3 Сессия

**Общее**

Необходимо разработать мобильное приложение для смартфона, удовлетворяющее следующим требованиям:

Минимальная версия ОС, поддерживаемая приложением, должна быть: Android - 11.0, iOS - 14.0.

В качестве бэкенда будет использован Supabase. Для авторизации в supabase используйте учетную запись, **выданную главным экспертом.**

**В работе необходимо использовать систему контроля версий Git, который предоставляет организатор.**

**Необходимо строго следовать предложенному дизайну.**

Необходимо осуществлять комментирование кода в созданных классах. Обязательны следующие комментарии:

* Описание назначения класса
* Дата создания
* Автор создания
* Описание назначения вложенных элементов программного кода

При разработке проекта приложения вам необходимо использовать архитектуру (см.файл с описанием архитектуры), в которой будут разделены слои бизнес-логики, представлений и домена. Изменение бизнес-логики и/или представления одного из экранов не должно повлечь за собой изменение других экранов и нарушение работоспособности приложения, за исключением переходов. Допускается использование SupaBase.

Файлы проекта распределены по папкам в соответствии с архитектурой. Допустимо использование папки Common для общих файлов.

**Вся верстка должна быть адаптивной (следует учитывать разные размеры экранов). Необходимо:**

* **Избегать появления большого пустого пространства;**
* **Следить за отсутствием искажения элементов;**
* **Все элементы должны полностью находится в границах и на месте, указанном в макете;**
* **Учитывать расстояние между элементами;**
* **Используйте шрифты согласно макету.**
* **Дизайн предложен в Figma:**

**https://www.figma.com/file/guyXW190UglGSwE2CCyafu/OECH-APP-Final?type=design&node-id=0-1&mode=design&t=lrg58KLyrOO3iQa5-0**

Необходимо корректно обрабатывать запросы к серверу. В случае получения ошибки от сервера или отсутствия соединения с сетью Интернет необходимо отобразить соответствующий текст ошибки в диалоговом окне, которое должно закрываться только пользователем.

В процессе обмена данными с сервером должна осуществляться стандартная индикация.

Необходимо во время сессии работать в ветке “Path-X”, где Х – это номер сессии. По завершению сессии необходимо сделать средствами Giltab Merge, с основной веткой, которая должна называться “main”, при этом ветка удалятся не должна.

1. Создайте нижнее меню экрана «Home», как на макете.
2. Создайте экран «Profile», как на макете:

* Реализуйте возможность переключения в Dark Mode и обратно
* Реализуйте возможность скрытия и появление текущего баланса (данные о балансе заменяются «звездочками»)
* При нажатии на кнопку «Card & Bank account settings» осуществляется переход на экран «Add Payment method»
* При нажатии на кнопку «Log Out» осуществляется выход пользователя из системы и переход на экран «Log In»
* Сведения о балансе берутся с сервера

1. Создайте экран «Notification», как на макете:

* Реализуйте возможность возврата на предыдущий экран
* Реализуйте доступ на экран «Notification» из соответствующего пункта экрана «Home»

1. Создайте экран «Send a package (Empty)», как на макете:

* Реализуйте возможность возврата на предыдущий экран
* Реализуйте доступ на экран «Send a package (Empty)» из соответствующего пункта экрана «Home»;
* Реализуйте возможность добавления нескольких пунктов доставки, экран соответствует макету «Send a package (Full (2 destination))»
* При нажатии на кнопку «Instant delivery», формируется трек-номер при помощи UUID который должен соответствовать паттерну: R-\*сгенерированный код\*;
* При нажатии на кнопку «Instant delivery»:
  + при условии выбора нескольких пунктов назначения и при заполнении всех обязательных полей (поле «Others» является необязательным) осуществляется переход на экран «Send a package (Full (2 destination))»
  + при выборе одного пункта назначения и при заполнении всех обязательных полей (поле «Others» является необязательным) осуществляется переход на экран «Send a package (Full (1 destination))»
  + при условии заполнения всех обязательных полей, осуществляется отправка данных на сервер
* Реализуйте функционал оформления доставки: адрес и страна откуда доставить формируется на основе геокодирования данных местоположения устройства

1. Создайте экран «Send a package (Receipt(1 destination))» как на макете:

* При нажатии на кнопку «Edit package» осуществляется возврат на предыдущий экран для редактирования введенных данных;
* Все данные на экране берутся с предыдущего;
* Поле «Delivery Charges» по формуле 2500 \* кол-во пунктов назначения;
* Поле «Instant Delivery» - фиксированное произвольное значение;
* Поле «Tax» – 5% от (Delivery Charges + Instant Delivery);
* Необходимо реализовать подсчёт итоговой суммы;
* Данные о доставке приходят с сервера
* При нажатии на кнопку «Make payment» осуществляется переход на экран «Transaction successful-1»

1. Создайте экран «Transaction successful-1», как на макетах:

* Реализуйте анимацию вращения в процессе осуществления оплаты, как на макете «Transaction successful-1»; после этого экран примет вид, как на макете «Transaction successful-2»
* Данные о номере трека берутся с сервера

1. Создайте экран «Transaction successful-2», как на макетах:

* Данные о трек номере необходимо отобразить с предыдущих экранов;
* При нажатии на кнопку «Track my item» осуществляется переход на экран «Tracking Package»
* При нажатии на кнопку «Go back to homepage» осуществляется переход на экран «Home»