****

# Задание для демонстрационного экзамена по комплекту оценочной документации № 1.1 по компетенции № F6 «Разработка мобильных приложений»

Вариант 1

Задание включает в себя следующие разделы:

1. Формат Демонстрационного экзамена
2. Формы участия
3. Вид аттестации
4. Модули задания, критерии оценки и необходимое время
5. Необходимые приложения

Продолжительность выполнения задания: 7 ч.

1. Формат Демонстрационного экзамена:

Очный / Дистанционный

1. Форма участия:

Индивидуальная

1. Вид аттестации:

ГИА

1. Модули задания, критерии оценки и необходимое время

Модули и время сведены в Таблице 1.

**Таблица 1.**

| **№ п/п** | **Модуль, в котором используется критерий** | | **Критерий** | **Время выполнения Модуля** | **Проверяемые  разделы WSSS** | **Баллы** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Судейские** | **Объективные** | **Общие** |
|  | Смартфоны (разработка мобильного приложения для онлайн-кинотеатра) | | A - Смартфоны | 3 часа | 1, 3, 4, 5 | 0 | 12,05 | 12,05 |
|  | Планшетные компьютеры (разработка мобильного приложения для онлайн-кинотеатра) | | B – Планшетные компьютеры | 2 часа | 1, 3, 4, 5 | 0 | 11,25 | 11,25 |
|  | Умные часы (разработка приложения для онлайн-кинотеатра) | | C – Умные часы | 45 минут | 1, 3, 4, 5 | 0 | 5,75 | 5,75 |
|  | Защита проекта | | F – Защита проекта | 1 час 15 минут | 2, 3, 5 | 6,3 | 1 | 7,3 |
|  | | **Итого** | | | | 6,3 | 30,05 | 36,35 |

Модули с описанием работ

Модуль 1: Смартфоны (разработка мобильного приложения для онлайн-кинотеатра)

Необходимо разработать мобильное приложение для смартфона, удовлетворяющее следующим требованиям:

Приложение должно поддерживать следующие версии ОС:

* Android 9.0 и новее
* iOS 13.0 и новее

В работе необходимо использовать систему контроля версий Git.

Для входа используйте учетную запись вида userXX, где XX, – это номер участника. Необходимо загрузить результаты выполнения модуля в отдельную ветку с именем “Module-X”, где Х – это номер модуля. Для каждого проекта необходим отдельный репозиторий.

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.

Необходимо строго следовать предложенному дизайну. Макеты приложения доступны по ссылке:

<https://www.figma.com/file/Npzmbq6gSrYD9C2F9EcoZ3/KOD1.1-Variant1?node-id=0%3A1> 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>

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.

Необходимо реализовать следующий функционал:

1. Создайте проект. Настройте иконку приложения согласно макету. Следует учесть разницу в отображении иконок на различных версиях операционной системы.
2. Реализуйте экран Launch Screen согласно макету. Текст должен быть отдельным элементом. Логотип приложения должен быть расположен по центру экрана.
3. Реализуйте экран SignUp Screen согласно макету:
   * При нажатии на кнопку "Зарегистрироваться" 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. Так же необходимо проверять равенство пароля и его повтора.
   * При корректном заполнении формы необходимо отправлять запрос регистрации на сервер. При получении ошибки от сервера ее необходимо отобразить с помощью диалогового окна. При успешной регистрации нужно автоматически осуществить авторизацию и перейти на Main Screen.
   * При нажатии на кнопку "У меня уже есть аккаунт" необходимо осуществлять переход на SignIn Screen.
   * При первом запуске приложения после Launch Screen должен отображаться SignUp Screen. При последующих - SignIn Screen.
4. Реализуйте экран SignIn Screen согласно макету:
   * 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 нажатии на кнопку "Регистрация" необходимо осуществлять переход на SignUp Screen.
   * При успешной авторизации необходимо осуществлять переход на экран Main Screen. При получении ошибки от сервера необходимо отобразить ее с помощью диалогового окна.
5. Реализуйте экран Main Screen согласно макету:
   * Содержимое экрана должно скроллиться по вертикали.
   * Реализуйте рекламный блок в верхней части экрана согласно макету. Для получения информации о фильме-обложке используйте соответствующий GET-запрос.
   * Изображение для обложки необходимо загружать с сервера.
   * При нажатии на кнопку "Смотреть" необходимо осуществлять переход на экран соответствующего фильма.
   * Реализуйте отображение подборок фильмов согласно макету. Подборки должны скроллиться по горизонтали. Для получения списка фильмов используйте соответствующий GET-запрос.
   * При нажатии на конкретный фильм должен осуществляться переход на Movie Screen для соответствующего фильма.
   * В нижней части экрана необходимо реализовать управляющий компонент согласно макету.
   * Реализуйте цветовую дифференциацию активной и неактивных вкладок согласно макету.
6. Реализуйте экран Movie Screen согласно макету:
   * Содержимое должно скроллиться по вертикали. Информацию о фильме необходимо получать с сервера.
   * Цвет возрастного ограничения должен зависит от рейтинга фильма согласно макету.
   * Список кадров должен скроллиться по горизонтали.

Модуль 2: Планшетные компьютеры (разработка мобильного приложения для онлайн-кинотеатра)

Необходимо разработать мобильное приложение для планшета, удовлетворяющее следующим требованиям:

Приложение должно поддерживать следующие версии ОС:

* Android 9.0 и новее
* iOS 13.0 и новее

В работе необходимо использовать систему контроля версий Git.

Для входа используйте учетную запись вида userXX, где XX – это номер участника. Необходимо загрузить результаты выполнения модуля в отдельную ветку с именем “Module-X”, где Х – это номер модуля. Для каждого проекта необходим отдельный репозиторий.

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.

Необходимо строго следовать предложенному дизайну. Макеты приложения доступны по ссылке:

<https://www.figma.com/file/Npzmbq6gSrYD9C2F9EcoZ3/KOD1.1-Variant1?node-id=0%3A1>

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>

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.

Разрешено использовать наработки Модуля 1.

Необходимо реализовать следующий функционал:

1. Создайте проект. Настройте иконку приложения согласно макету. Следует учесть разницу в отображении иконок на различных версиях операционной системы.
2. Реализуйте экран SignIn Screen:
   * Макет для данного экрана отсутствует, Вам необходимо спроектировать экран самостоятельно с учетом рекомендаций конкретной целевой платформы. Следует соблюдать цветовую схему приложения и основные принципы построения пользовательского интерфейса.
   * На экране должны присутствовать следующие элементы: логотип, название, два текстовых поля для ввода email и пароля, кнопка «Войти».
   * 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. Реализуйте экран Main Screen согласно макету:
   * Данные о пользователе необходимо запрашивать с сервера.
   * 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.
   * Реализуйте отображение последнего просмотренного видео. Для получения информации необходимо использовать запрос /usermovies с параметром filter = ”lastView”. На экране необходимо отобразить только последний просмотренный фильм.
   * 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. Реализуйте экран Movie Screen согласно макету:
   * Информацию о фильме необходимо получать с сервера.
   * Цвет возрастного ограничения должен зависит от рейтинга фильма согласно макету.
   * Список кадров должен скроллиться по горизонтали.
   * Реализуйте отображение списка эпизодов согласно макету. Данные необходимо запрашивать с сервера.
   * При нажатии на кнопку «Назад» необходимо возвращаться на Main Screen.

Модуль 3: Умные часы (разработка приложения для онлайн-кинотеатра)

Необходимо разработать мобильное приложение для умных часов.

В работе необходимо использовать систему контроля версий Git.

Для входа используйте учетную запись вида userXX, где XX – это номер участника. Необходимо загрузить результаты выполнения модуля в отдельную ветку с именем “Module-X”, где Х – это номер модуля. Для каждого проекта необходим отдельный репозиторий.

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.

Необходимо строго следовать предложенному дизайну. Макеты приложения доступны по ссылке:

<https://www.figma.com/file/Npzmbq6gSrYD9C2F9EcoZ3/KOD1.1-Variant1?node-id=0%3A1>

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>

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.

Разрешено использовать наработки Модулей 1 и 2.

Необходимо реализовать следующий функционал:

1. Создайте проект. Настройте иконку приложения согласно макету. Следует учесть разницу в отображении иконок на различных версиях операционной системы.
2. Реализуйте экран SignIn Screen согласно макету:
   * 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. Реализуйте экран Main Screen согласно макету. При нажатии на иконку должен осуществляться переход на соответствующий экран. Если экран не описан в Задании - нужно отобразить экран с надписью “В разработке”. При нажатии на кнопку Cancel (для Android - при свайпе вправо) нужно возвращаться на SignIn Screen.
4. Реализуйте экран Chat List Screen согласно макету:
   * Данные необходимо запрашивать с сервера.
   * При нажатии на кнопку Cancel (для Android - при свайпе вправо) нужно возвращаться на Main Screen.

Модуль 4: Защита проекта

Подготовьте презентацию с отчетом о разработанных в рамках Экзамена приложениях. Сравните требования к приложениям и итоговый результат.

Целевая аудитория выступления - менеджеры проекта. Продолжительность выступления - 5 минут.