**Введение**

На учебное проектирование была поставлена задача – разработать сайт для продажи автомобилей.

Цель учебного проекта заключается в разработке сайта, при помощи которого пользователь может приобрести автомобиль.

Созданная программа рассчитана для пользователей любого возраста, которые хотят приобрести автомобиль.

Приведём краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нём можно ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи. Также в этом разделе можно узнать о том, как данная задача решается в настоящее время. Описаны все входные и выходные данные. В подразделе «Инструменты обработки» рассмотрена среда, в которой был создан курсовой проект. Здесь также установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачи.

В разделе «Проектирование задачи» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе описан пользовательский интерфейс, составление алгоритмов процесса обработки информации, описана разработка системы справочной информации.

«Реализация задачи» − это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые использованы при реализации данного приложения. В этом разделе описаны функции пользователя и их структура. Также здесь можно увидеть таблицу, в которой представлена полная аннотация файлов, используемых в данном проекте.

Четвёртый раздел – «Тестирование». В нём описано полное и функциональное тестирование данной программы, т.е. протестирован каждый пункт меню, каждая операция, которая выполняется приложением. Смоделированы всевозможные действия пользователя при работе с программой, начиная от запуска до выхода.

Пятый раздел – это «Руководство пользователя», в котором описана подробная инструкция пользованием этого ПП для будущих пользователей интернет-ресурсом.

«Заключение» содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В разделе «Список используемых источников» приведён список используемых при разработке источников.

В приложениях к пояснительной записке приведены виды диаграмм и пользовательского интерфейса.

**1 Анализ задачи**

* 1. **Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

Наименование задачи: разработка сайта для продажи автомобилей;

Цель разработки: дать возможность пользователю покупать автомобили;

Назначение: данный программный продукт разрабатывается для людей любого возраста, желающим приобрести автомобиль или выставить свой автомобиль на продажу.

Периодичность использования: по мере необходимости.

Источники и способы получения данных: справочная служба.

Обзор существующих аналогичных ПП: в качестве примера рассмотрим сайт <https://av.by>.

**1.1.2 Организационно-экономическая сущность задачи**

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Администратор:

1. Редактирование страниц сайта.

2. Вход в административную панель.

Гость:

1. Просмотр автомобилей.

2. Регистрация.

Пользователь:

1. Просмотр автомобилей.

2. Вход в учетную запись.

3. Добавление автомобиля на продажу.

4. Приобретение автомобиля.

**1.1.3 Описание исходной (входной) и результативной (выходной) информации**

Таблица 1 – Функции программы с описанием с входной, выходной и условно постоянной информации

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Категория пользователей | Наименование процесса | Краткое описание алгоритма  выполнения  процесса | Входная информация | Выходная информация | Условно-постоянная информация |
| 1 | Пользователь | Фильтр списка автомобилей | Пользователь настраивает подбор автомобиля из списка | Данные которые вводит пользователь в фильтр | Автомобили подходящие по фильтру | Данные автомобилей |
| 2 | Администратор | Редактирование страниц сайта | Администратор редактирует код сайта | Новые данные страницы | Обновление страниц сайта | Отсутствует |
| 3 | Пользователь | Выставление автомобиля на продажу в список | Пользователь вводит информацию о своем автомобиле и цену за которую он хочет продать автомобиль | Данные об автомобиле | Обновление списка автомобилей | Данные автомобиля |

**1.1.4 Нефункциональные требования**

Требования к применению: помогает приобрести автомобиль или выставить свой автомобиль на продажу.

Требования к реализации: Для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS. Для реализации интерактивных элементов клиентской части должны использоваться языки Java.

Требования к надежности: Система может быть недоступна не более чем 24 часа в год. У администратора сайта должна быть возможность выгрузить и загрузить копию сайта.

Требования к интерфейсу: При разработке сайта должны быть использованы преимущественного красные/оранжевые оттенки. Основные разделы сайта должны быть доступны с первой страницы. Грамотный пользовательский интерфейс. Сайт должен адаптироваться под компьютер, телефон и планшет.

* 1. **Диаграмма вариантов использования**

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования.

Актером или действующим лицом является любая сущность, взаимодействующая с системой извне (Рисунок 1). Это может быть, как живое существо, так и любая друга система, которая может служить источником воздействия на моделируемую систему так, как определяет сам разработчик.

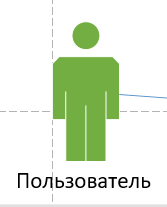


Рисунок 1 – Графическое обозначение актера

Вариант использования является стандартным языком UML и применяется для спецификаций общих особенностей системы и любой другой сущности. Отдельные варианты использования обозначаются на диаграмме эллипсом, в котором содержится его краткое название (Рисунок 2).

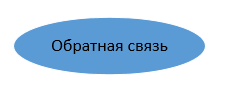


Рисунок 2 – Графическое обозначение вариантов использования

Отношение ассоциации является главным понятием языка UML и используется при построении всех графических моделей. Оно служит для обозначения роли актера и отдельном варианте использования. На диаграмме отношение ассоциации обозначается сплошной линией между актером и вариантом использования (Рисунок 3).



Рисунок 3 – Графическое обозначение отношения ассоциации

Для отображения взаимосвязи экземпляра отдельного варианта использования с более общим вариантом, используется отношение расширения, обозначаемое направленной пунктирной линией со стрелкой от исходного варианта. Данная линия помечается ключевым словом <<extend>> (Рисунок 4).



Рисунок 4 – Графическое обозначение отношения расширения

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данная линия помечается ключевым словом <<include>> (Рисунок 5).



Рисунок 5 – Графическое обозначение отношения включения

Определяя для выбранного актера варианты использования и устанавливая отношения между вариантами использования, получим полную диаграмму вариантов использования, её можно увидеть в Приложении А.

**1.3 Модель данных**

Целью моделирования данных является обеспечение разработчика концептуальной схемой базы данных в форме одной модели и нескольких локальных моделей. Наиболее распространенным средством моделирования данных является диаграмм «сущность-связь».

Сущность – объект, имеющий существенное значение для проектируемой предметной области, информация которой подлежит хранению.

Приложение будет содержать 5 сущностей полный список с описанием приведены в Таблице 2.

Таблица 2 – Сущности

|  |  |  |
| --- | --- | --- |
| № | Сущность | Описание |
| 1 | Покупатель | Содержит информацию об покупателях |
| 2 | Автомобиль | Содержит информацию об автомобилях |
| 3 | Категория | Содержит информацию о категории автомобиля |
| 4 | Сделка о покупке | Содержит информацию о покупке автомобиля |
| 5 | Продавец | Содержит информацию об продавцах |

Атрибут – характеристика сущности, предназначенная для идентификации, классификации, количественной характеристики или свойств, ассоциируемых с множеством других объектов.

Экземпляр – это определение характеристики отдельного элемента множества. Например, сущность «Категория» будет содержать следующие атрибуты (Рисунок 6):



Рисунок 6 – Графическое изображение сущности

Экземпляр атрибута – определяется типом характеристик и его значения.

Связь – это ассоциация между двумя сущностями. Связь может быть один-к-одному, один-ко-многим. Пример связи один-ко-многим-представлен ниже (Рисунок 7):

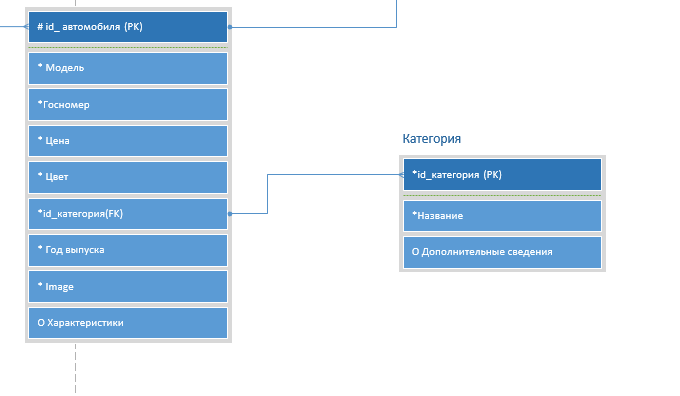


Рисунок 7 – Графическое изображение связи один-ко-многим

Таким образом, определяя атрибуты для каждой сущности и устанавливая связи между ними, получим модель данных, представленную в Приложении Б.

**1.4 Техническое задание**

**1.4.1 Введение**

Наименованием программного продукта является сайт для продажи автомобилей.

Его ответственным лицом является разработчик Борис И.Д.

Срок сдачи программного продукта – 4 января 2021 года.

Сама программа предназначена для людей, которым необходимо приобрести автомобиль.

**1.4.2 Основания для разработки**

**1.4.2.1 Основания для проведения разработки**

Основание для разработки является задание на практику на тему «Разработка сайта для продажи автомобиля». Тема согласована с преподавателем, именуемым в дальнейшем заказчиком.

**1.4.2.2 Наименование и условное обозначение темы разработки**

Наименование темы разработки – «Разработка сайта для продажи автомобиля». Условное обозначение темы разработки (шифр темы) – «РСДПА».

**1.4.3 Требования к программе или программному изделию**

**1.4.3.1 Требования к функциональным характеристикам**

Программа должна обеспечивать возможность выполнения следующих функций гостя:

* Просмотр страниц;
* Просмотр цен;
* Просмотр новостей;
* Просмотр автомобилей.

Администратор (права администратора представлены всему персоналу

организации):

1) Добавление новой информации (новости)

2) Удаление информации (удаление ненужной или устаревшей

информации);

3) Редактирование новой информации (в случае опечатки или для того, чтобы дополнить информацию);

**1.4.4 Исходные данные**

Входные данные будут вноситься в базу данных, организованные в файлы формата СУБД MySQL.

База данных «Автомобили»

Данная база содержит следующие таблицы:

* Таблица «Покупатель»;
* Таблица «Продавец»;
* Таблица «Автомобиль»;
* Таблица «Категория»;
* Таблица «Сделка о покупке».

Ознакомиться можно в разделе «1.3 Модель данных», Приложении Б.

**1.4.5 Требование к надежности**

Система может быть недоступна не более чем 24 часа в год. Резервирование данных осуществляет хостинг-провайдер. У администратора сайта должна быть возможность выгрузить и загрузить копию сайта.

Программа поставляется в виде веб-ресурса – на хостинге. Хостинг позволяет создавать и хранить контент на сайте. Безопасный и бесплатный хостинг Wix позволяет сосредоточиться на самом важном — создании сайта.

Все данные сайта защищены согласно международным стандартам.

Хостинг автоматически включается в момент публикации сайта. Дополнительные настройки не требуются.

Wix включает до 500 МБ трафика. Если подключить [премиум-план](https://ru.wix.com/upgrade/website), то получите до 50 ГБ в облачном хранилище и неограниченную пропускную способность.

**1.4.5.1 Требование к обеспечению надежного (устойчивого) функционирования программы**

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

Организацией бесперебойного питания технических средств;

Использование лицензионного программного обеспечения;

Регулярным выполнением рекомендаций Министерства труда и социального развития;

Регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

Требования к обеспечению надежного (устойчивого) функционирования программы не предъявляются.

**1.4.5.2 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30 минут при условии соблюдения условий технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

**1.4.5.3 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

**1.4.6 Условия эксплуатации**

**1.4.6.1 Требование к видам обслуживания**

Веб-сайт требует проведение каких-либо видов обслуживания. Администратор будет обновлять, редактировать и удалять какую-то информацию на сайте.

**1.4.6.2 Требование к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы сайта, должно составлять не менее 1 штатной единицы – системный администратор.

**1.4.6.3 Требование к составу и параметрам технических средств**

Доступ на данный сайт может осуществляться с помощью любого компьютера и мобильного устройства. Для хорошей работы сайт нужен устойчивое интернет-соединение.

**1.4.7.1 Требования к представлению сайта**

Для организации эффективной работы пользователя нужно создать целостный программный продукт данной предметной области, в котором все компоненты будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Ознакомиться можно в разделе «2.4 Разработка пользовательского интерфейса», Приложении В.

**1.4.7.2 Требования к структуре сайта**

Все названия разделов сайта, приведенные ниже, являются условными и могут корректироваться по соглашению с Заказчиком в ходе проектирования. При помощи системы управления сайтом структура и состав разделов сайта в дальнейшем могут быть изменены и дополнены. Ознакомиться можно в Приложении Г.

**1.4.8 Требование к программной документации**

**1.4.8.1 Предварительный состав программной документации**

Состав программной документации должен включать в себя:

* Постановка задачи;
* Модель данных;
* Диаграмма вариантов использования;
* Диаграмма последовательности;
* Диаграмма состояний;
* Техническое задание.

Ознакомится можно в разделах «1.1 Постановка задачи», «1.3 Модель данных», «1.4 Техническое задание», «2.3 Разработка UML-диаграмм», Приложениях А, Б.

**1.4.9 Технико-экономические показатели**

Ориентировочная экономическая эффективность не рассчитываются.

Предполагаемое число использований программы в год – 365 сеансов работы на одном рабочем месте.

**1.4.10 Стадии и этапы разработки**

Для отслеживания прогресса и анализа объема работ был использован веб-ресурс Wrike. С его помощью можно быстро составить план разработки проекта и отметить степень его готовности. Подробнее ознакомиться можно в разделе «2.6 Разработка плана работы над проектом», Приложение Г.

**1.4.10.1 Стадии разработки**

Разработка должна быть проведена в три стадии:

* Разработка технического задания;
* Рабочее проектирование;
* Внедрение.

**1.4.10.2 Этапы разработки**

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

* Разработка программы;
* Разработка программной документации;
* Испытание программы.

**2 Проектирование задачи**

**2.1 Выбор стратегии разработки и модели жизненного цикла**

**Рассматриваемая процедура состоит из следующей последовательности шагов:**

***1-й шаг****.* Проанализировать отличительные черты проекта по критериям категорий, представленным в виде вопросов.

***2-й шаг****.* Ответить на вопросы по анализируемому проекту, отметив слова «да» или «нет» в соответствующих строках табл. 3.1 – 3.4. Если слов «да» или «нет» в строке несколько, необходимо отметить все из них (все «да» или все «нет»).

В качестве примера в табл. 3.1 выделены варианты ответов для проекта разработки сложного и критичного программного средства, требования к которому заранее не известны и будут уточняться по ходу разработки.

***3-й шаг****.* Расположить по степени важности категории (таблицы) и/или критерии, относящиеся к каждой категории (вопросы внутри таблиц), относительно проекта, для которого выбирается модель ЖЦ.

***4-й шаг.***Выбрать из моделей (см. табл. 3.1 – 3.4) ту модель, которая соответствует столбцу с наибольшим количеством отмеченных ответов с учетом их степени важности (с наибольшим количеством отмеченных ответов в верхней части приоритетных таблиц). Выбранная модель ЖЦ является наиболее приемлемой для анализируемого проекта.

Таблица 3.1 – Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Продолжение Таблицы 3.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2 | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3 | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4 | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| 5 | Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6 | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |

Продолжение Таблицы 3.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Каскадная – 2, V-образная – 2, RAD – 5, инкрементная – 3, быстрого прототипирования – 5, эволюционная – 5.

На основе результатов заполнения табл. 3.1 наиболее подходящей является RAD, прототипирования и эволюционная модель.

Таблица 3.2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2 | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 3 | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4 | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |

Продолжение Таблицы 3.2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6 | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

Каскадная – 4, V-образная – 4, RAD – 5, инкрементная – 4, быстрого прототипирования – 3, эволюционная – 2.

На основе результатов заполнения табл. 3.2 наиболее подходящими являются RAD модель.

Таблица 3.3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |

Каскадная – 1, V-образная – 1, RAD – 1, инкрементная – 2, быстрого прототипирования – 3, эволюционная – 3.

На основе результатов заполнения табл. 3.3 наиболее подходящими являются каскадная и эволюционная модели.

Таблица 3.4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2 | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3 | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4 | Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5 | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6 | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7 | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8 | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |

Продолжение Таблицы 3.4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10 | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Каскадная – 3, V-образная – 4, RAD – 2, инкрементная – 6, быстрого прототипирования – 6, эволюционная – 8.

На основе результатов заполнения табл. 3.4 наиболее подходящей является эволюционная модель.

Исходя из результатов заполнения табл. 3.1 – 3.4 (каскадная – 10, V-образная – 11, RAD – 13, инкрементная – 15, быстрого прототипирования – 17, эволюционная – 18) наиболее подходящей моделью, для разработки данного ПП, является эволюционная модель.

**2.2 Инструменты разработки**

Проект будет разрабатываться в среде программирования Java, т.к. Java — строго типизированный объектно-ориентированный язык программирования общего назначения.

Выгоды от проектирования в среде Windows с помощью Java:

* устраняется необходимость в повторном вводе данных;
* обеспечивается согласованность проекта и его реализации;
* увеличивается производительность разработки и переносимость программ.

Для оптимального функционирования разрабатываемого приложения необходимо наличие у компьютера следующих параметров:

− процессор AMD Ryzen 7 4800H;

− объем ОЗУ 8 ГБ;

− объем места на SSD 512 ГБ;

− ОС – Windows 10.

Итак, для разработки программы необходимы:

− Intellij Idea – главная среда разработки программы, в которой прописывается код;

− Google-браузер – нужен для нахождения информации;

− Word 2016 – нужен для написания пояснительной записки;

− Figma – среда для разработки интерфейса;

- Visio 2016 – для написания диаграммы Ганта.

**2.3 Разработка UML-диаграмм**

В ходе создания проекта требовалось создать две UML-диаграммы, что помогли бы понять определенные процессы самого интернет-ресурса. Этими диаграммами являются диаграмма последовательности и диаграмма деятельности.

Диаграмма последовательности (англ. sequence diagram) — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента (Рисунок 8).

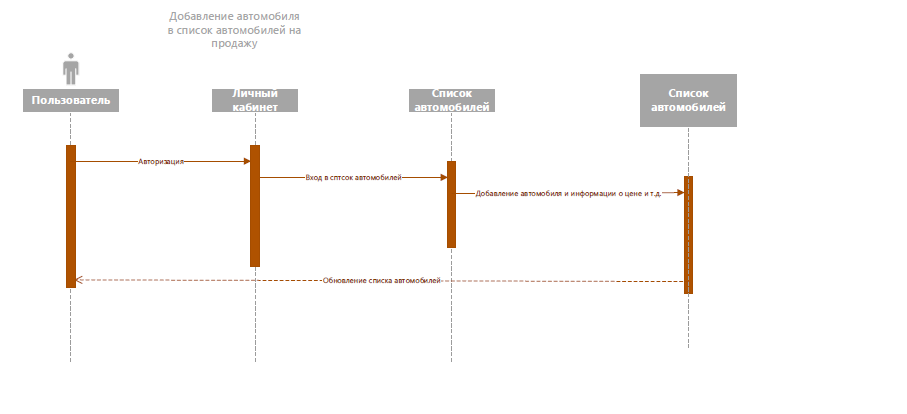


Рисунок 8 – Диаграмма последовательности

Диаграмма состояний — это, по существу, [диаграмма состояний](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B9_(%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%BE%D0%B2)) из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до [бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81) (Рисунок 9).



Рисунок 9 – Диаграмма состояний

**2.4 Разработка пользовательского интерфейса**

Пользовательский интерфейс - это набор программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги. Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи. Каждый диалог состоит из отдельных процессов ввода/вывода, которые физически обеспечивают связь пользователя и компьютера. Обмен информацией осуществляется передачей сообщения.

В случае создания пользовательского интерфейса данного интернет-ресурса были созданы основные страницы сайта, а также гайд по стилизации этих страниц сайта.

Гайд по стилизации: в основном текст имеет два вида шрифтов – Regular и Bold. Размеры шрифта используются разные, в диапазоне от 15 пикс. до 54 пикс. Размеры шрифта от 15 до 18 используются в основном для описания иль неких деталей, остальные вариации идут уже на заголовки или вывески.

Цвета для самого оформления сайта – это белый фон и синий фон для подвала.

Сам пользовательский интерфейс можно увидеть в Приложении В.

**2.5 Тесты на использование**

В ходе разработки программного продукта были составлены тесты, которые необходимо выполнить в дальнейшем. Тесты составлены таким образом, чтобы предусмотреть максимальное количество возможных действий.

Таблица 4 – Тесты на использование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название теста | Действия | Исходная информация | Ожидаемая инфор-  мация |
| 1 | 2 | 3 | 4 | 5 |
| Т1 | Просмотр страниц | Нажатие кнопок на сайте | - | Просмотр страниц |
| Т2 | Регистрация | Ввод необходимых данных для регистрации | Данные пользователя | Успешная регистрация |
| Т3 | Удаление автомобиля | Удаление автомобиля из списка на продажу | - | Успешное удаление автомобиля |

Продолжение Таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Т4 | Редактирование профиля | Вход в личный кабинет и изменения имени, фамилии и т.д. | Данные пользователя | Успешное редактирование профиля |
| Т5 | Добавление автомобиля | Ввод данных автомобиля | Данные об автомобиле | Успешное добавление автомобиля |

**2.6 Разработка плана над проектом**

Разработка плана над проектом была выполнена в виде диаграммы Ганта.

Диагра́мма Га́нта (англ. Gantt chart, также ленточная диаграмма, график Ганта, календарный график) — это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по управлению проектами.

По сути, диаграмма Ганта состоит из полос, ориентированных вдоль оси времени. Каждая полоса на диаграмме представляет отдельную задачу в составе проекта (вид работы), её концы — моменты начала и завершения работы, её протяженность — длительность работы. Вертикальной осью диаграммы служит перечень задач. Кроме того, на диаграмме могут быть отмечены совокупные задачи, проценты завершения, указатели последовательности и зависимости работ, метки ключевых моментов (вехи), метка текущего момента времени «Сегодня» и др. Пример представлен ниже (Рисунок 10):

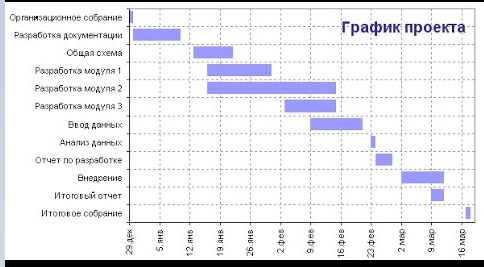
****

Рисунок 10 – Диаграмма Ганта

Таким образом, определяя перечень задач и её сроки, получим план разработки сайта (диаграмму Ганта), представленную в Приложении Д.

**3 Реализация**

**3.1 Руководство программиста**

**3.1.1 Организация данных**

В данном проекте мы используем базу данных PostgresSQL, в которой хранится информация об автомобилях и пользователях. Переменные в основном типа String – ФИО пользователя, пароль и т.д.

**3.1.2 Структура программы**

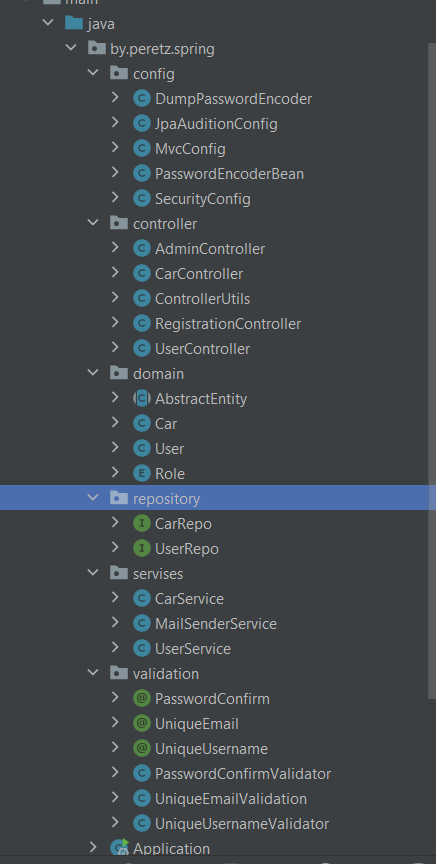
****

Рисунок 11 – Структура программы

Каждый класс в проекте выполняют свою функцию. Для того чтобы облегчить написание кода использовалось наследование.

**3.1.3 Структура и описание процедур и функций пользователя**

Пользователь может зарегистрироваться, выставить свой автомобиль на продажу и редактировать свой профиль.

**3.1.4 Спецификация программы**

Данный проект поможет пользователю продать/автомобиль. Также данный проект можно улучшать, добавляя больше функций.

**4 Тестирование**

Тестирование программного обеспечения было осуществлено в соответствии с тестами на использование, описанными в разделе «2.5 Тесты на использование». Отчёт о результатах тестирования предоставлен в Таблице 5.

Таблица 5 – Отчёт о результатах тестирования

|  |  |
| --- | --- |
| № | Статус |
| Т1 | Выполнено успешно |
| Т2 | Выполнено частично |
| Т3 | Выполнено успешно |
| Т4 | Выполнено успешно |
| Т5 | Выполнено успешно |

В ходе данных тестов выяснилось, что были ошибки, которые были устранены полностью или частично.

**5 Руководство пользователя**

**5.1 Общие сведения**

Наименованием программного продукта является сайт для продажи автомобилей. Назначение - для людей любого возраста, которые желают приобрести/продать автомобиль. Периодичность использования ПП - по мере необходимости.

**5.2 Запуск сайта**

Для того, чтобы запустить данный сайт на вашем компьютере, вам нужно лишь войти в любой из существующих браузеров, ввести ссылку сайта (http://localhost:8080/) в поисковик и вы на самом сайте(Рисунок 6/ Рисунок 7).

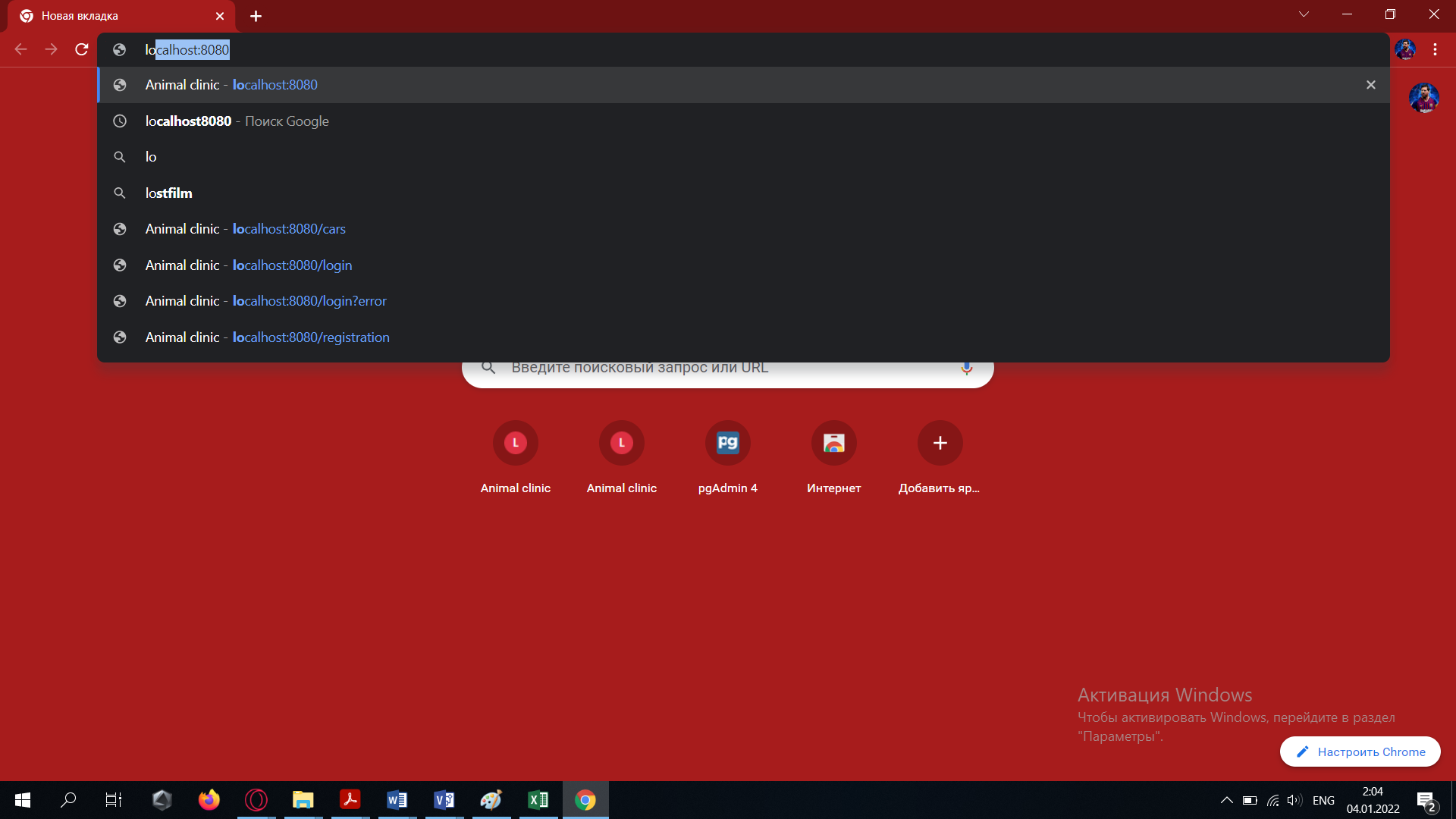


Рисунок 6 – Ввод ссылки

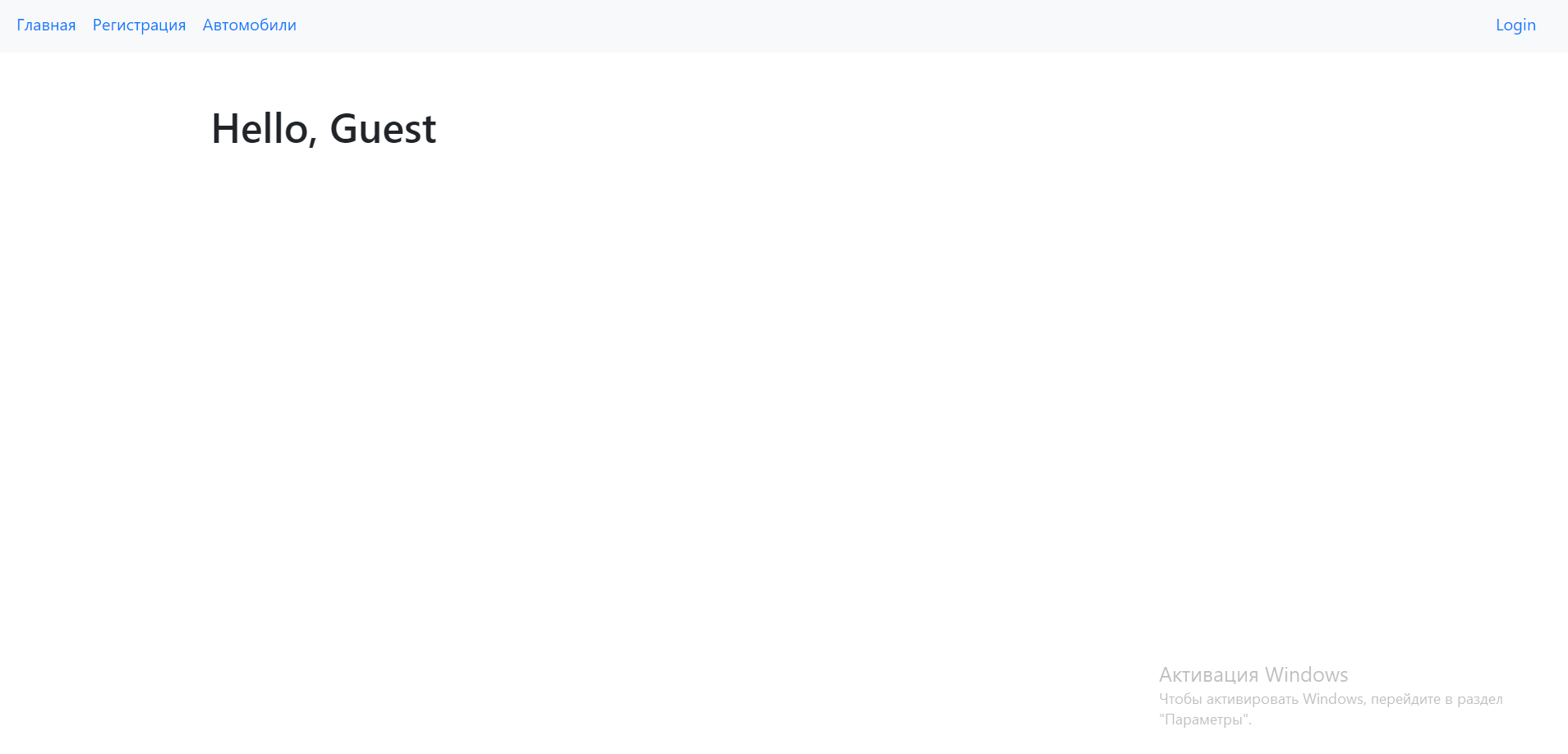


Рисунок 7 – Главная страница

**5.2 Инструкция по работе с сайтом**

Для начала нам необходимо зарегистрироваться(Рисунок 8)



Рисунок 8 – Окно регистрации

Далее после этого мы сможем работать на сайте. При нажатии кнопки “Add car” мы сможем добавить автомобиль на продажу. Также по желанию пользователь сможет изменить свои данные. Помимо этого, пользователь может удалить свой автомобиль из списка на продажу.

**5.4 Завершение работы с сайтом**

Закрываем полностью вкладку или браузер(Рисунок 9/Рисунок 10).



Рисунок 9 – Закрытие вкладки



Рисунок 9 – Закрытие браузера

**Заключение**

Целью данного учебного проекта являлась разработка сайта для продажи автомобилей.

В ходе реализации поставленной задачи были укреплены знания по использованию Java, а также курс дисциплины ТРПО.

Следует также учесть, что в поставленной задачи был реализован простой интерфейс, который позволяет использовать приложение пользователю, не обладающему дополнительными знаниями ЭВМ.

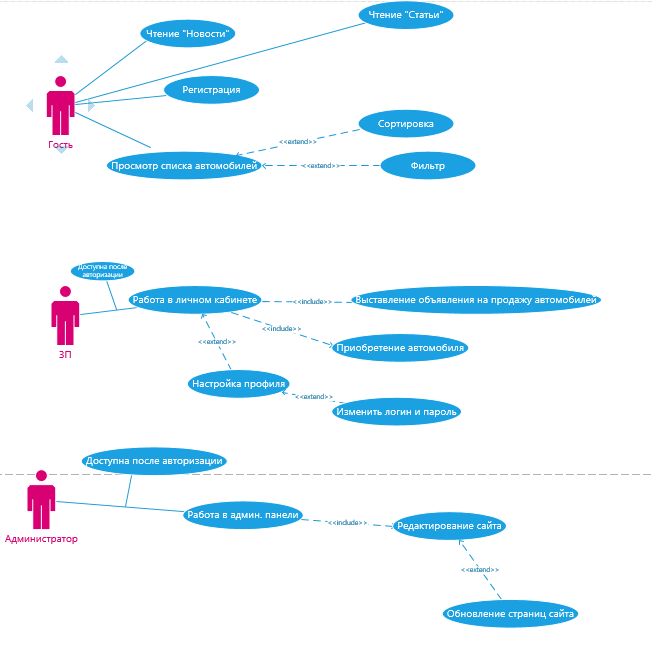
После тщательного тестирования приложения были выявлены некоторые недоработки, которые были частично исправлены на стадии проектирования или полностью исключены на стадии тестирования программы. В целом при реализации программы, были выполнены все условия, перечисленные в предыдущих разделах пояснительной записки. Таким образом, можно сказать, что программа была реализована вполне успешно.

**Список использованных источников**

1. Сайт о программировании. [Электронный ресурс]. – Режим доступа:  [https://metanit.com/](%20https://metanit.com/). – Дата доступа 20.12.2021.
2. Создание приложений с Spring Boot. [Электронный ресурс]. – Режим доступа: <http://spring-projects.ru/guides/spring-boot/>. – Дата доступа 22.12.2021.
3. EasyJava. [Электронный ресурс]. – Режим доступа: https://easyjava.ru/data/hibernate/. – Дата доступа 24.12.2021.

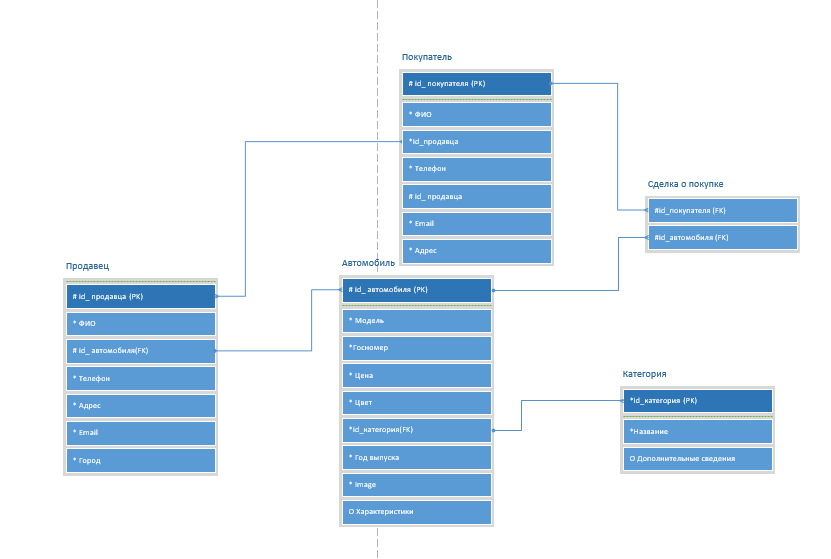
**Приложение А**

**Диаграмма вариантов использования**



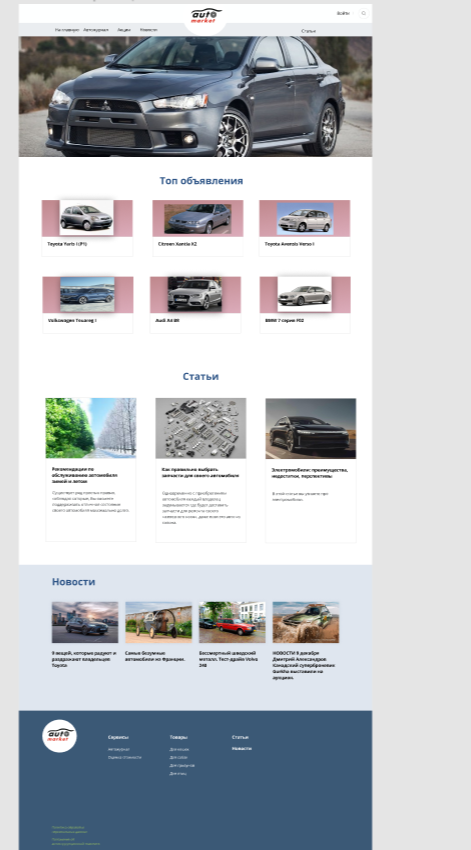
**Приложение Б**

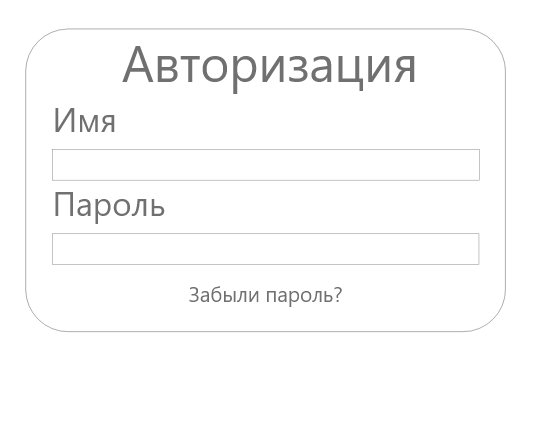
**Модель данных**

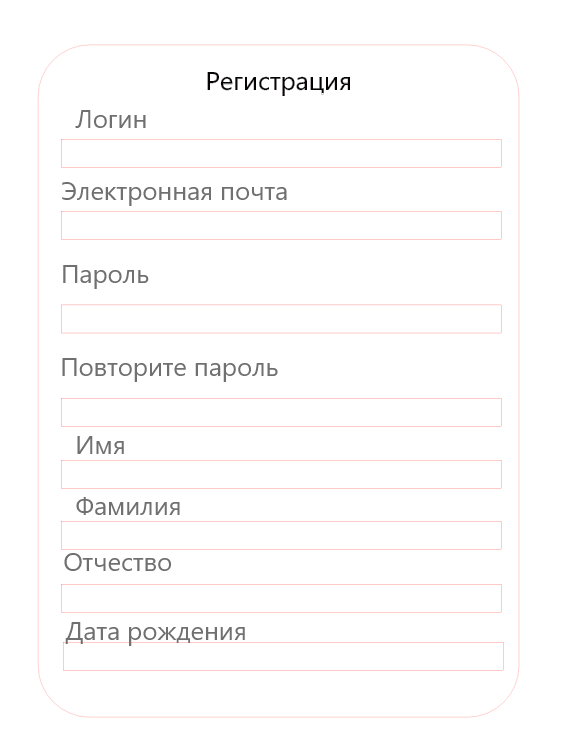


**Приложение В**

**Пользовательский интерфейс**

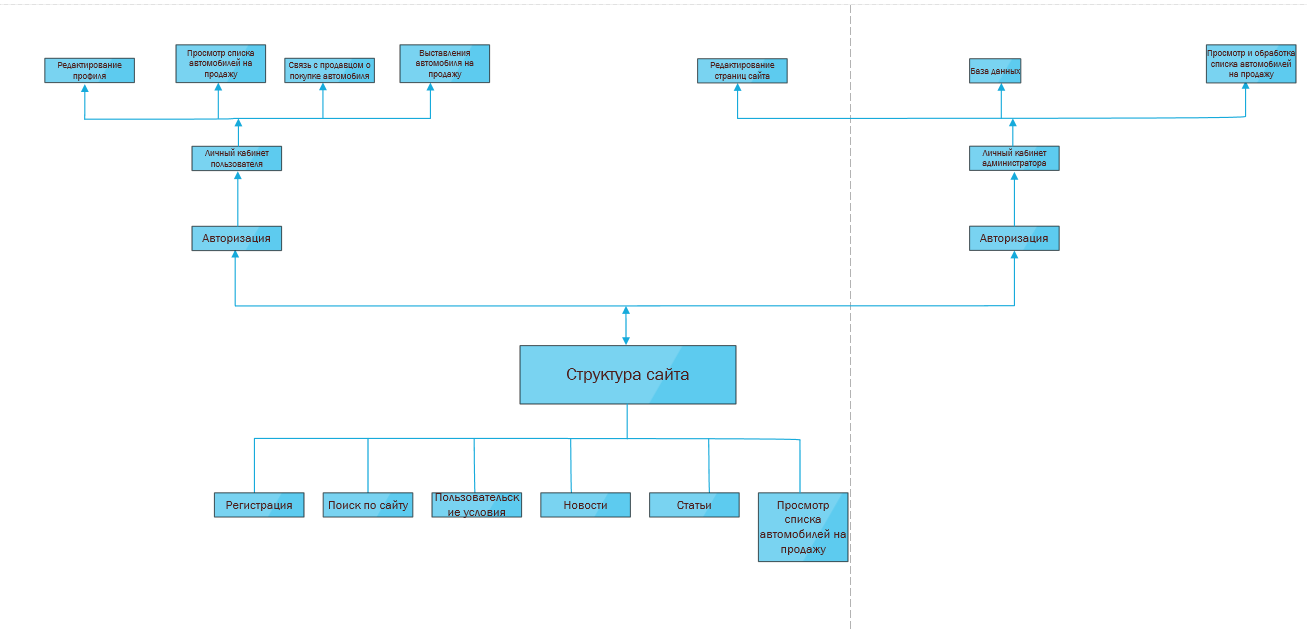






**Приложение Г**

**Структура сайта**



**Приложение Д**

**План работы над проектом**

