Федеральное государственное автономное образовательное учреждение высшего образования «Казанский (Приволжский) федеральный университет.»

*ПРОЕКТНАЯ РАБОТА*

По дисциплине / профессиональному модулю:

«Разработка приложений на C#»

Тема: «Список рекомендаций. Автомобили»

|  |  |
| --- | --- |
|  | **Студенты:** Ларичев В.В, Ганиев М.Р, Логинова Д.Р  **Руководители**: Фадеева Е.В, Латыпова Д.С |
|  |  |
|  |  |

Казань, 2024

Оглавление

[1. Отправка подборки на почту 3](#_Toc166121067)

[2. Авторизация через Яндекс 4](#_Toc166121068)

[3. Адаптивный дизайн 6](#_Toc166121069)

[4. Общая подборка для нескольких пользователей 7](#_Toc166121070)

## 1. Отправка подборки на почту

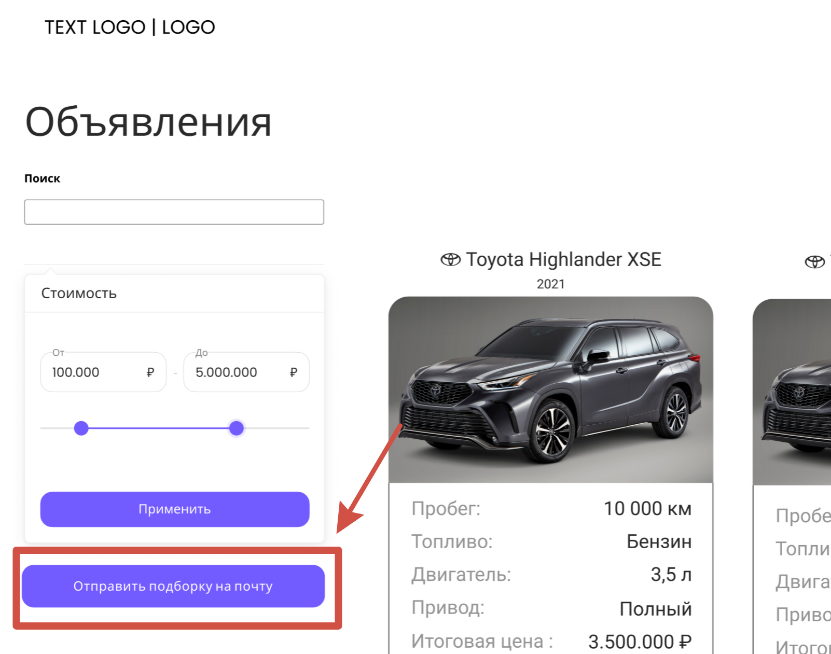
1. **Userstory**: Я, как пользователь приложения, хочу иметь возможность отправить текущую подборку автомобилей из страницы объявлений на свой электронный адрес, чтобы легко делиться объявлениями с другими.
2. **Пользователи**: Все зарегистрированные пользователи приложения.
3. **Место**: Страница с объявлениями.
4. **Задачи**:
   1. Добавить кнопку "Отправить на почту" на странице с объявлениями.
   2. Разработать функционал отправки подборки на почту, используя email, указанный в аккаунте пользователя.
5. **Техническое задание**:
   1. В окне "Объявления" добавить кнопку "Отправить подборку на почту" (рисунок 1)

Рисунок 1 - кнопка "Отправить на почту" Рисунок 2 - вид письма

* 1. Реализовать функционал отправки подборки на email пользователя.
  2. При нажатии на кнопку "Отправить подборку на почту", система должна автоматически генерировать письмо с подборкой.
  3. Заголовок письма может быть форматом "Моя подборка на сегодня:", а содержание — списком перечисляются названия автомобилей с годом их производства (рисунок 2).

## 2. Авторизация через Яндекс

1. **Userstory**: Я, как пользователь приложения, хочу иметь возможность авторизоваться в приложении через Яндекс, чтобы упростить процесс входа в систему.
2. **Пользователи**: Все пользователи приложения.
3. **Место**: Страница авторизации пользователя.
4. **Задачи**:
   1. Добавить возможность авторизации через Яндекс.
   2. В личном кабинете пользователя отображать информацию о типе аккаунта (внутренний, Яндекс).
5. **Техническое задание**:
   1. Добавить кнопку "Войти с Яндекс ID" в окне авторизации (рисунок 3).

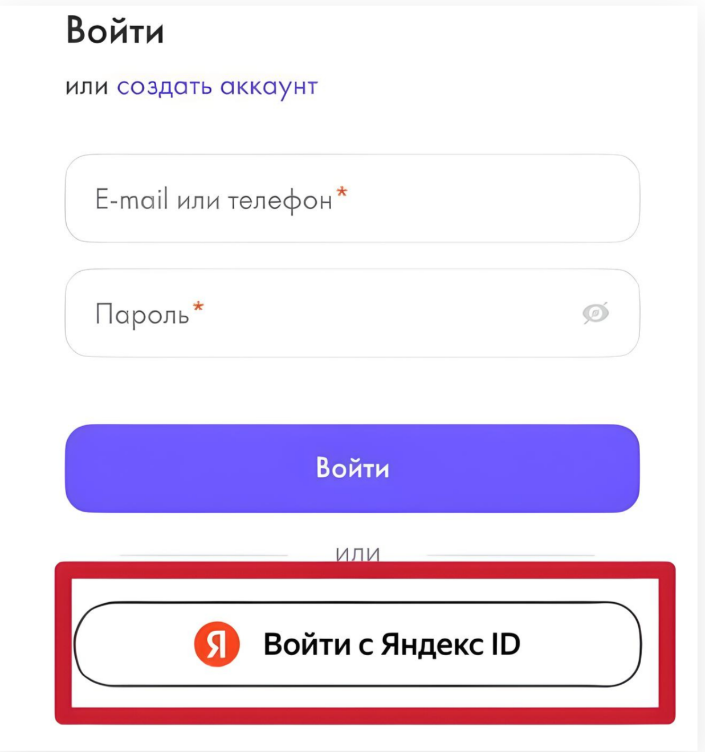


Рисунок 3- кнопка Войти с Яндекс ID

* 1. Зарегистрировать свое приложение в Яндексе и получить ключи API для доступа к сервисам Яндекса.
  2. Использовать Яндекс.Оauth для авторизации пользователей. Это включает в себя создание страницы авторизации, на которой пользователи смогут войти в свой аккаунт Яндекса и предоставить приложению разрешения на доступ к их данным.
  3. Информация о пользователе, которую взять из Яндекса: полное имя пользователя, его логин в Яндексе; электронная почта пользователя, связанная с его аккаунтом в Яндексе; ID пользователя в Яндексе.
  4. После успешной авторизации Яндекс вернет код авторизации, который затем можно обменять на токен доступа.
  5. После получения токена доступа, использовать его для выполнения запросов к API Яндекса, чтобы получить информацию о пользователе и сохранить её в базе данных.
  6. После успешной авторизации через Яндекс и сохранения информации о пользователе в системе, пользователь должен быть автоматически авторизован в приложении и должна быть метка о типе аккаунта в личном кабинете пользователя (рисунок 4).

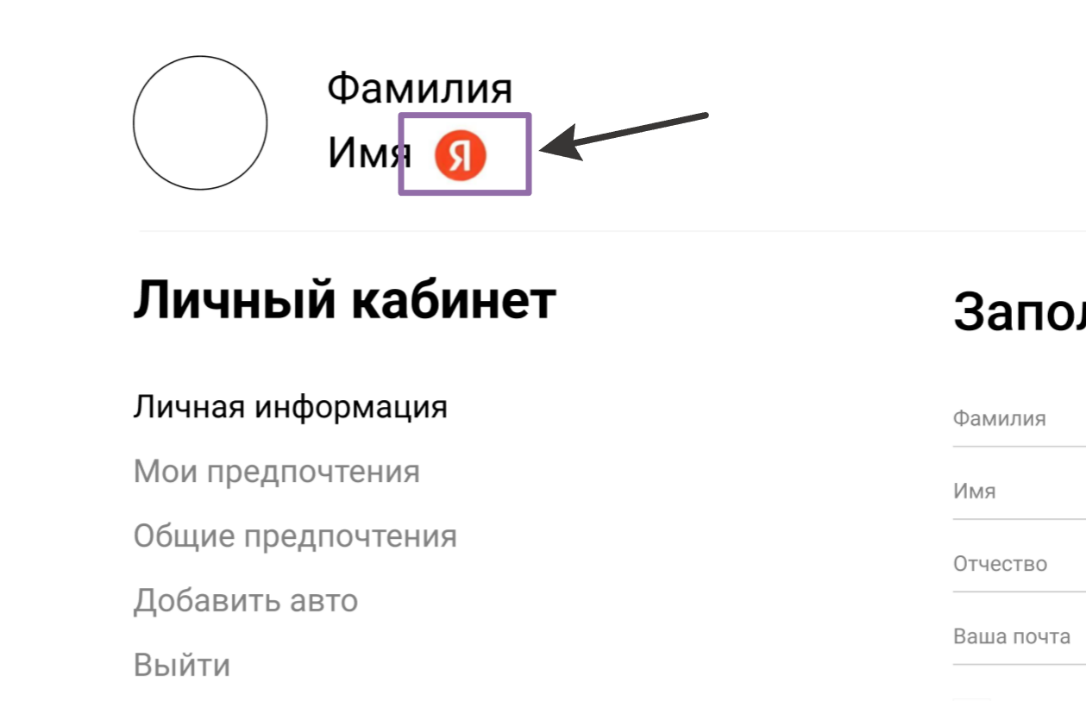


Рисунок 4 - метка о типе аккаунта

* 1. Реализовать перенаправление пользователя после успешной авторизации на главную страницу приложения – страницу с объявлениями.

## 3. Адаптивный дизайн

1. **Userstory**: Я, как разработчик приложения, хочу обеспечить адаптивность интерфейса приложения для ноутбуков и компьютеров, чтобы он корректно отображался на различных размерах экрана и обеспечивал удобство использования.
2. **Пользователи**: Все пользователи приложения, использующие ноутбуки или компьютеры.
3. **Место**: Весь интерфейс приложения.
4. **Задачи**:
   1. Разработать несколько версий форм для разных размеров экранов (например, 1366 x 768, 1920 x 1080).
   2. Подобрать подходящие шрифты, размеры изображений и интервалы для каждой версии формы.
   3. Применить свойства AutoScaleMode и Anchor для автоматической масштабируемости элементов на форме.
   4. Протестировать адаптивный дизайн на разных размерах экранов и проверить корректное отображение и удобство использования интерфейса.
5. **Техническое задание**:
   1. Создать отдельно для каждой версии формы соответствующие шрифты, размеры интерфейсных элементов.
   2. Настроить свойства AutoScaleMode и Anchor на форме для автоматической адаптации элементов к различным разрешениям экранов.
   3. Дополнительно протестировать на разных устройствах (например, монитор с разрешением 1366 x 768 и монитор с разрешением 1920 x 1080) для проверки корректного отображения и удобства использования на каждом из них.

## 4. Общая подборка для нескольких пользователей

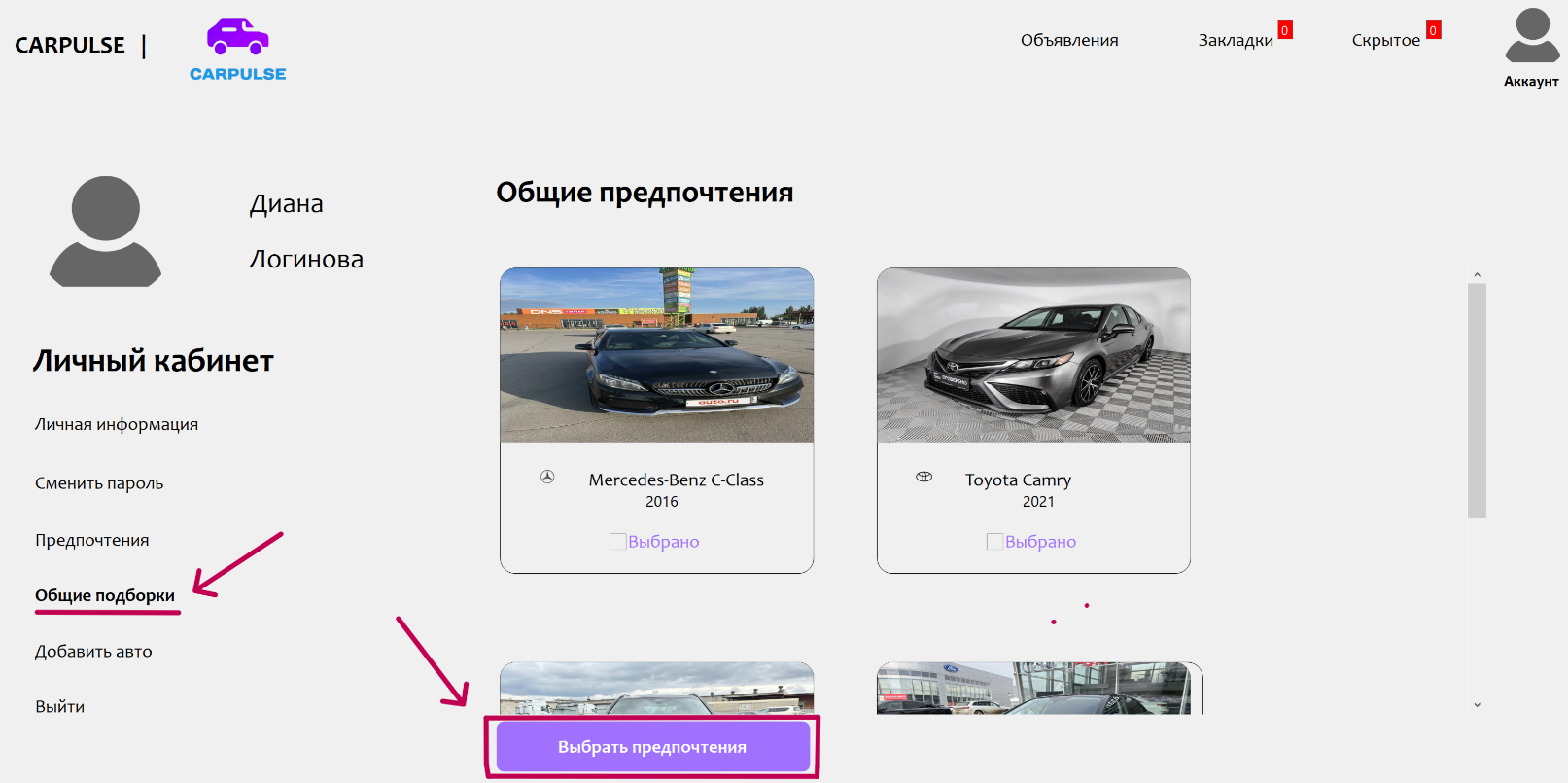
1. **Userstory**: Я, как пользователь приложения, хочу иметь возможность видеть общую подборку автомобилей, которую могут использовать несколько пользователей, что позволит мне быстрее находить автомобили, которые могут меня заинтересовать.
2. **Пользователи**: Все зарегистрированные пользователи приложения, которые хотят увидеть общую подборку.
3. **Место**: Страница с общими подборками.
4. **Задачи**:
   1. . Разработать отдельное окно в приложении – "Общие предпочтения", где будут отображаться общие подборки автомобилей. Это окно должно быть доступно для всех пользователей приложения.
   2. . Реализовать механизм, который будет отслеживать совпадения между подборками разных пользователей. Например, если у 3 пользователей есть один и тот же автомобиль в их личных предпочтениях, этот автомобиль должен отображаться в общих подборках первым.
   3. . В окне "Общие подборки" отображать 10 позиций, совпавшие с личными предпочтениями разных пользователей.
5. **Техническое задание**:
   1. Разработать структуру базы данных:
      1. Таблица Пользователей: хранит информацию о пользователях, включая ID пользователя, имя, фамилию, телефон, email, пароль.
      2. Таблица Подборки: хранит информацию о каждой подборке пользователя, включая ID подборки, ID пользователя, список автомобилей в подборке.
      3. Таблица Автомобилей: хранит информацию о всех автомобилях, доступных для выборки, включая ID автомобиля, марка, модель, тип кузова, тип двигателя, топливо, привод, КПП, Руль, Цвет.
      4. Таблица Совпадений: хранит информацию о совпадениях между подборками разных пользователей, включая ID совпадения, ID первого пользователя, ID второго пользователя, ID третьего пользователя и список совпадающих автомобилей.
   2. Сравнивать список автомобилей в подборках и определять совпадения по критериям, таким как марка, модель, тип кузова тип двигателя, топливо, привод, КПП, Руль, Цвет.
   3. Разработать алгоритм ранжирования, который будет учитывать количество совпадений предпочтений нескольких пользователей при отображении автомобилей в списке общих подборок. Чем больше совпадений предпочтений, тем выше автомобиль должен быть ранжирован в списке.
   4. Разработать пользовательский интерфейс для окна "Общие подборки", который будет отображать список совпадающих автомобилей. Интерфейс схож с интерфейсом уже существующего окна "Предпочтения" с добавлением изменений:
      1. Слева сверху должна быть добавлена кнопка для перехода на это окно (рисунок 5).
      2.  Можно выбрать галочкой понравившиеся из общих предпочтений автомобили, чтобы добавить их в свои личные предпочтения (рисунок 5).

Рисунок 5 - окно "Общие предпочтения"