021 (дата) _____ /Мурадов Н.Н. /
(подпись и ф.и.о. студента)

Руководитель: Матчин Василий Тимофеевич, ст. преподаватель

Работа допущена к защите _____ (дата) _____ / Матчин В.Т. /
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____/_____/_____
_____/_____/_____

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших
защиту)

М. РТУ МИРЭА. 2021 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ЗАДАНИЕ на выполнение курсовой работы

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Программная инженерия

Студент: Мурадов Натик Намигович

Группа: ИКБО-16-19

Срок представления к защите:

Руководитель: Старший преподаватель ИиППО Матчин Василий Тимофеевич

Тема: «Разработка приложения «Магазин компьютерных комплектующих»

Исходные данные: Документация и средства языка Java.

Перечень вопросов, подлежащих разработке, и обязательного графического материала:

1. Провести анализ предметной области; 2. Провести анализ сравнения с существующими аналогами; 3. Разработать клиентскую часть приложения (принятие и обработка запросов);

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: _____ / Болбаков Р. Г. /, « _____ » _____ 2021 г.

Задание на КР выдал: _____ / Матчин В. Т. /, « _____ » _____ 2021 г.

Задание на КР получил: Мурадов / Мурадов Н. Н. /, « 17 » 02 2021 г.

Мурадов Н.Н. Приложение «Магазин компьютерных комплектующих» / Курсовая работа по дисциплине «Шаблоны программных платформ языка Джава» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04. «Программная инженерия» (4ый семестр) / руководитель В.Т. Матчин, ст. преподаватель / кафедры ИиППО Института ИТ РТУ МИРЭА.

Целью работы является проведение анализа предметной области разрабатываемого ресурса, сравнения его с аналогами, разработать клиентскую часть приложения с применением JavaFX, также в качестве дополнительных целей была выбрана реализовать на языке Java при помощи различных технологий улучшенную версию своего сайта(<https://cursach228.000webhostapp.com/>), то есть практика с 3D технологиями на языке Java, воссоздать или создать новые анимации при помощи JavaFX, в ходе создания проекта были определены ещё две цели: создание базы данных на PostgreSQL для дальнейшего его использования в разработанном приложении и сделать проект по возможности максимально автономным, чтоб приложение можно было использовать в качестве портативной версии сайта.

Muradov N.N. Application "Shop of computer components" / Course work on the discipline "Patterns of software platforms of the Java language" profile "Development of software products and design of information systems" of the direction of professional training of bachelor's degree 09.03.04. "Software Engineering" (4th semester) / supervisor V.T. Matchin, Art. Lecturer / Department of IiPPO, Institute of IT RTU MIREA.

The purpose of the work is to analyze the subject area of the resource being developed, compare it with analogues, develop the client part of the application using JavaFX, and as additional goals, it was chosen to implement an improved version of its website in Java using various technologies ([https://cursach228.000webhostapp.com /](https://cursach228.000webhostapp.com/)), that is, practice with 3D technologies in Java, recreate or create new animations using JavaFX, two more goals were defined during the creation of the project: creating a database on PostgreSQL for its further use in the developed application and making the project as autonomous as possible so that the application can be used as a portable version of the site.

Аннотация

Целью данной курсовой работы является воссоздание сайта при помощи языка Java и основанных на нём технологиях, а также реализация полезных дополнений, которые придают особый смысл этому проекту. Такими дополнениями можно назвать портативностью, автономностью и частичной независимостью от интернета.

Раздел «Содержание» содержит названия основных разделов и подразделов с указанием номера страницы.

Раздел «Введение» передает общую характеристику КР, а именно актуальность выбранной темы, указание цели работы и задач для ее выполнения, описание предмета исследования.

Основная часть состоит из трёх разделов: общие сведения, функциональное назначение и описание логической структуры.

В разделе «Общие сведения» излагаются обозначение и наименование, а также программное обеспечение, необходимое для разработки и эксплуатации интернет-ресурса.

В разделе «Функциональное назначение» показан ряд характеристик интернет-ресурса, характеризующихся конкретными характеристиками ряда функций, способных удовлетворять заданные или подразумеваемые потребности.

Раздел «Описание логической структуры» определяет список основных вопросов и подразделов, такие как анализ предметной области, набор выбранных технологий для создания улучшенной копии интернет-ресурса на языке Java.

В разделе «Заключение» излагаются теоретические выводы и предложения, описывающие успех в достижении поставленной ранее цели.

Курсовой проект содержит 46 страницы отчёта, 8 иллюстраций, 6 использованных источников литературы.

Оглавление

Аннотация.....	6
Введение	8
1. Общие сведения	10
1.1. Обозначение и наименование приложения	10
1.2. Прикладное программное обеспечение, необходимое для разработки и функционирования приложения	10
1.3. Языки и технологии, с помощью которых реализовано приложение	12
2. Функциональное назначение	16
3. Описание логической структуры	17
3.1. Анализ предметной области разрабатываемого приложения	17
3.2. Описание структуры данных с использованием UML диаграмм	20
3.3.1 Формирование UML диаграммы	20
3.3. Реализация алгоритма приложения	21
3.3.1 ru.mirea.MireaApplication	21
3.3.2 ru.mirea. SetCtx	22
3.3.3 ru.mirea. StartSpring	22
3.3.4 ru.mirea. StartSpring	22
3.3.5 ru.mirea. ThreeD. Xform	28
3.3.6 ru.mirea. ThreeD. Scene3D	30
3.3.7 ru.mirea. data. User	33
3.3.8 ru.mirea. data. UserRep	34
3.3.9 ru.mirea. data. UsersImpl	34
3.3.10 ru.mirea. data. UsersImpl	35
Заключение.....	43
Список информационных источников	45
Приложения	46
Приложение об исходном коде приложения	46

Введение

Актуальность

В настоящее время в мире быстрыми темпами идет процесс информатизации всех сторон жизни общества, развития и внедрения новых информационных технологий. Компьютерные технологии привели к впечатляющим изменениям в сфере промышленного производства и бизнеса, социальной жизни и образования, науки и культуры. Информация превратилась в глобальный неистощимый ресурс человечества, вступившего в новую эпоху развития цивилизации, которую называют информационной. По оценкам ИТУ (International Telecommunication Union) число пользователей интернета в России на 2018 год 114 920 477 человек. Что составляет 80,9% от общего населения страны. По всему миру число пользователей на 2019 год составляет примерно 4 100 000 000 человек или же 53,6% от общего населения земли. После данных цифр сомнений не остаётся в необходимости использования интернет-ресурсов при создании коммерческих проектов.

Цель работы

Целью данной курсовой работы является, разработка улучшенной версии интернет-ресурса, приложением работающем в системе Windows 10 без браузера, являющемся портативным и частично независимым от сети интернет. Перечень вопросов, подлежащих разработке, и обязательного графического материала:

1. Провести анализ предметной области.
2. Обосновать выбор технологий разработки приложения
3. Провести анализ сравнения с существующими аналогами.
4. Разработать клиентскую часть приложения (принятие и обработка запросов)
5. Создать презентацию по выполненной курсовой работе.

Объект исследования

- Технологии Java

- CSS3
- Scene Builder
- JavaFX
- Spring Framework
- Gradle
- HeidiSQL
- PostgreSQL
- Slf4j
- Launchj4
- Lombok

Предмет анализа

- CSS3
- Современные стандарты разработки веб-ресурсов
- Технологии Java
- Scene Builder
- JavaFX
- Spring Framework
- Gradle
- HeidiSQL
- Launchj4

1. Общие сведения

1.1. Обозначение и наименование приложения

Наименование разрабатываемого приложения – вебсайт «PCBY».

Название было взято от интернет-ресурса, который выступил основой в дизайне данного приложения. Использование английского языка в названии разрабатываемого приложения, обусловлено тем, что он является самым распространённым языком мира. А “PCBY” от PC(Personal Computer) и BY(что созвучно слову Buy(купить)).

1.2. Прикладное программное обеспечение, необходимое для разработки и функционирования приложения

Одним из первых этапов является установка программного обеспечения для проектирования и дальнейшей корректной работы приложения или его разработки.

Для разработки приложения была использована программа IntelliJ IDEA это интегрированная среда разработки программного обеспечения для многих языков программирования в том числе и для языка Java. Удобная программа, имеющая богатое сообщество, большое количество плагинов, которые очень ускоряют разработку приложений, как в команде, так и в одиночку. Также ПО изначально имеет возможность взаимодействия с VCS. Можно работать с репозиториями локальными и удалёнными.

Во время работы с JavaFX было использовано приложение Scene Builder. Ему есть альтернатива, это редактировать в Notepad++ или IntelliJ IDEA, файлы сцен с расширением(форматом) FXML, но это будет менее удобно. Так как Scene Builder даёт возможность полноценно редактировать сцену в визуализированном режиме, при этом не уступая кодовому режиму в возможностях реализации, хоть и иногда её замедляя, но иногда можно выиграть время за счёт не допущенных ошибок и их поисков при кодовой реализации сцены. А можно и комбинировать подходы, взяв плюсы с обеих сторон.

При работе с PostgreSQL, понадобилось ПО для управления БД. Оно было необходимо сразу по нескольким причинам, прежде всего конечно же визуализация текущего состояния БД, а также возможности изменять, удалять и создавать таблицы. Под эту роль отлично подошёл HeidiSQL. Попробовал pgAdmin, но, к сожалению, интерфейс не оказался достаточно удобен, в связи с чем пришлось искать аналоги.

Для компилирования исполнительного файла понадобилось ПО Launch4j. Изначально было достаточно использовать соответствующий плагин в Gradle. Но позднее выросла планка, появились новые цели и функционала, который предоставлял плагин, стало не хватать поэтому пришлось устанавливать одноименную программу.

Notepad++ (Рисунок 1) – это бесплатный редактор программных кодов и обычных текстов. Из названия (notepad по англ. «блокнот») уже понятно, что это за программа и нужна ли она именно вам. Она является незаменимым помощником для программистов, вебмастеров и всех тех, кому необходимо редактировать текстовые данные. Программа идет в комплекте с функцией подсветки синтаксиса для нескольких десятков языков программирования, как многофункциональный блокнот для программиста.

Notepad++ часто сравнивают со стандартным приложением Windows Блокнот. Оценивать плюсы и минусы обеих программ не имеет смысла. Функция Блокнота ограничивается просмотром и незначительным редактированием текстовых файлов. А возможности Нотепада обширны благодаря встраиваемым плагинам. Но он также является равноценной заменой Блокноту.

Все его плюсы делают его простым, лаконичным редактором, без лишних инструментов.

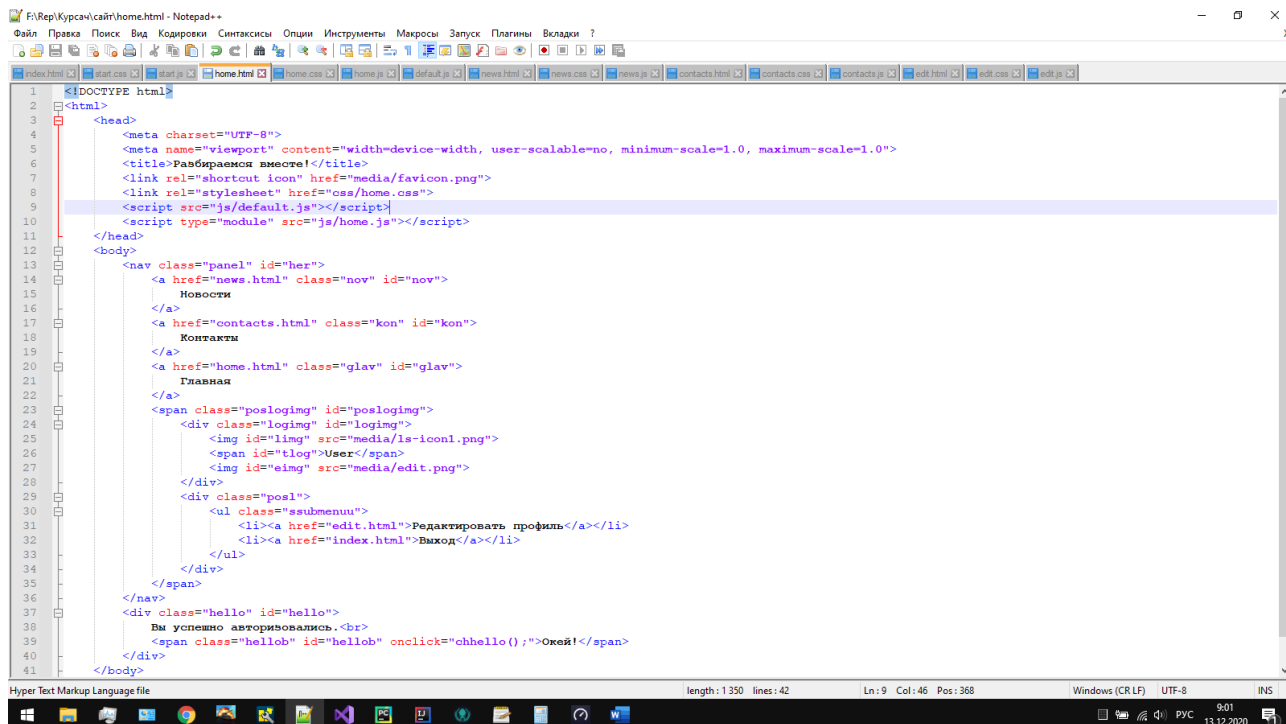


Рисунок 1 – Notepad++

1.3. Языки и технологии, с помощью которых реализовано приложение

Программа написана на популярном языке программирования Java, на версии 17.0.1. Этот язык отлично подходит для разработки и решения разнообразных задач. К его плюсам можно отнести...

1. Независимый код. Любая платформа, которая поддерживает виртуальную машину Java, воспроизведет ваш код
2. Надежный код. Строгая статистическая типизация дает главное преимущество — надежность вашего кода
3. Высокая функциональность. На Java можно написать практически все: от простого приложения на смартфон до программ по машинному обучению для беспилотных автомобилей
4. Синтаксис средней сложности. Данный язык поддается изучению новичкам, которые раньше вообще не имели дела с программированием

5. Java для Android. Android — самая популярная ОС для смартфонов, а Java — самый популярный язык для приложений на Android, соответственно, изучив Java, будет очень широкое поле для деятельности

CSS3 — представляет собой формальный язык, реализованный с помощью языка разметки. Самая масштабная редакция по сравнению с CSS1, CSS2 и CSS2.1. Главной особенностью CSS3 является возможность создавать анимированные элементы без использования JS, поддержка линейных и радиальных градиентов, теней, сглаживания и прочее. Данная технология в проекте используется для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида JavaFX сцен.

Gradle — система автоматической сборки, которую используют для упрощения работы с Java. С помощью (условно) стандартизированных средств она помогает разработчикам собрать нужный продукт без потери его уникальности. Ведь процесс работы с Gradle — не просто выбор шаблона. Данная технология в проекте отвечает, непосредственно за сборку проекта.

Spring Framework — по сути Spring Framework представляет собой просто контейнер внедрения зависимостей, с несколькими удобными слоями. Это все позволяет вам быстрее и удобнее создавать Java-приложения. Но на самом Spring имеет множество проектов. Для нашего проекта интересен только Spring Data JPA, так же при помощи Gradle и Spring удалось все файлы проекта вложить внутрь JAR.

Spring Data JPA — дополнительный удобный механизм для взаимодействия с сущностями базы данных, организации их в репозитории, извлечение данных, изменение, в каких-то случаях для этого будет достаточно объявить интерфейс и метод в нем, без имплементации. В проекте используется для связи PostgreSQL с Java.

JavaFX — это, по сути, инструментальный GUI для Java. JavaFX нацелен на создание игр и настольных приложений на Java. Также, он позволяет нам стилизовать файлы компоновки GUI (XML) и сделать их элегантнее с помощью

CSS. JavaFX дополнительно работает с интегрированной 3D-графикой, а также аудио, видео и встроенными сетевыми приложениями в единый инструментальный GUI... Он прост в освоении и хорошо оптимизирован. Он поддерживает множество операционных систем, а также Windows, UNIX системы и Mac OS. Вызвало удивление, хороший уровень работы с 3D, что позволило перебраться jME3D(java monkey engine) на родную трёхмерную графику. Изначально было затрачено огромное количество времени, на то, чтобы заставить работать jME3D вместе с JavaFX, а потом выяснились фатальные недостатки, связанные с чрезмерным потреблением ресурсов, а также с невозможностью их высвобождения. И при поиске новых библиотек случайно наткнулся на статью, показывающую возможности JavaFX в 3D. Жаль, что это не произошло изначально... В проекте используется для использования сцен и работы с 3D.

Launch4j — ПО позволяющее конвертировать JAR в EXE, с возможностью добавить иконку приложению, информацию об авторе, версии и прочем, указать, какую версию JRE следует использовать, откуда на неё ссылаться. Все настройки хранятся в xml-файле, его описание лежит на сайте программы. Также пытался найти способы, как разместить JRE внутри EXE и воспользоваться им. Но, к сожалению, смог только разместить, но воспользоваться не смог. Я нашёл способы для того, чтобы это можно было реализовать. Но реализация не подвержена к автоматизации процесса. К сожалению, сроки поджимают и не хватает времени поискать информацию в данном направлении. В итоге смог с помощью данного ПО привязать JRE по относительному пути к исполняемому файлу. JRE находится в одной директории с EXE.

PostgreSQL — это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных.

Lombok — проект по добавлению дополнительной функциональности в Java с помощью изменения исходного кода перед Java компиляцией. По сути,

проект Lombok позволяет избавиться от многословности Java в большинстве случаев и перестать писать огромные простыни кода из геттеров, сеттеров, equals, hashCode и toString (да их обычно генерит IDE, но читать и менять все равно приходится программисту), в результате Java становится почти такой же краткой как Kotlin, Scala или C#.

Slf4j — SLF4J расшифровывается как simple logging facade for java. Он обеспечивает простую абстракцию всех каркасов логирования в Java. Таким образом, он позволяет пользователю работать с любой из сред ведения журналов, таких как Log4j, Logback и JUL (java.util.logging), используя единственную зависимость. Вы можете перейти на необходимую инфраструктуру ведения журналов во время выполнения / развертывания.

2. Функциональное назначение

Исходя из заданной темы, приложение сделано про компьютерные комплектующие, а если быть точным, то помогает разобраться в их устройстве и назначении, а также ознакомливает с базовыми понятиями с целью приобретения пользователем поверхностных знаний про составляющие компьютера. Приложение обладает всеми необходимыми свойствами:

1. Интуитивно понятный и дружелюбный к пользователю дизайн
2. Автономность
3. Портативность
4. Частичная независимость от интернета

3. Описание логической структуры

3.1. Анализ предметной области разрабатываемого приложения

Основной стиль приложения сформировался после обзора таких сайтов, как citilink.ru (Рисунок 2), dns-shop.ru (Рисунок 3) и yandex.ru/search/?lr=213&text=gecnj (Рисунок 4).

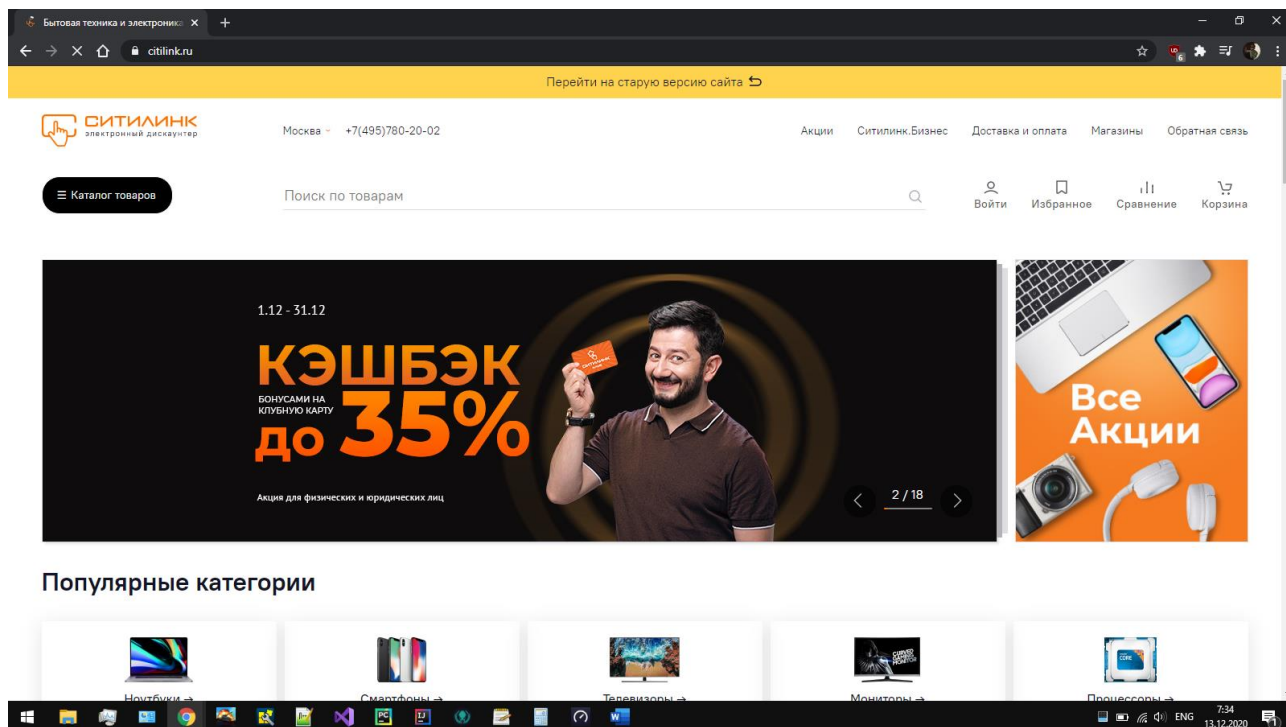


Рисунок 2 – Ситилинк

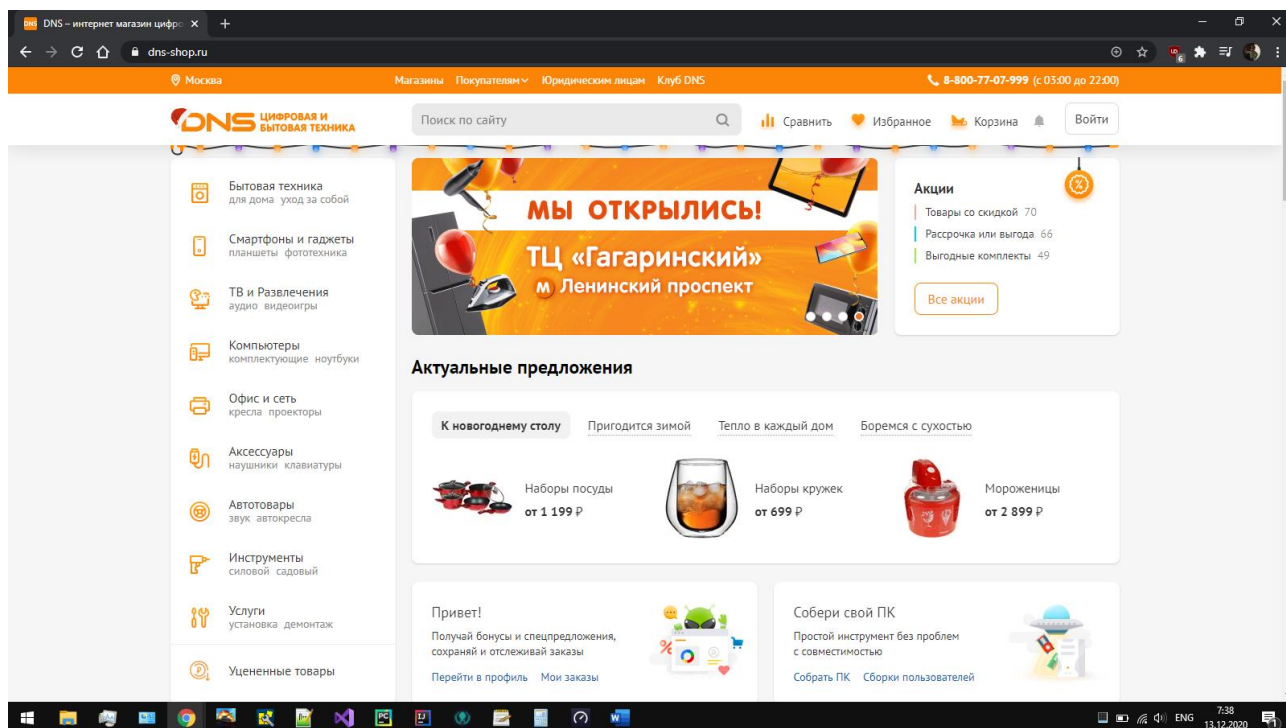


Рисунок 3 – DNS

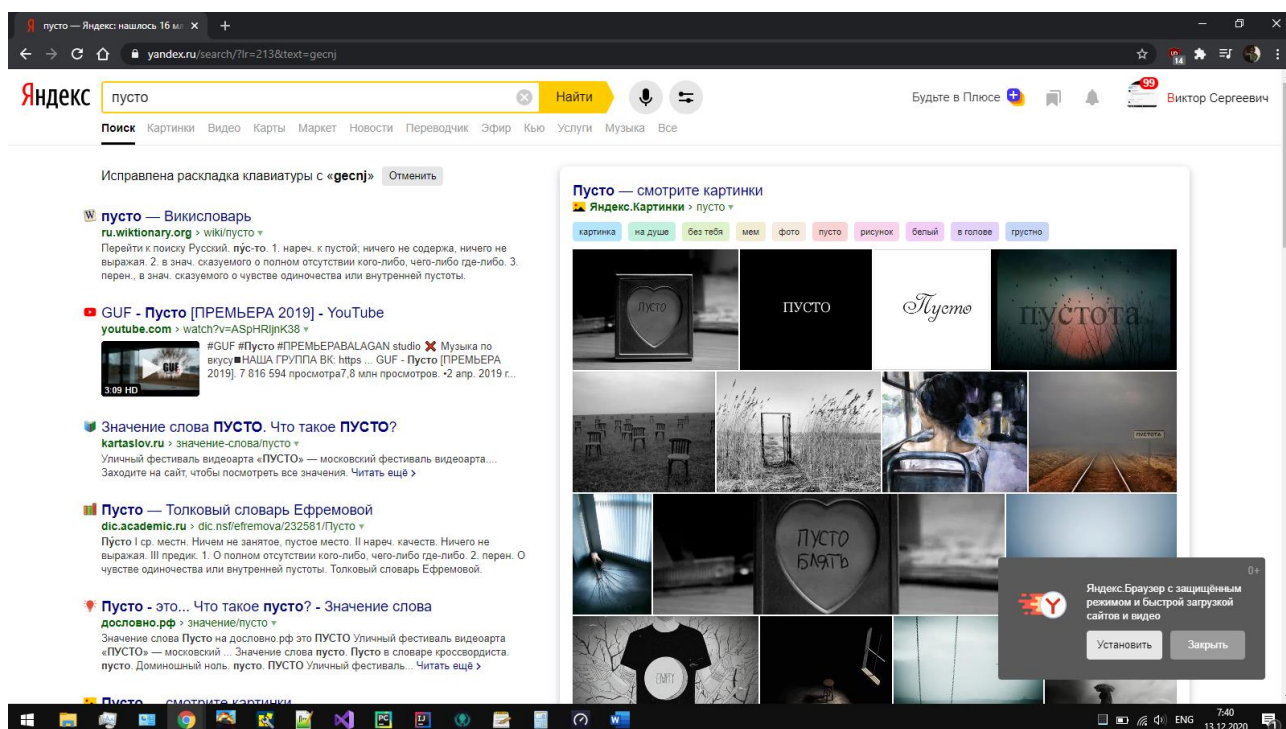


Рисунок 4 – поисковая страница Яндекса

Стили Ситилинка и DNS очень схожи, и я позаимствовал их оранжевой темой. Поисковая страница Яндекса, частично повлияла на оформление панели межстраничной навигации моего приложения.

За текстовый контент отвечает гид с сайта МВидео (Рисунок 5).

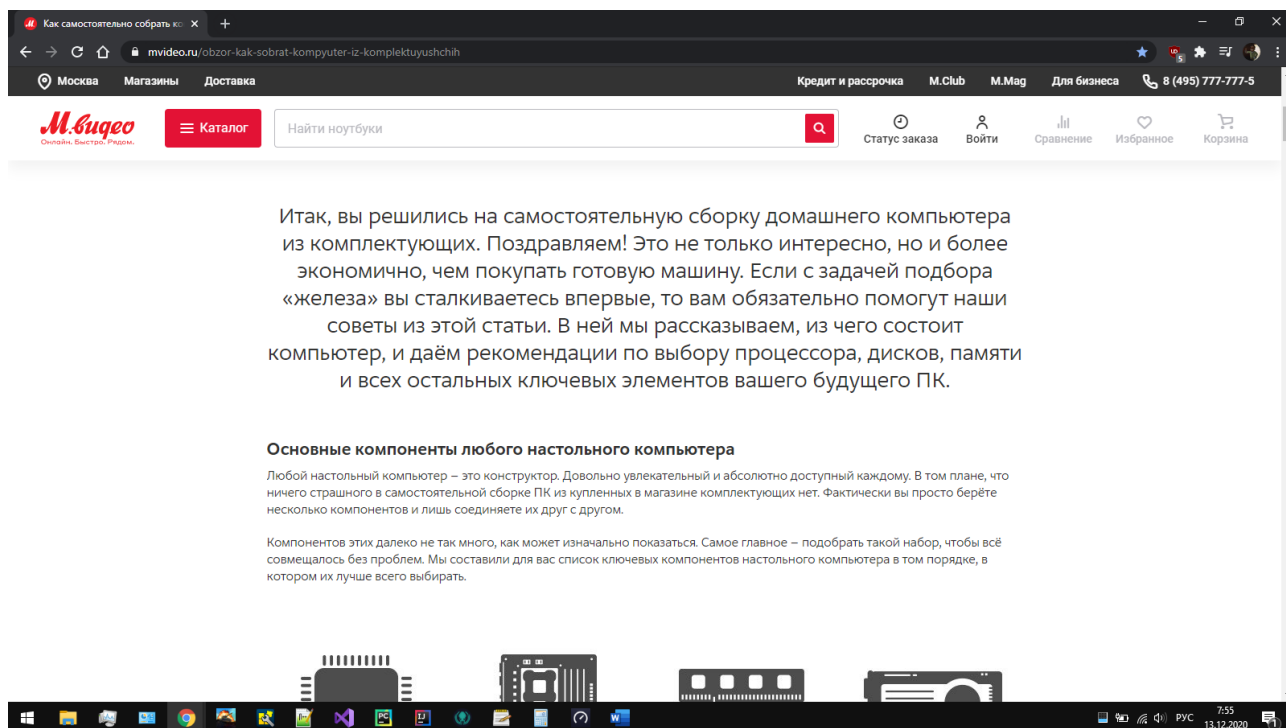


Рисунок 5 - mvideo.ru/obzor-kak-sobrat-kompyuter-iz-komplektuyushchih

Так как пришла идея с 3D наполнением страницы, то пришлось искать и 3D модели. К сожалению, сразу готовых вариантов найти не удалось. Чаще всего приходилось пользоваться дополнительной программой Blender, чтобы оптимизировать модели, если бы я его этого не сделал, то из-за большой детализации, FPS (**F**ramerate **P**er **S**econd) был низким на большинстве слабых устройств и о комфорте на устройствах, таких как смартфоны или же ультрабюджетные ноутбуки. С объёмным контентом мне чаще всего помогали два портала Sketchfab (Рисунок 6) и Turbosquid (Рисунок 7).

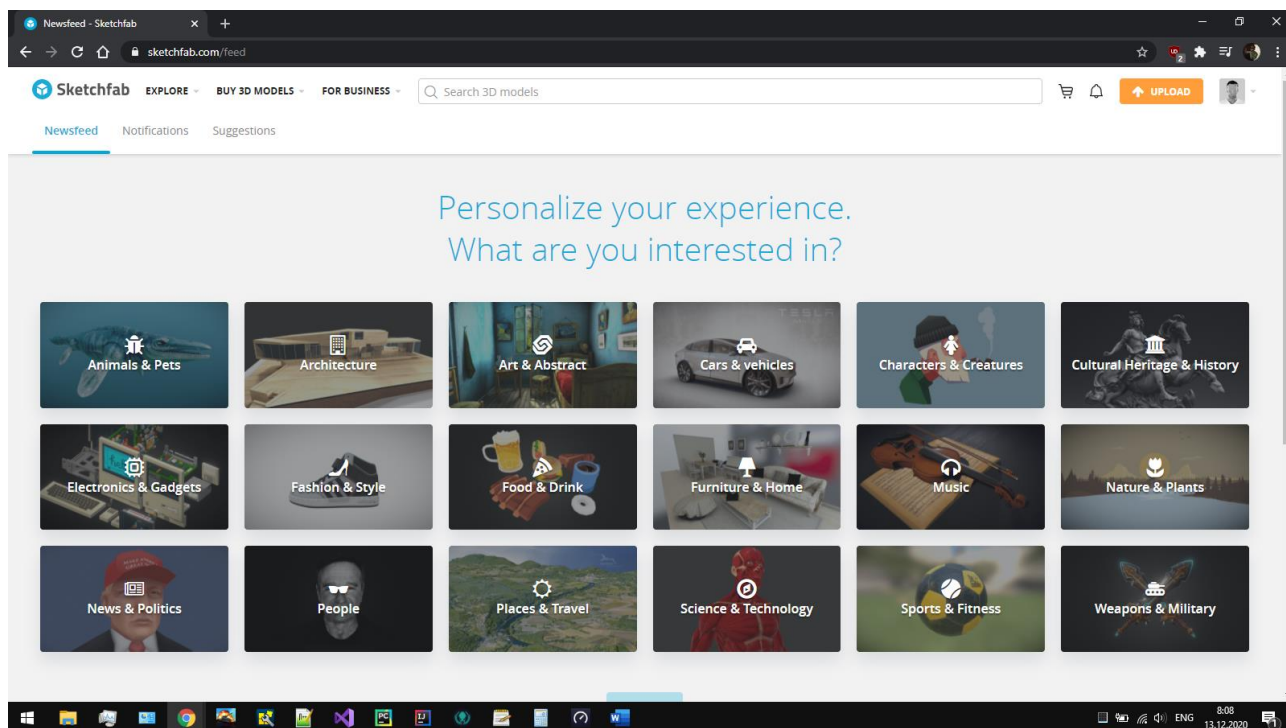


Рисунок 6 - sketchfab.com/feed

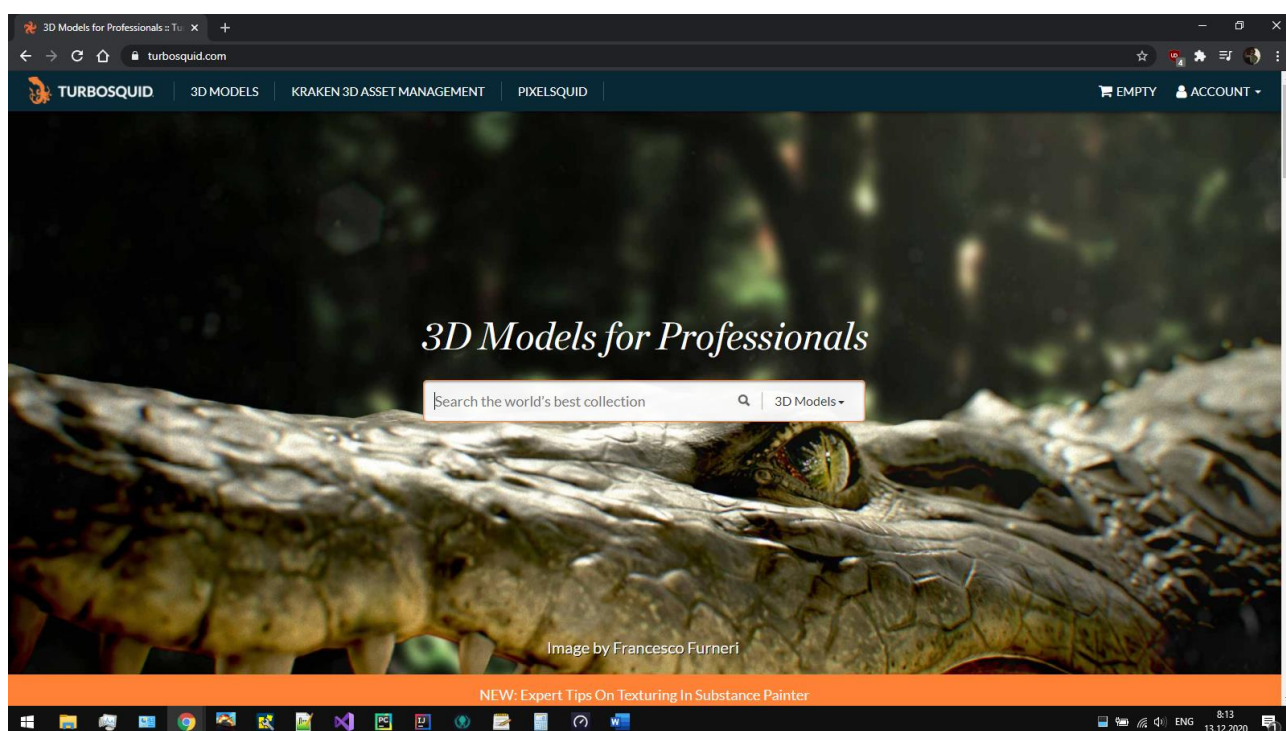


Рисунок 7 – turbosquid.com

3.2. Описание структуры данных с использованием UML диаграмм

3.3.1 Формирование UML диаграммы


```
import org.springframework.context.ConfigurableApplicationContext;

@SpringBootApplication
public class MireaApplication {

    public static ConfigurableApplicationContext ctx;

    public static void main(String[] args) {
        System.setProperty("java.awt.headless", "false");
        Application.launch(Start.class);
    }

}
```

3.3.2 ru.mirea. SetCtx

```
5.      package ru.mirea;

import org.springframework.context.ConfigurableApplicationContext;

public class SetCtx {

    private ConfigurableApplicationContext ct;

    private Start st;

    public SetCtx() {
    }

    public void setCt(ConfigurableApplicationContext ct1) {
        ct = ct1;
        MireaApplication.ctx = ct;
        Start.starts();
    }

}
```

3.3.3 ru.mirea. StartSpring

```
package ru.mirea;

import org.springframework.boot.SpringApplication;

public class StartSpring extends Thread {

    private final SetCtx comp;

    public StartSpring(SetCtx comp) {
        this.comp = comp;
    }

    @Override
    public void run() {
        synchronized (comp) {
            comp.setCt(SpringApplication.run(MireaApplication.class));
        }
    }

}
```

3.3.4 ru.mirea. StartSpring

```
package ru.mirea;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
```

```

import javafx.scene.SceneAntialiasing;
import javafx.scene.image.Image;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;
import javafx.scene.media.MediaView;
import javafx.stage.Stage;
import javafx.stage.WindowEvent;
import lombok.extern.slf4j.Slf4j;
import ru.mirea.Controllers.ModelController;
import ru.mirea.Controllers.StartController;
import ru.mirea.data.User;
import ru.mirea.data.UsersImpl;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.PrintWriter;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.text.SimpleDateFormat;
import java.util.*;

@Slf4j
public class Start extends Application {

    public static Map<String, Object> roots;

    public static FXMLLoader loader;

    public static Stage primStage;

    public static String username;

    public static String path = System.getenv("APPDATA") + "\\CompMagaz\\data";

    public static List<String> off_replace = new ArrayList<>();

    public static List<String> off_save = new ArrayList<>();

    public static List<String> off_reg = new ArrayList<>();

    public static final MediaPlayer player = new MediaPlayer( new
Media(Start.class.getResource("/video/background.mp4").toExternalForm()));

    @Override
    public void start(Stage primaryStage) throws IOException {
        primStage = primaryStage;
        primStage.getIcons().add(new
Image(getClass().getResourceAsStream("/img/ico.png")));
        primStage.setTitle("Веб-сайт «PCBY»");
        primStage.setResizable(false);
        load_users();
        load_commands();
        start_scene("/fxml/start.fxml");
        ((StartController) loader.getController()).onLogos();
        start_spring();
        //close_start();
        //start_scene("/fxml/project.fxml");
        //((ProjController) loader.getController()).init();
    }

    private static List<Integer> cont(List<String> list, String log)

```

```

{
    List<Integer> mas = new ArrayList<>();
    int id = 1;
    for(String str : list)
    {
        if (Objects.equals(str, log)) mas.add(id);
        id++;
    }
    return mas;
}

public static void start_scene(String path) throws IOException {
    loader = new FXMLLoader();
    loader.setLocation(Start.class.getResource(path));
    primStage.setScene(new Scene(loader.load(), 1280, 720, true,
SceneAntialiasing.DISABLED));
    roots = loader.getNamespace();
    primStage.show();
    primStage.setOnCloseRequest(Start::close);
}

private static void close(WindowEvent event) {
    try {
        if (!Files.exists(Paths.get(path))) new File(path).mkdirs();
        PrintWriter out = new PrintWriter(path + "\\users");
        for(User user : UsersImpl.map.values()) out.println(user.toSave());
        out.flush();
        out.close();
        out = new PrintWriter(path + "\\commands");
        out.println(off_replace);
        out.println(off_save);
        out.println(off_reg);
        out.flush();
        out.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    System.exit(0);
}

public static void agree_with_bd()
{
    UsersImpl usersImpl = (UsersImpl)
MireaApplication.ctx.getBean("usersImpl");
    int id = 0;
    Map<String, User> map = new HashMap<>();
    Map<String, User> map1 = new HashMap<>();
    User user;
    List<String> on_reg = new ArrayList<>();
    for(String log : off_reg) {
        user = UsersImpl.map.get(log);
        if(usersImpl.getuser(log) == null)
        {
            usersImpl.addorsave(user);
            on_reg.add(user.getUsername());
        }
    }
    user = null;
    String lg = null;
    for(int i = 0; i < off_replace.size(); i++) {
        String log = off_replace.get(i);
        if(lg == null) {
            lg = log;

```



```

        continue;
    }
    if(log == null) break;
    List<Integer> list = cont(off_replace, log);
    boolean mod = list.get(0) % 2 == 0;
    boolean rem = false;
    for(int df : list)
    {
        if(mod && !(df % 2 == 0))
        {
            Collections.replaceAll(off_replace, log, lg);
            rem = true;
        }
    }
    lg = off_replace.get(i);
    while (off_replace.indexOf(lg) != off_replace.lastIndexOf(lg) &&
rem) off_replace.remove(lg);
    }
    for(String log : off_replace) {
        switch (id) {
            case 0 -> {
                if(!off_reg.contains(log) || on_reg.contains(log)) user =
usersImpl.getUser(log);
                if(user == null) {
                    if (map.containsKey(log))
                        user = map.get(log);
                    else user = UsersImpl.map.get(log);
                }
                if(user != null) map.put(log, user);
            }
            case 1 -> {
                User user1 = UsersImpl.map.get(log);
                if(user1 != null) {
                    user.setId(0);
                    user.setCh_u(Objects.equals(user.getUsername(),
user1.getUsername()));
                    user.setId(user1.getId());
                    String log1 = user.getUsername();
                    user.setUsername(user1.getUsername());
                    user.setPassword(user1.getPassword());
                    user.setIcons(user1.getIcons());
                    user.setSohr(user1.getSohr());
                    map.put(log, user);
                    map1.put(log1, user);
                }
                id = 0;
                user = null;
                continue;
            }
        }
        id++;
    }
    ModelController controller = loader.getController();
    for(String log : off_reg) {
        System.out.println(!on_reg.contains(log) + " " +
!map1.containsKey(log)); //!cont(map1, log));
        if (!on_reg.contains(log) && !map1.containsKey(log)) {
            controller.neWarn(controller.ernull);
            controller.neWarn(controller.ernpat);
            controller.neWarn(controller.gen);
            controller.time_reg.setText(new
SimpleDateFormat("HH:mm").format(new Date()));
            controller.onWarn(controller.net_reg);

```

```

        controller.setPer_reg(log);
        break;
    }
}
for(Map.Entry<String, User> entry : map1.entrySet()) {
    User us = entry.getValue();
    if (usersImpl.getuser(us.getUsername()) != null) {
        if(!us.isCh_u()) {
            controller.neWarn(controller.ernull);
            controller.neWarn(controller.erpat);
            controller.neWarn(controller.gen);
            map1.remove(entry.getKey());
            controller.time_rep.setText(new
SimpleDateFormat("HH:mm").format(new Date()));
            controller.onWarn(controller.net_rep);
            controller.setPer_rep(us.getUsername());
            break;
        } else {
            off_save.add(entry.getKey());
            map1.remove(entry.getKey());
        }
    }
}
for(Map.Entry<String, User> entry : map1.entrySet()) {
    User us = entry.getValue();
    if (usersImpl.getuser(us.getUsername()) == null)
    {
        if(!us.isCh_u()) {
            User us1 = usersImpl.getuser(entry.getKey());
            if(us1 != null)
            {
                us1.setUsername(us.getUsername());
                us1.setPassword(us.getPassword());
                us1.setIcons(us.getIcons());
                us1.setSohr(us.getSohr());
                usersImpl.addorsave(us1);
            } else usersImpl.addorsave(us);
            map1.remove(entry.getKey());
        }
    }
}
off_replace.clear();
for(Map.Entry<String, User> entry : map1.entrySet()) {
    off_replace.addAll(List.of(entry.getKey(),
entry.getValue().getUsername()));
}
for(String log : off_save) {
    User user1 = usersImpl.getuser(log);
    if(map.containsKey(log))
        user = map.get(log);
    else
        user = user1 != null ? user1 : usersImpl.getuser(log);
    user1.setSohr(user.getSohr());
    usersImpl.addorsave(user1);
}
off_save.clear();
}

private static boolean cont(Map<String, User> map, String log)
{
    boolean a = false;
    for(User user : map.values())
    {

```

```

        if (Objects.equals(user.getUsername(), log)) {
            a = true;
            System.out.println(user.getUsername() + " " + log);
            break;
        }
    }
    return a;
}

private void load_commands()
{
    try {
        int id = 0;
        for (String str : Files.readAllLines(Paths.get(path + "\\commands"),
StandardCharsets.UTF_8))
        {
            str = str.replace("[", "").replace("]", "");
            if (str.isEmpty())
            {
                id++;
                continue;
            }
            switch (id) {
                case 0 -> off_replace.addAll(List.of(str.split(", ")));
                case 1 -> off_save.addAll(List.of(str.split(", ")));
                case 2 -> off_reg.addAll(List.of(str.split(", ")));
            }
            id++;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private void load_users()
{
    try {
        int id = 0;
        User newuser = new User();
        for (String str : Files.readAllLines(Paths.get(path + "\\users"),
StandardCharsets.UTF_8))
        {
            switch (id) {
                case 0 -> newuser.setId(Integer.parseInt(str));
                case 1 -> newuser.setUsername(str);
                case 2 -> newuser.setPassword(str);
                case 3 -> newuser.setIcons(Integer.parseInt(str));
                case 4 -> {
                    newuser.setSohr(Integer.parseInt(str));
                    id = 0;
                    UsersImpl.map.put(newuser.getUsername(), newuser);
                    newuser = new User();
                    continue;
                }
            }
            id++;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static void start_spring()

```

```

{
    new StartSpring(new SetCtx()).start();
}

public static void starts() {
    ((StartController) loader.getController()).init();
    start_vid();
}

public static void close_start()
{
    player.stop();
    primStage.close();
}

public static void close_project()
{
    primStage.close();
}

public static void start_vid()
{
    ((MediaView) roots.get("medWiu")).setMediaPlayer(player);
    player.setCycleCount(MediaPlayer.INDEFINITE);
    player.play();
}
}

```

3.3.5 ru.mirea. ThreeD. Xform

```

package ru.mirea.ThreeD;

import javafx.scene.Group;
import javafx.scene.transform.Rotate;
import javafx.scene.transform.Scale;
import javafx.scene.transform.Translate;

public class Xform extends Group {

    public enum RotateOrder {
        XYZ, XZY, YXZ, YZX, ZXY, ZYX
    }

    public Translate t = new Translate();
    public Translate p = new Translate();
    public Translate ip = new Translate();
    public Rotate rx = new Rotate();
    { rx.setAxis(Rotate.X_AXIS); }
    public Rotate ry = new Rotate();
    { ry.setAxis(Rotate.Y_AXIS); }
    public Rotate rz = new Rotate();
    { rz.setAxis(Rotate.Z_AXIS); }
    public Scale s = new Scale();

    public Xform() {
        super();
        getTransforms().addAll(t, rz, ry, rx, s);
    }

    public Xform(RotateOrder rotateOrder) {
        super();
        // choose the order of rotations based on the rotateOrder
        switch (rotateOrder) {
            case XYZ:

```

```

        getTransforms().addAll(t, p, rz, ry, rx, s, ip);
        break;
    case XZY:
        getTransforms().addAll(t, p, ry, rz, rx, s, ip);
        break;
    case YXZ:
        getTransforms().addAll(t, p, rz, rx, ry, s, ip);
        break;
    case YZX:
        getTransforms().addAll(t, p, rx, rz, ry, s, ip); // For Camera
        break;
    case ZXY:
        getTransforms().addAll(t, p, ry, rx, rz, s, ip);
        break;
    case ZYX:
        getTransforms().addAll(t, p, rx, ry, rz, s, ip);
        break;
    }
}

public void setTranslate(double x, double y, double z) {
    t.setX(x);
    t.setY(y);
    t.setZ(z);
}

public void setTranslate(double x, double y) {
    t.setX(x);
    t.setY(y);
}

// Cannot override these methods as they are final:
// public void setTranslateX(double x) { t.setX(x); }
// public void setTranslateY(double y) { t.setY(y); }
// public void setTranslateZ(double z) { t.setZ(z); }
// Use these methods instead:
public void setTx(double x) { t.setX(x); }
public void setTy(double y) { t.setY(y); }
public void setTz(double z) { t.setZ(z); }

public void setRotate(double x, double y, double z) {
    rx.setAngle(x);
    ry.setAngle(y);
    rz.setAngle(z);
}

public void setRotateX(double x) { rx.setAngle(x); }
public void setRotateY(double y) { ry.setAngle(y); }
public void setRotateZ(double z) { rz.setAngle(z); }
public void setRx(double x) { rx.setAngle(x); }
public void setRy(double y) { ry.setAngle(y); }
public void setRz(double z) { rz.setAngle(z); }

public void setScale(double scaleFactor) {
    s.setX(scaleFactor);
    s.setY(scaleFactor);
    s.setZ(scaleFactor);
}

public void setScale(double x, double y, double z) {
    s.setX(x);
    s.setY(y);
    s.setZ(z);
}

```

```

    }

    // Cannot override these methods as they are final:
    // public void setScaleX(double x) { s.setX(x); }
    // public void setScaleY(double y) { s.setY(y); }
    // public void setScaleZ(double z) { s.setZ(z); }
    // Use these methods instead:
    public void setSx(double x) { s.setX(x); }
    public void setSy(double y) { s.setY(y); }
    public void setSz(double z) { s.setZ(z); }

    public void setPivot(double x, double y, double z) {
        p.setX(x);
        p.setY(y);
        p.setZ(z);
        ip.setX(-x);
        ip.setY(-y);
        ip.setZ(-z);
    }

    public void reset() {
        t.setX(0.0);
        t.setY(0.0);
        t.setZ(0.0);
        rx.setAngle(0.0);
        ry.setAngle(0.0);
        rz.setAngle(0.0);
        s.setX(1.0);
        s.setY(1.0);
        s.setZ(1.0);
        p.setX(0.0);
        p.setY(0.0);
        p.setZ(0.0);
        ip.setX(0.0);
        ip.setY(0.0);
        ip.setZ(0.0);
    }

    public void resetTSP() {
        t.setX(0.0);
        t.setY(0.0);
        t.setZ(0.0);
        s.setX(1.0);
        s.setY(1.0);
        s.setZ(1.0);
        p.setX(0.0);
        p.setY(0.0);
        p.setZ(0.0);
        ip.setX(0.0);
        ip.setY(0.0);
        ip.setZ(0.0);
    }
}

```

3.3.6 ru.mirea. ThreeD. Scene3D

```

package ru.mirea.ThreeD;

import com.interactivemesh.jfx.importer.obj.ObjModelImporter;
import javafx.scene.*;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;
import javafx.scene.shape.CullFace;
import javafx.scene.shape.MeshView;

```

```

import lombok.extern.slf4j.Slf4j;

@Slf4j
public class Scene3D {

    private Group root = new Group();
    private final Xform world = new Xform();
    private final PerspectiveCamera camera = new PerspectiveCamera(true);
    private final Xform cameraXform = new Xform();
    private final Xform cameraXform2 = new Xform();
    private final Xform cameraXform3 = new Xform();
    private final double cameraDistance = 450;
    private double mousePosX;
    private double mousePosY;
    private double mouseOldX;
    private double mouseOldY;
    private double mouseDeltaX;
    private double mouseDeltaY;

    private SubScene scene;

    private final Color lightColor = Color.rgb(204, 205, 200);

    public SubScene getScene() {
        return scene;
    }

    public Scene3D(String comp) {
        buildCamera();
        buildScene(comp);
        scene = new SubScene(root, 640, 512, true, SceneAntialiasing.DISABLED);
        scene.setFill(Color.valueOf("#242424"));
        handleMouse();
        scene.setCamera(camera);
    }

    public void destroy(){
        scene.setOnMousePressed(null);
        scene.setOnMouseDragged(null);
        world.getChildren().remove(world.getChildren().get(3));
        cameraXform.ry.setAngle(320.0);
        cameraXform.rx.setAngle(40);
        mouseOldX = 0;
        mouseOldY = 0;
        mousePosX = 0;
        mousePosY = 0;
        mouseDeltaX = 0;
        mouseDeltaY = 0;
        camera.setTranslateZ(-cameraDistance);
        cameraXform2.t.setX(0);
        cameraXform2.t.setY(0);
    }

    public void undestroy(String comp){
        handleMouse();
        buildModel(comp);
    }

    private void buildModel(String comp) {
        Group mesh = new Group();
        ObjModelImporter importer = new ObjModelImporter();
        importer.read(getClass().getResource("/models/" + comp + "/" + comp +
".obj"));
    }

```

```

        for (MeshView view : importer.getImport()) {
            view.setCullFace(CullFace.NONE);
            mesh.getChildren().add(view);
        }
        importer.close();
        world.getChildren().add(mesh);
    }

    private void buildScene(String comp) {
        PointLight pointLight = new PointLight(lightColor);
        pointLight.setLayoutX(0);
        pointLight.setLayoutY(-5000);
        pointLight.setTranslateZ(-1000);

        Color ambientColor = Color.rgb(80, 80, 80, 0);
        AmbientLight ambient = new AmbientLight(ambientColor);
        world.getChildren().add(pointLight);
        world.getChildren().add(ambient);
        root.getChildren().add(world);
        buildModel(comp);
    }

    private void buildCamera() {
        world.getChildren().add(cameraXform);
        cameraXform.getChildren().add(cameraXform2);
        cameraXform2.getChildren().add(cameraXform3);
        cameraXform3.getChildren().add(camera);
        cameraXform3.setRotateZ(180.0);
        cameraXform3.setRotateX(200.0);

        camera.setNearClip(0.1);
        camera.setFarClip(10000.0);
        camera.setTranslateZ(-cameraDistance);
        cameraXform.ry.setAngle(320.0);
        cameraXform.rx.setAngle(40);
    }

    public void handleVR(MouseEvent me) {
        mousePosX = me.getSceneX();
        mousePosY = me.getSceneY();
        mouseOldX = me.getSceneX();
        mouseOldY = me.getSceneY();
    }

    public void handle(MouseEvent me) {
        mouseOldX = mousePosX;
        mouseOldY = mousePosY;
        mousePosX = me.getSceneX();
        mousePosY = me.getSceneY();
        mouseDeltaX = (mousePosX - mouseOldX);
        mouseDeltaY = (mousePosY - mouseOldY);

        double modifier = 2.0;
        double modifierFactor = 0.1;

        if (me.isControlDown()) {
            modifier = 0.1;
        }
        if (me.isShiftDown()) {
            modifier = 10.0;
        }
        if (me.isPrimaryButtonDown()) {
            cameraXform.ry

```



```

        .setAngle(cameraXform.ry.getAngle() - mouseDeltaX *
modifierFactor * modifier * 2.0); // +
        cameraXform.rx
        .setAngle(cameraXform.rx.getAngle() + mouseDeltaY *
modifierFactor * modifier * 2.0); // -
    } else if (me.isSecondaryButtonDown()) {
        double z = camera.getTranslateZ();
        double newZ = z + mouseDeltaX * modifierFactor * modifier;
        camera.setTranslateZ(newZ);
    } else if (me.isMiddleButtonDown()) {
        cameraXform2.t.setX(cameraXform2.t.getX() + mouseDeltaX *
modifierFactor * modifier * 0.3); // -
        cameraXform2.t.setY(cameraXform2.t.getY() + mouseDeltaY *
modifierFactor * modifier * 0.3); // -
    }
}

private void handleMouse() {
    scene.setOnMousePressed(this::handleVR);
    scene.setOnMouseDragged(this::handle);
}
}

```

3.3.7 ru.mirea.data. User

```

package ru.mirea.data;

import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

import javax.persistence.*;

@Entity
@Table(name = "users")
public class User {
    @Id
    @Column(name = "id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "log")
    private String username;

    @Column(name = "par")
    private String password;

    @Column(name = "ico")
    private int icons;

    @Column(name = "shr")
    private int sohr;

    @Transient
    private boolean ch_u = false;

    public boolean conf_auth(String pass)
    {
        return password.equals(pass);
    }
}

```

```

@Override
public String toString() {
    return "User{" +
        "id=" + id +
        ", username='" + username + '\'' +
        ", password='" + password + '\'' +
        ", icons=" + icons +
        ", sohr=" + sohr +
        '}';
}

public String toSave() {
    StringBuilder res = new StringBuilder();

    res.append(id).append("\n").append(username).append("\n").append(password).append("\n").append(icons).append("\n").append(sohr);
    return res.toString();
}
}

```

3.3.8 ru.mirea. data. UserRep

```

package ru.mirea.data;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRep extends JpaRepository<User, Integer> {
    User findByUsername(String username);
}

```

3.3.9 ru.mirea. data. UsersImpl

```

package ru.mirea.data;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import ru.mirea.Controllers.ModelController;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Service
public class UsersImpl {
    private final UserRep userRep;
    public static final Map<String, User> map = new HashMap<>();

    @Autowired
    public UsersImpl(UserRep userRep) {
        this.userRep = userRep;
    }

    public void addorsave(User user) {
        if (ModelController.inet) {
            userRep.save(user);
        }
        map.put(user.getUsername(), user);
    }

    public List<User> getAll() {
        return ModelController.inet ? userRep.findAll() :
        map.values().stream().toList();
    }

    public void delete(int id) {

```

```

        if (ModelController.inet)
            userRep.delete(userRep.getById(id));
    }

    public void delete(String log) {
        if (ModelController.inet)
            userRep.delete(getuser(log));
    }

    public User getuser(String log)
    {
        User user = ModelController.inet ? userRep.findByUsername(log) :
map.get(log);
        if (user != null) map.put(user.getUsername(), user);
        return user;
    }

    public boolean checkpar(String log, String par)
    {
        User user = getuser(log);
        if (user == null) return false;
        return user.conf_auth(par);
    }
}

```

3.3.10 ru.mirea.data.UsersImpl

```

package ru.mirea.Controllers;

import javafx.animation.*;
import javafx.beans.property.BooleanProperty;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.*;
import javafx.scene.effect.DropShadow;
import javafx.scene.effect.Glow;
import javafx.scene.effect.InnerShadow;
import javafx.scene.input.Clipboard;
import javafx.scene.input.ClipboardContent;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.util.Duration;
import ru.mirea.Start;
import ru.mirea.data.User;

import java.awt.*;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Date;
import java.util.List;
import java.util.regex.Pattern;

```

```

public class ModelController{

    @FXML
    public Pane ernull;

    @FXML
    public Pane erpat;

    @FXML
    public Pane gen;

    @FXML
    protected Pane net;

    @FXML
    public Pane net_reg;

    @FXML
    public Pane net_rep;

    @FXML
    protected Label time_pat;

    @FXML
    protected Label time_null;

    @FXML
    protected Label time_gen;

    @FXML
    protected Label time_inet;

    @FXML
    public Label time_reg;

    @FXML
    public Label time_rep;

    @FXML
    protected Label sgerpar;

    @FXML
    protected RadioButton ral;

    @FXML
    protected RadioButton ra2;

    @FXML
    protected RadioButton ra3;

    @FXML
    protected VBox Notif;

    protected int ico = 0;

    protected String per_reg;

    protected String per_rep;

    protected final Interpolator inter = Interpolator.EASE_BOTH;

    protected final ArrayList<String> list_nonlat = new ArrayList<>();

```

```

protected final ArrayList<String> list_null = new ArrayList<>();

public static Boolean inet = false;

public void init()
{
    Timeline timeline = new Timeline(new KeyFrame(Duration.seconds(5),
this::ping));
    timeline.setCycleCount(Animation.INDEFINITE);
    timeline.play();
}

private void ping(ActionEvent actionEvent)
{
    if(ModelController.inet != pingh("https://yandex.ru"))
    {
        ModelController.inet = !ModelController.inet;
        if(ModelController.inet) Start.agree_with_bd();
    }
    if(ModelController.inet)
        neWarn(net);
    else {
        time_inet.setText(new SimpleDateFormat("HH:mm").format(new Date()));
        onWarn(net);
    }
}

public void rad1()
{
    ico = 0;
    ra1.setSelected(true);
}

public void rad2()
{
    ico = 1;
    ra2.setSelected(true);
}

public void rad3()
{
    ico = 2;
    ra3.setSelected(true);
}

public String getPer_reg() {
    return per_reg;
}

public void setPer_reg(String per_reg) {
    this.per_reg = per_reg;
}

public String getPer_rep() {
    return per_rep;
}

public void setPer_rep(String per_rep) {
    this.per_rep = per_rep;
}

public void onNREG() throws IOException {

```

```

        if(this.getClass().getSimpleName().equals("StartController"))
        {
            StartController startController = (StartController) this;
            User user = startController.usersImpl.getuser(per_reg);
            startController.toreg(user);
            startController.down();
        } else {
            ProjController projController = (ProjController) this;
            projController.toBegin();
            StartController startController = Start.loader.getController();
            startController.setPer_rep(per_rep);
            startController.setPer_reg(per_reg);
            startController.neWarn(startController.ernull);
            startController.neWarn(startController.erpat);
            startController.neWarn(startController.gen);
            User user = startController.usersImpl.getuser(per_reg);
            System.out.println(user);
            startController.time_reg.setText(new
SimpleDateFormat("HH:mm").format(new Date()));
            startController.onWarn(startController.net_reg);
            if(this.net_rep.isVisible())
startController.onWarn(startController.net_rep);
            startController.toreg(user);
            startController.down();
        }
    }

    public void onNREP() throws IOException {
        Start.username = per_rep;
        if(this.getClass().getSimpleName().equals("StartController"))
        {
            Start.close_start();
            Start.start_scene("/fxml/project.fxml");
            ProjController projController = Start.loader.getController();
            projController.setPer_rep(per_rep);
            projController.setPer_reg(per_reg);
            projController.neWarn(projController.ernull);
            projController.neWarn(projController.erpat);
            projController.neWarn(projController.gen);
            User user = projController.usersImpl.getuser(Start.username);
            projController.init();
            projController.time_rep.setText(new
SimpleDateFormat("HH:mm").format(new Date()));
            projController.onWarn(projController.net_rep);
            if(this.net_reg.isVisible())
projController.onWarn(projController.net_reg);
            projController.toEdit(user);
        } else {
            ProjController projController = (ProjController) this;
            User user = projController.usersImpl.getuser(Start.username);
            projController.toEdit(user);
        }
    }

    public void neNREG()
    {
        neWarn(net_rep);
        Start.off_reg.remove(per_reg);
    }

    public void neNREP()
    {
        neWarn(net_rep);
    }

```

```

    }

    public void textch(KeyEvent keyEvent)
    {
        TextField text = (TextField) keyEvent.getSource();
        if (text.getText().isEmpty())
        {
            if (!list_null.contains(text.getId())) add_null(text.getId());
            if (list_nonlat.contains(text.getId())) rem_nonlat(text.getId());
        } else {
            rem_null(text.getId());
            if (Pattern.matches("[A-Za-z0-9]+", text.getText()))
                rem_nonlat(text.getId());
            else {
                if (!list_nonlat.contains(text.getId()))
                    add_nonlat(text.getId());
            }
        }
    }

    public boolean pingh(String host) {
        try {
            URL siteURL = new URL(host);
            HttpURLConnection connection = (HttpURLConnection)
siteURL.openConnection();
            connection.setRequestMethod("GET");
            connection.setConnectTimeout(1000);
            connection.connect();
            return connection.getResponseCode() == 200;
        } catch (Exception e) {
            return false;
        }
    }

    public boolean getstat(TextField text)
    {
        if (!list_null.contains(text.getId()) &&
!list_nonlat.contains(text.getId()))
        {
            if (text.getText().isEmpty()) {
                add_null(text.getId());
                return false;
            }
            return true;
        }
        return false;
    }

    private void add_null(String id)
    {
        list_null.add(id);
        redfield((TextField) Start.roots.get(id), 1);
        onWarn(ernull);
        time_null.setText(new SimpleDateFormat("HH:mm").format(new Date()));
    }

    private void rem_null(String id)
    {
        list_null.remove(id);
        if (list_null.isEmpty()) {
            neWarn(ernull);
            if (!list_nonlat.contains(id)) redfield((TextField)
Start.roots.get(id), 0);

```

```

    }
}

private void add_nonlat(String id)
{
    list_nonlat.add(id);
    redfield((TextField) Start.roots.get(id), 1);
    onWarn(erpat);
    time_pat.setText(new SimpleDateFormat("HH:mm").format(new Date()));
}

private void rem_nonlat(String id)
{
    list_nonlat.remove(id);
    if(list_nonlat.isEmpty()) {
        neWarn(erpat);
        if (!list_null.contains(id)) redfield((TextField)
Start.roots.get(id), 0);
    }
}

private void redfield(TextField text, double opacity)
{
    InnerShadow innerShadow = (InnerShadow) text.getEffect();
    DropShadow dropShadow = (DropShadow) innerShadow.getInput();
    List<KeyValue> kv = new ArrayList<>();
    kv.add(new KeyValue(innerShadow.colorProperty(),
Color.web("#ff0000",opacity), inter));
    kv.add(new KeyValue(dropShadow.colorProperty(),
Color.web("#ff0000",opacity), inter));
    played(kv, 100);
}

public void onWarn(Pane pane)
{
    if(!pane.isVisible()) pane.setVisible(true);
    if(pane.getParent() != Notif){
        if(pane.getParent() != null)
((Pane)pane.getParent()).getChildren().remove(pane);
        pane.setTranslateY(-175);
        pane.setOpacity(0);
        Notif.getChildren().add(pane);
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(pane.translateYProperty(), 0, inter));
        kv.add(new KeyValue(pane.opacityProperty(), 1, inter));
        played(kv, 500);
    }
}

public void neWarn(Pane pane)
{
    if(pane.isVisible()) {
        pane.visibleProperty().addListener(this::changed);
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(pane.translateYProperty(), -175, inter));
        kv.add(new KeyValue(pane.opacityProperty(), 0, inter));
        kv.add(new KeyValue(pane.visibleProperty(), false, inter));
        played(kv, 500);
    }
}

private void changed(ObservableValue<? extends Boolean> ov, Boolean old_val,
Boolean new_val) {

```



```

        if(!new_val) {
            Pane pane = (Pane)((BooleanProperty) ov).getBean();
            Notif.getChildren().remove(pane);
            pane.visibleProperty().removeListener(this::changed);
        }
    }

    public void ranpar>PasswordField[] passFields)
    {
        StringBuilder password = new StringBuilder();
        String symbols =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
        for (int i = 0; i < 15; i++){
            password.append(symbols.charAt((int) Math.floor(Math.random() *
symbols.length())));
        }
        for>PasswordField text : passFields){
            text.setText(password.toString());
        }
        sgerpar.setText(password.toString());
        onWarn(gen);
        time_gen.setText(new SimpleDateFormat("HH:mm").format(new Date()));
        Clipboard clipboard = Clipboard.getSystemClipboard();
        ClipboardContent content = new ClipboardContent();
        content.putString(password.toString());
        clipboard.setContent(content);
    }

    public void onBut(MouseEvent mouseEvent)
    {
        Button but = (Button) mouseEvent.getSource();
        InnerShadow innerShadow = (InnerShadow) but.getEffect();
        Glow glow = (Glow) innerShadow.getInput();
        Label label = (Label) but.getChildrenUnmodifiable().get(0);
        DropShadow dropShadow = (DropShadow) label.getEffect();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(innerShadow.offsetYProperty(), 0, inter));
        kv.add(new KeyValue(glow.levelProperty(), 0.3, inter));
        kv.add(new KeyValue(dropShadow.colorProperty(), Color.web("#000",0.5),
inter));
        played(kv, 100);
    }

    public void neBut(MouseEvent mouseEvent)
    {
        Button but = (Button) mouseEvent.getSource();
        InnerShadow innerShadow = (InnerShadow) but.getEffect();
        Glow glow = (Glow) innerShadow.getInput();
        Label label = (Label) but.getChildrenUnmodifiable().get(0);
        DropShadow dropShadow = (DropShadow) label.getEffect();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(innerShadow.offsetYProperty(), -5, inter));
        kv.add(new KeyValue(glow.levelProperty(), 0, inter));
        kv.add(new KeyValue(dropShadow.colorProperty(), Color.web("#000",1),
inter));
        played(kv, 100);
    }

    public void onPRB(MouseEvent mouseEvent)
    {
        Button but = (Button) mouseEvent.getSource();
        InnerShadow innerShadow = (InnerShadow) but.getEffect();
        Glow glow = (Glow) innerShadow.getInput();

```

```

        Label label = (Label) but.getChildrenUnmodifiable().get(0);
        DropShadow dropShadow = (DropShadow) label.getEffect();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(innerShadow.offsetYProperty(), 0, inter));
        kv.add(new KeyValue(innerShadow.widthProperty(), 60, inter));
        kv.add(new KeyValue(innerShadow.heightProperty(), 60, inter));
        kv.add(new KeyValue(innerShadow.colorProperty(), Color.web("#000",1),
inter));
        kv.add(new KeyValue(glow.levelProperty(), 0, inter));
        kv.add(new KeyValue(dropShadow.colorProperty(), Color.web("#000",0.5),
inter));
        played(kv, 100);
    }

    public void nePRB(MouseEvent mouseEvent)
    {
        Button but = (Button) mouseEvent.getSource();
        InnerShadow innerShadow = (InnerShadow) but.getEffect();
        Glow glow = (Glow) innerShadow.getInput();
        Label label = (Label) but.getChildrenUnmodifiable().get(0);
        DropShadow dropShadow = (DropShadow) label.getEffect();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(innerShadow.offsetYProperty(), -5, inter));
        kv.add(new KeyValue(innerShadow.widthProperty(), 0, inter));
        kv.add(new KeyValue(innerShadow.heightProperty(), 1, inter));
        kv.add(new KeyValue(innerShadow.colorProperty(), Color.web("#fff",0.25),
inter));
        kv.add(new KeyValue(glow.levelProperty(), 0, inter));
        kv.add(new KeyValue(dropShadow.colorProperty(), Color.web("#000",1),
inter));
        played(kv, 100);
    }

    public void onLink(MouseEvent mouseEvent)
    {
        Hyperlink label = (Hyperlink) mouseEvent.getSource();
        Glow glow = (Glow) label.getEffect();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(glow.levelProperty(), 1.0, inter));
        played(kv, 100);
    }

    public void neLink(MouseEvent mouseEvent)
    {
        Hyperlink label = (Hyperlink) mouseEvent.getSource();
        Glow glow = (Glow) label.getEffect();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(glow.levelProperty(), 0, inter));
        played(kv, 100);
    }

    public void onPL(MouseEvent mouseEvent)
    {
        Hyperlink label = (Hyperlink) mouseEvent.getSource();
        Glow glow = (Glow) label.getEffect();
        InnerShadow innerShadow = (InnerShadow)
((DropShadow)glow.getInput()).getInput();
        List<KeyValue> kv = new ArrayList<>();
        kv.add(new KeyValue(glow.levelProperty(), 0, inter));
        kv.add(new KeyValue(innerShadow.chokeProperty(), 0.3, inter));
        played(kv, 100);
    }

```

```

public void nePL(MouseEvent mouseEvent)
{
    Hyperlink label = (Hyperlink) mouseEvent.getSource();
    InnerShadow innerShadow = (InnerShadow)
((DropShadow) ((Glow) label.getEffect()).getInput()).getInput();
    List<KeyValue> kv = new ArrayList<>();
    kv.add(new KeyValue(innerShadow.chokeProperty(), 0, inter));
    played(kv, 100);
}

public void handle1(ActionEvent event)
{
    Timeline time1 = (Timeline) event.getSource();
    time1.stop();
    time1.getKeyFrames().remove(0, time1.getKeyFrames().size());
    time1 = null;
}

public void played(Collection<KeyValue> kv, double time)
{
    Timeline time1 = new Timeline();
    for(KeyValue key : kv)
    {
        time1.getKeyFrames().add(new KeyFrame(Duration.millis(time), key));
    }
    time1.setOnFinished(this::handle1);
    time1.play();
}

public void browse_git() throws URISyntaxException, IOException {
    Desktop d = Desktop.getDesktop();
    d.browse(new URI("https://github.com/Ddudde/Kursach-Java"));
}
}

```

Заключение

Текущее приложение отвечает всем требованиям и является хорошим дополнением к изначальному сайту.

В ходе выполнения курсового проекта были достигнуты следующие задачи:

1. Был проведён анализ предметной области.
2. Обоснован выбор технологий разработки приложения
3. Проведён анализ сравнения с существующими аналогами.
4. Разработана клиентская часть приложения (принятие и обработка запросов)
5. Создана презентация по выполненной курсовой работе.

Все пункты, указанные в техническом задании, работают корректно и в полной мере. Исходя из всего вышеперечисленного, можно сделать вывод, что данный курсовой проект выполнен успешно.

Список информационных источников

1. Java Documentation [Электронный ресурс] – URL: <https://docs.oracle.com/en/java/> (Первое обращение 25.06.2020)
2. Руководство по JavaFX [Электронный ресурс] – URL: <https://metanit.com/java/javafx/> (Первое обращение 25.06.2020)
3. Справочник по CSS [Электронный ресурс] – URL: <http://htmlbook.ru/css> (Первое обращение 25.09.2020)
4. Хабитуев, Б. В. Программирование на языке Java: практикум : учебное пособие / Б. В. Хабитуев. — Улан-Удэ : БГУ, 2020. — 94 с. — ISBN 978-5-9793-1548-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/171791> (дата обращения: 25.05.2021). — Режим доступа: для авториз. пользователей.
5. Руководство по Spring [Электронный ресурс]. URL: <https://proselyte.net/tutorials/spring-tutorial-full-version/> (дата обращения: 29.04.21). Заголовок с экрана.
6. Гуськова, О. И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. — Москва : МПГУ, 2018. — 240 с. — ISBN 978-5-4263-0648-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/122311> (дата обращения: 25.06.2021). — Режим доступа: для авториз. пользователей.

Приложения

Приложение об исходном коде приложения

Исходный код данного приложения, включающий в себя все файлы, является слишком объемным, чтобы его было уместно прикрепить в качестве приложения, поэтому он прилагается в качестве ссылки на гитхаб.

<https://github.com/Ddudde/Kursach-Java>