**Модуль Ж - Подготовка продукта (вариатив)**

(все сессии) 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”.

**Сессия – 1**

Реализуйте приветственные экраны (Onboard) согласно методологии TDD и макету:

Логика экрана:

* + У Вас есть определенная последовательность изображений и текста. Это значит, что у Вас должна быть очередь (queue) этих изображений и текстов. При нажатии на кнопку «Next» необходимо извлекать по одному изображению и тексту и показывать на экране до тех пор, пока изображения и текст в очереди не закончатся. Изображения и текст при переключении должны сопровождаться анимацией проявления (исчезает текущий и появляется следующий);
  + Когда в очереди останется последнее изображение и текст — нужно поменять кнопки;
  + Когда пользователь увидит последнее изображение и нажмет кнопку "Sing Up", нужно показать пустой экран «Holder».
  + При нажатии на кнопку «Skip», осуществить переход на экран «Holder» приложения и сохранить информацию о том, что пользователь уже смотрел Onboard (не показывать ее при последующих запусках), а также очистить очередь.

1. 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.
2. Экран «Onboard»:
   * Данные очереди не должны храниться в компоненте (в компоненте хранятся только данные текущего элемента);
   * Очередь создаётся единственный раз при первом запуске и не пересоздаётся;
   * Сохраняется прогресс отображения Onboard (При перезапуске должен отобразиться следующий набор изображения и текста, на котором остановился пользователь);

Создайте необходимые классы для последующего тестирования. Добавьте объявления методов, реализовывать их нужно.

RED - Реализуйте тесты (UI или модульные), проверяющие поведение приложения:

* + - Изображение и текста из очереди извлекается правильно (в порядке добавления в очередь).
    - Корректное извлечение элементов из очереди (количество элементов в очереди уменьшается на единицу).
    - В случае, когда в очереди несколько картинок, устанавливается правильная надпись на кнопке.
    - Случай, когда очередь пустая, надпись на кнопке должна измениться на "Sing Up".
    - Если очередь пустая и пользователь нажал на кнопку “Sing in”, происходит открытие пустого экрана «Holder» приложения. Если очередь не пустая – переход отсутствует.

Минимальная структура тестов следующая:

* объект класса, который мы тестируем;
* корректные/некорректные входные данные;
* сравнение полученного результата с ожидаемым (Assert).

Сделайте коммит с текущим состоянием тестов (Failed) и оставьте сообщение RED.

GREEN - напишите реализацию логики так, чтобы все тесты проходили успешно (в случае изменения теста на данном этапе, тест не будет учитываться в оценке).

Сделайте коммит с текущим состоянием тестов (Succes) и оставьте сообщение GREEN.

REFACTOR: проведите рефакторинг не менее трех различных участков кода, созданного на предыдущем этапе (в случае изменения теста на данном этапе, тест не будет учитываться в оценке). Рефа́кторинг (англ. refactoring), или перепроектирование кода, переработка кода, равносильное преобразование алгоритмов — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы.

Сделайте коммит с текущим состоянием тестов (Succes) и оставьте сообщение REFACTOR.

**Сессия -2**

1. Создайте экран «Sign Up», как на макете:

* При нажатии на кнопку «Sign Up» осуществляется переход на экран «Log In»
* Регистрация и переход на экран «Log In» осуществляется только при согласии с Условиями и политикой конфиденциальности
* Реализуйте возможность перехода на экран «Log In» при нажатии на «Sign in»

2. Создайте экран «Log In», как на макете:

* При нажатии на «Forgot Password» осуществляется переход на экран «Forgot Password»
* При успешной авторизации осуществляется переход на экран «Home»
* Реализуйте возможность перехода на экран «Sign Up» при нажатии на «Sign Up»

3. Создайте экран «Forgot Password», как на макете:

* Реализуйте возможность перехода на экран «Log In» при нажатии на «Sign in»

4. Создайте экран «OTP Verification», как на макете:

* При корректном коде при нажатии на кнопку «Set New Password» осуществляется переход на экран «New Password»

5. Создайте экран «New Password», как на макете:

При нажатии на «Log In» осуществляется изменение пароля и переход на экран «Home»;

**Модуль В - Клиент-серверное взаимодействие приложения (инвариант)**

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.

1. Экран «Sign Up», как на макете:

* Реализуйте проверку email на корректность (соответствие паттерну «name@domenname.ru», где имя и доменное имя может состоять только из маленьких букв и цифр). При некорректном заполнении необходимо отобразить ошибку любым способом;
* Реализуйте возможность отображения пароля;
* Реализуйте повторный ввод пароля для подтверждения;
* Реализовать просмотр политики конфиденциальности, которая хранится локально в проекте в виде PDF файле и открывается в свободной форме;
* Регистрация и переход на экран «Log In» осуществляется только при согласии с Условиями и политикой конфиденциальности
* Реализуйте отправку запроса на сервер для регистрации
* Реализуйте возможность регистрации посредством использования активного аккаунта Google

2. Экран «Log In», как на макете:

* Реализуйте возможность отображения пароля
* Реализуйте отправку запроса на сервер для авторизации с помощью почты и пароля
* Реализуйте возможность авторизации посредством использования активного аккаунта Google
* Реализуйте возможность сохранение пароля

3. Экран «Forgot Password», как на макете:

* При нажатии на кнопку «Send OTP», при наличии в поле ввода корректного e-mail, осуществляется переход на экран «OTP Verification»
* Реализуйте отправку запроса на сервер для получения кода

4. Экран «OTP Verification», как на макете:

* Реализуйте возможность повторного запроса кода по истечению таймера 01:00 минута;
* Пока код не введён, кнопка «Set New Password» не активна;
* После ввода символа, соответствующий квадрат окрашивается в синий цвет;
* Если код-пароль введён не верно, то все квадраты становятся красными;
* Реализуйте отправку кода на сервер для верификации
* Реализуйте повторную отправку кода на сервер для верификации

5. Экран «New Password», как на макете:

* Реализуйте проверку совпадения паролей;
* Реализуйте отправку запроса на сервер для изменения пароля

**Модуль Г - Хранение информации (инвариант)**

1. Экран «Log In», как на макете:

* Обеспечить безопасное хранение пароля используя SHA-512

**Сессия – 3**

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»

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))»

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;
* 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»

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

* 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»

**Модуль В - Клиент-серверное взаимодействие приложения (инвариант)**

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.

1. Экран «Profile», как на макете:

* При нажатии на кнопку «Log Out» осуществляется выход пользователя из системы и переход на экран «Log In»
* Сведения о балансе берутся с сервера

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

* 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. Экран «Send a package (Receipt(1 destination))» как на макете:

* 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
* Данные о номере трека берутся с сервера

**Сессия – 4**

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).

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

* Доступ к экрану возможен только при наличии активного заказа, в остальных случаях пункт меню «Track» неактивен.
* При нескольких активных заказах, на экране отображается последний сформированный.
* Реализуйте возможность масштабирования, перемещения и поворота карты посредством жестов.
* При нажатии на кнопку «View Package Info» осуществляется переход на экран «Send a Package».

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».

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).
* 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.

**Модуль В - Клиент-серверное взаимодействие приложения (инвариант)**

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.

1. Экран «Wallet» как на макете:

* Сведения о балансе берутся с сервера.
* Сведения о транзакциях берутся с сервера.

1. Экран «Tracking Package», как на макете:

* На экране отображается в реальном времени текущее состояние доставки.
* Данные для построения трека на карте берутся с сервера (точки).
* 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).

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

* Сведения о доставке берутся с сервера.
* Сведения о сборах берутся с сервера.

1. Экран «Delivery Successful», как на макете:

* Сведения о выставленной оценки отправляются на сервер.
* Содержимое отзыва отправляется на сервер.
* При нажатии на кнопку «Done» осуществляется переход на экран «Home».

**Модуль Г - Хранение информации (инвариант)**

Все медиа ресурсы должны кэшироваться.

1. Экран «Tracking Package», как на макете:

* Реализуйте кэширование карты

**Модуль Д - Взаимодействие с аппаратными расширениями устройства (инвариант)**

1. Экран «Delivery Successful», как на макете:

* 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.

**Сессия - 5**

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

* Реализуйте возможность возврата к предыдущему экрану.
* При авторизации в приложении как курьера экран «Chats» должен содержать список пользователей.
* При авторизации под ролью клиента экран «Chats» должен содержать список курьеров.
* Реализуйте поиск по имени клиента / курьера.
* Реализуйте счётчик непрочитанных сообщений.
* Диалоги необходимо сортировать по дате последнего сообщения.
* При выборе диалога, осуществляется переход на экран «Chat Rider» соответствующего диалога.

1. Создайте экран «Chat Rider», как на макете:

* Реализуйте возможность возврата на предыдущий экран.
* Имя человека, с которым ведется диалог необходимо передать с предыдущего окна.
* При нажатии на соответствующую кнопку осуществляется переход на экран «Call Rider».

1. При нажатии на кнопку завершения вызова осуществляется переход на предыдущий экран
2. Создайте экран «Home», как на макете:

* Реализовать горизонтальный скролл рекламы для просмотра;
* При удержании блока меню, появляется выделение соответствующее макету;

**Модуль В - Клиент-серверное взаимодействие приложения (инвариант)**

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.

1. Экран «Chats», как на макете:

* Фото берется с сервера.
* Фамилия и имя берется с сервера.
* Реализуйте получение новых сообщений realtime.

1. Экран «Chat Rider», как на макете:

* Реализуйте загрузку с сервера истории сообщений.
* Текущие сообщения должны отправляться на сервер.

1. Экран «Call Rider», как на макете:

* Номер телефона доставщика берется с сервера.

1. Экран «Home», как на макете:

* Сведения о пользователе берутся с сервера
* Изображение для рекламы берется с сервера
* Реализовать отправку аватарки пользователя на сервер
* Реализовать получение аватарки пользователя с сервера

**Модуль Г - Хранение информации (инвариант)**

* Создайте экран «Home», как на макете:
* Реализуйте кэширование медиа-ресурсов рекламного блока

**Модуль Д - Взаимодействие с аппаратными расширениями устройства (инвариант)**

1. Экран «Home», как на макете:

* Реализуйте изменение аватарки профиля (по средством возможности выбора изображение из галереи или при помощи камеры устройства), при нажатии на аватарку;
* Реализуйте функционал для работы с галереей и камерой устройства для получения аватарки пользователя.

**Сессия – 6**

* + - 1. Создайте презентацию:
  + Презентация должна быть рассчитана на разработчиков (Вы не должны продавать приложение!).
  + В презентации нужно рассказать о реализации приложения.
  + Показать схему классов.
  + Производительность.
  + Используемые архитектурные решения.
  + Используемые библиотеки или описание почему не использовались.

1. Подготовьте информацию для подготовки публикации приложения в магазине. Необходимо сохранить все разработанные материалы в папку Publication и загрузить вместе с проектом в систему контроля версий (Текстовые документы, изображения, APK (Архивный файл, файл сборки):
   * Создайте приложение с названием, «Фамилия + Delivery» (замените слово Фамилия на свою фамилию).
   * Приложение должно быть бесплатным.
   * Подготовьте иконки приложения для магазина (32х32, 64x64, 128x128).
   * Выберите Категорию.
   * Добавьте краткое описание.
   * Добавьте основное описание.
   * Добавьте Скриншоты приложения (необходимо загрузить все размеры необходимые для публикации на смартфон не менее 5 штук).