

**Конкурсное задание**

**по компетенции**

**«Разработка мобильных приложений»**

Сессия 3

Необходимо проанализировать приложение, разработанное другим специалистом, на наличие недостатков и ошибок и исправить их.

В работе необходимо использовать систему контроля версий Git. Ссылка на Git: . Для входа используйте учетную запись вида wsruserX, где X – это номер участника. Необходимо загрузить каждую сессию в отдельную ветку с именем “Session-X”, где Х – это номер сессии. Для каждого проекта необходим отдельный репозиторий.

Проект приложения для анализа доступен по ссылке:

Android (Java):

Android (Kotlin):

iOS:

Необходимо скопировать проект в свой репозиторий.

Во время работы не будет доступа в Интернет, кроме документации и API. Описание протокола API доступно по ссылке:

https://app.swaggerhub.com/apis-docs/WorldSkills-MAD/WorldCinema/1.0.0

Необходимо строго следовать предложенному дизайну. Макеты приложения доступны по ссылке: https://www.figma.com/file/CZyxek6xaYYILO0Ln2eAxW/WorldCinema-Session-3-%D0%9C%D0%B5%D0%B6%D0%B2%D1%83%D0%B7?node-id=0%3A1

1. Проведите Code Review данного проекта. Найденные недостатки (исходного кода, архитектуры, несоответствия code style языка реализации, БЕЗ УЧЕТА БАГОВ И НЕСООТВЕТСТВИЙ МАКЕТУ) следует описать в текстовом документе по образцу:

|  |  |
| --- | --- |
| **Описание найденного недостатка** | Слишком длинный метод (760 строк). |
| **Что плохого в данной реализации?** | Спагетти-код трудно читать, трудно искать ошибки. |
| **Расположение в проекте** | MainActivity.java, строки 345 - 1105. |
| **Возможное решение проблемы** | Разделить длинный метод на части. |

1. Необходимо найти и устранить ошибки (баги) и несоответствия макету и описанию приложения:

Необходимо корректно обрабатывать запросы к серверу. В случае получения ошибки от сервера или отсутствия соединения с сетью Интернет необходимо отобразить соответствующий текст ошибки с помощью диалогового окна.

Функционал приложения:

1. Приложение поддерживает альбомную ориентацию экрана. Портретную ориентацию необходимо запретить.
2. Экран SignIn Screen:
   * При нажатии на кнопку "Войти" необходимо проверять поля для ввода на пустоту, а также email на корректность (требования к email описаны в документации к API). При некорректном заполнении необходимо отобразить ошибку с помощью диалогового окна. При корректном заполнении формы необходимо отправить на сервер соответствующий запрос.
   * При успешной авторизации необходимо осуществлять переход на экран Main Screen. При получении ошибки от сервера необходимо отобразить ее с помощью диалогового окна.
3. Экран Main Screen:
   * Данные о пользователе необходимо запрашивать с сервера.
   * При нажатии на кнопку "Изменить" необходимо реализовать изменение аватара пользователя:
     1. Пользователь выбирает источник фотографии (камера или Галерея), выбор источника следует реализовать с помощью диалогового окна.
     2. Пользователь выбирает фотографию.
     3. Фотография отправляется на сервер.
     4. В случае успеха аватар пользователя заменяется на новый, в случае ошибки она отображается с помощью диалогового окна.
   * При нажатии на кнопку "Выход" необходимо осуществлять переход на экран авторизации.
   * Изображение для обложки необходимо загружать с сервера.
   * Подборки фильмов должны скроллиться по горизонтали. Для получения списка фильмов необходимо использовать соответствующий GET-запрос.
   * При нажатии на конкретный фильм должен осуществляться переход на Movie Screen для соответствующего фильма.
   * При нажатии на изображение последнего просмотренного фильма должен осуществляться переход на экран соответствующего фильма.
4. Экран Movie Screen:
   * Информацию о фильме необходимо получать с сервера.
   * Цвет возрастного ограничения должен зависит от рейтинга фильма согласно макету.
   * Список кадров должен скроллиться по горизонтали.
5. Экран чата еще не реализован, но в проекте уже добавлены методы для реализации следующего функционала:
   * Для сегодняшних сообщений необходимо отобразить заголовок "Сегодня".
   * Необходимо валидировать поле для ввода сообщения чата на пустоту.
6. Необходимо добавить чат согласно макету:
   * Сообщения необходимо упорядочить от старых к новым сверху вниз. Для сегодняшних сообщений необходимо отобразить заголовок "Сегодня".
   * "Облако" сообщения должно растягиваться по содержимому.
   * Последовательно идущие сообщения одного автора необходимо группировать (расстояния между сообщениями должны быть меньше, как на макете).
   * Реализуйте блок отправки сообщения как на макете. При вводе сообщения поле для ввода должно растягиваться по вертикали.
   * При нажатии на кнопку "Отправить" необходимо отправить сообщение на сервер. При позитивном ответе от сервера необходимо отобразить сообщение в чате. При возникновении ошибки - отобразить ошибку с помощью диалогового окна.
   * Необходимо валидировать поле для ввода на пустоту. При отсутствии текста сообщения необходимо отобразить ошибку с помощью диалогового окна.

3. Измените структуру проекта в соответствии с требованием: Проект приложения должен быть структурирован по экранам, то есть исходные файлы конкретного экрана должны быть в соответствующей папке. Общие для нескольких экранов классы необходимо поместить в папку common.

4. Реализуйте Unit-тесты для тестирования следующих методов:

1. Метод валидации email.
2. Метод получения цвета возрастного ограничения.
3. Метод определения даты чата (замены на слово “Сегодня”).
4. Метод валидации поля для ввода сообщения чата.

5. Реализуйте UI-тест отображения ошибки при пустых полях формы авторизации.

6. Реализуйте UI-тест корректной работы кнопки “Выход”.