4 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

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 экран «Wallet» как на макете:

* Реализуйте возможность скрытия текущего баланса (данные о балансе заменяются «звездочками»).
* Реализуйте отображение истории транзакций с помощью прокручиваемого списка, отсортированного по дате (от последней к первой транзакции).
* Сведения о балансе берутся с сервера.
* Сведения о транзакциях берутся с сервера.

1. Создайте экран «Tracking Package», как на макете:

* Доступ к экрану возможен только при наличии активного заказа, в остальных случаях пункт меню «Track» неактивен. Как это лучше реализовать? И как это доступ возможен? Переключиться нельзя, или можно, но он будет пустой с надписью нет активных заказов???
* При нескольких активных заказах, на экране отображается последний сформированный.
* Реализуйте возможность масштабирования, перемещения и поворота карты посредством жестов.
* На экране отображается в реальном времени текущее состояние доставки.
* Данные для построения трека на карте берутся с сервера (точки). так-то легко, но с картой как работать?
* Номер трека берется с сервера.
* Данные о состояниях доставки берутся с сервера.
* Состояние доставки обновляется при его изменении (после обновления статуса на сервере, в приложении также должно отобразиться изменение статуса). Просто подгружать каждую секунду данные с сервера?
* При нажатии на кнопку «View Package Info» осуществляется переход на экран «Send a Package».
* Реализуйте кэширование карты

1. Доработайте экран «Send a Package-2»:

* По окончании доставки экран принимает вид согласно макету.
* Реализуйте возможность возврата на предыдущий экран.
* При нажатии кнопку «Successful» осуществляется переход на экран «Delivery successful-1».
* Сведения о доставке берутся с сервера.
* Сведения о сборах берутся с сервера.

1. Создайте экран «Delivery Successful», как на макете:

* Реализуйте анимацию вращения (2 полных оборота), как на макете «Delivery Successful-1»; после этого экран должен принимать вид, как на макете «Delivery Successful-2».
* Реализуйте возможность добавления отзыва (не более 10 символов).
* Реализуйте возможность выставления оценки посредством наклона устройства: при наклоне на 50 градусов вправо происходит увеличение количества звезд на 1, в противоположную сторону – уменьшение количества на 1.
* За один наклон можно добавить или убрать не более одной звезды.
* При нулевом количестве звезд наклон влево не функционален.
* Аналогично при полном количестве звезд и наклоне вправо.
* Сведения о выставленной оценки отправляются на сервер.
* Содержимое отзыва отправляется на сервер.
* При нажатии на кнопку «Done» осуществляется переход на экран «Home».

1. Создайте экран «Add Payment method», как на макете:

* Реализуйте возможность возврата к предыдущему экрану;
* При выборе способом оплаты «кредитке/дебетовой картой» раскрывается список с доступными картами. А стоит ли оно того?