# *Задание на практику*

# *ПП.01.01 Разработка мобильных приложений*

# *Цель практики:*

# *Разработать мобильное приложение заданной тематики*

# *Структура задания:*

1. Модуль 1 Разработка технического задания:

* описание функциональных требований;
* проектирование структуры данных;
* создание макета приложения.

1. Модуль 2 Разработка серверной части;
2. Модуль 3 Верстка мобильного приложения;
3. Модуль 4 Разработка функционала мобильного приложения;
4. Модуль 5 Тестирование приложения;
5. Модуль 6 Документирование результатов разработки.

# *Общие требования к приложению:*

1. в работе необходимо использовать систему контроля версий Git, и ежедневно сохранять разработанные элементы проекта;
2. проект приложения должен быть структурирован, то есть исходные файлы должны быть в соответствующей папке, общие классы необходимо поместить в папку common.

Например, при использовании в проекте MVC, файлы проекта должны быть распределены по следующим папкам: view, model, controller, common.

1. в приложении должны корректно обрабатываться 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.

# *Этапы выполнения*

Перечисленные далее модули являются обязательными, в независимости от темы индивидуального задания. Внимательно изучите задачи и этапы модулей прежде чем приступать к заданию. Необходимо добавить дополнительные функции, соответствующие теме индивидуального задания. Допускается изменение обязательных функций, с разрешения преподавателя.

## Модуль 1 – Разработка технического задания

Модуль 1 является начальным и обязательным этапом разработки мобильного приложения и включает в себя следующие шаги:

1. определение целей и задач приложения;

* Обеспечить пользователя удобным и быстрым способом поиска и бронирования отелей.
* Предоставить актуальную информацию об отелях: цены, описание, фотографии, отзывы.
* Создать личный кабинет для управления бронированиями и сохранения понравившихся отелей.
* Обеспечить высокий уровень удобства и простоты использования.

1. определение функциональных требований. Все перечисленные в модулях функции должны быть реализованы в обязательном порядке.

* Регистрация и аутентификация пользователей.
* Восстановление пароля через OTP.
* Управление профилем пользователя (редактирование данных, загрузка аватара).
* Управление данными об отелях.
* Отображение отображение понравившихся и забронированных отелей.
* Общение с управляющими отелем.
* Фильтрация и поиск отелей по различным параметрам.

1. проектирование и описание структуры данных, с которыми будет работать приложение;

Users (id, name, email, password\_hash, avatar\_url, balance, created\_at)

Chats (id, user\_id, manager\_id, hotel\_id, last\_message\_at, is\_active)

Messages (id, chat\_id, sender\_id, text, media\_url, sent\_at, is\_read)

Favorites (id, user\_id, hotel\_id, added\_at)

Hotels (id, name, address, description, stars, price\_per\_night, rating, created\_at)

Bookings (id, user\_id, hotel\_id, check\_in\_date, check\_out\_date, total\_price, status, created\_at)

1. разработка дизайн-макета приложения в Figma.

Требования при разработке прототипа приложения:

* макет должен включать все ключевые части приложения. Если какие-то части макета не соответствуют требованиям, их необходимо переработать или добавить;
* макет должен быть реализован аккуратно, с вниманием к деталям. Приложение, реализованное по этому макету, должно быть удобно в использовании пользователем.

## Модуль 2 - Разработка серверной части

Для создания базы данных в соответствии с разработанной структурой следует использовать сервис *Supabase*.

Требования при разработке серверной части:

1. реализованная база данных должна покрывать все задачи приложения;
2. все данные, которые используется в системе, должны храниться в базе данных, за исключением локальных и данных для тестирования;
3. для управления доступом к данным должна быть реализована политика безопасности;
4. в базе данных должно быть реализовано хранение изменяемых файлов и изображений;
5. аутентификация пользователя должна быть реализована по почте и паролю;
6. восстановление пароля пользователя по почте должно быть реализовано с помощью шестизначного OTP-кода;
7. база данных должна быть заполнена достаточным количеством данных для проверки работоспособности всех функций приложения;

## Модуль 3 Верстка мобильного приложения

Требования при реализации дизайна приложения:

1. дизайн приложения должен строго следовать разработанному макету;
2. вся верстка должна быть адаптивной (следует учитывать разные размеры экранов);
3. приложение должно иметь настроенную иконку в соответствии с макетом;
4. приложение должно иметь следующие экраны:

* загрузочный экран;
* экран Приветствия (Onboarding), который отображается, если пользователь не авторизован;
* экран Входа;
* экран Авторизации;
* экран Ввода ПИН-кода;
* экран Входа по ПИН-коду;
* экран Регистрации;
* экран Восстановление пароля;
* экран Ввода OTP-кода;
* экран Изменения пароля;
* экран Личного профиля;
* экран Получение списка отелей;
* экран Получения отелей с фильтрацией;
* экран Получения подробностей об отеле;
* экраны Экран Создания / Редактирования отеля(допускается объединение);
* экран Избранное
* экран Забронированные отели.
* экран Чаты сообщения
* экран Поиска и фильтрации
* диалоговые окна для ошибок.

## Модуль 4 Разработка функционала клиентского приложения

В мобильном приложении должны быть реализованы все функции, заявленные в техническом задании.

Минимальные требования к функционалу приложения:

1. аутентификация: неавторизованные пользователи должны быть ограничены в доступе к данным;
2. регистрация пользователя: для создания нового пользователя необходимо ввести: имя пользователя, электронную почту и пароль. Данные пользователя отправляются на сервер для регистрации;
3. авторизация пользователя: для авторизации пользователя необходимо ввести: электронную почту и пароль. Данные пользователя отправляются на сервер для авторизации.
4. пользовательские сессии:

* после успешной аутентификации данные пользователя сохраняется в системе для дальнейшего использования;
* электронная почта и пароль сохраняются в локальное хранилище, они используются для повторного входа и удаляются при выходе из системы;
* данные токена пользователя сохранятся в виде локальной переменной и очищаются при закрытии приложения.

1. валидация вводимых данных:

* проверять поля на соответствие типу данных;
* проверять поля для ввода на пустоту, на длину (короткий текст и числа: не более 20 символов, пароль: не более 8 символов);
* email на корректность (соответствие паттерну "name@domenname.ru", где имя и доменное имя может состоять только из маленьких букв и цифр).
* проверять повторный ввод пароля на соответствие исходному;
* проверять даты на корректность;

1. навигация в приложении:

* реализовать навигацию между всеми необходимыми экранами приложения
* для реализации навигации рекомендуется использовать фрагменты или библиотеки навигации;
* для перехода между основными экранами приложения (например, главный экран, профиль, корзина) рекомендуется использовать нижнюю навигационную панель

1. ПИН-код:

* после аутентификации пользователю открывается экран ПИН-кода, на котором он может ввести свой ПИН-код;
* если пользователь успешно авторизован в системе, при повторном входе в приложение открывается Экран входа по ПИН-коду, в котором пользователь может ввести ПИН-код либо выйти из приложения для повторной аутентификации.

1. восстановление пароля:

* пользователь может перейти к функции восстановления пароля либо с экрана входа, либо с экрана профиля;
* пользователь вводит электронную почту, на которую приходит шестизначный OTP-код;
* после отправки кода на почту приложение переводит пользователя на экран Ввода OTP-кода, где пользователь должен ввести корректный OTP-код;
* если OTP-код корректен, пользователь переходит на экран Изменения пароля, где вводит и подтверждает новый пароль.

1. локальное хранилище:

* необходимо хранить данные пользователя, необходимые для автоматической аутентификации, в локальном хранилище приложения.

1. профиль пользователя:

* личные данные пользователя сохраняются в его профиле и отображаются на экране Личного профиля;
* на экране Личного профиля пользователь может изменить данные о пользователе: имя пользователя, электронную почту, пароль, аватар и т.д.;
* пользователь может загрузить аватар из галереи устройства.

1. отображение множества объектов данных пользователю:

* данные должны отображаться в удобном для пользователя формате в виде горизонтального и вертикального списка;
* обновление списка объектов по жесту;
* детализация элементов списка:
  + переход на подробное описание элемента по нажатию на элемент;
  + вызов контекстного меню при долгом нажатии на элемент;
  + действия при нажатии на детали на элементе списка (например, добавление в избранное при нажатии на кнопку на элементе списка)
* экран Получение подробностей об одном объекте.

1. фильтрация данных:

* поиск по названию и/или описанию объекта данных;
* фильтрация по нескольким свойствам объекта данных (например, категориям, кол-ву товара, просмотрам и т.д.) отдельно или одновременно.

1. изменение данных о объекте данных:

* создание, редактирование и удаление объектов данных;
* система должна корректно аутентифицировать пользователя и позволять управлять только разрешенными объектами;
* изменение данных должно быть реализовано в удобном для пользователя виде.

1. работа со сложными объектами:

* отображение, фильтрацию, создание, удаление, редактирование сложных объектов (имеющих две и более связи с другими таблицами) или сложных для восприятия (неструктурированных, или связующих таблиц) в удобном человеко-читаемом формате;

1. работа с изображениями:

* все изменяемые медиа должны храниться, создаваться и добавляться из сервиса Supabase.

1. обработка ошибок:

* необходимо корректно обрабатывать запросы к базе данных;
* 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.

1. выход:

* выход из пользовательской сессии;
* удаление данных пользователя из локального хранилища по нажатию соответствующей кнопки.

1. локализация:

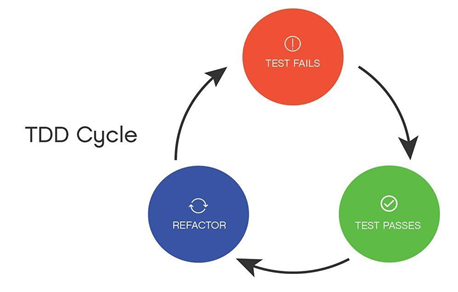
* использовать ресурсы Android там, где это возможно;
* приложение должно быть локализовано на русский и английский язык.

1. Чат с менеджером отеля

* Сообщения пользователя сохраняются в чате с менеджером и отображаются на соответствующем экране.
* На экране Чат с менеджером можно отправлять и редактировать сообщения

## Модуль 5 Тестирование приложения

Реализуйте тестирование согласно методологии TDD и макету:



## Модуль 6 Документирование результатов разработки

Важным этапом любой разработки является документирование результатов разработки, поэтому итогом практики и вашей разработки должны быть следующие шаги:

1. создание отчета по выполненной работе:

* укажите цели и задачи;
* отобразите разработанные вами экраны и опишите их;
* укажите разработанные вами функции приложения.

1. создание презентации. В презентации необходимо рассказать о реализации приложения

* презентация должна быть рассчитана на разработчиков (Вы не должны продавать приложение!);
* о реализованных задачах;
* важные / интересные моменты проделанной работы;
* возникшие трудности;
* используемые архитектурные решения;
* используемые библиотеки и инструменты.

1. подготовка файлов для публикации приложения:

* файлы для публикации загружайте в папку «Publication» в основную ветку main;
* создайте установочный файл с названием «[первая буква фамилии латиницей]-Matule»;
* загрузите иконки приложения для магазина с размерами:
  + 32х32;
  + 64х64;
  + 128х128.
* создайте текстовый документ с кратким описанием приложения;
* добавьте скриншоты экранов приложения в количестве 5 штук.