09.02.07 ПР-418

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПП.12.01 Коммерческие аспекты разработки**

**профессионального модуля**

**ПМ.12 Разработка программных решений**

|  |  |  |
| --- | --- | --- |
| **Практиканты** |  | **Дмитриев Е.В.** |
| **Руководители практики** |  | **Агафонов И.В.** |

**ОГЛАВЛЕНИЕ**

# ВВЕДЕНИЕ

Главными целями и задачами производственной практики ПП.12.01 «Коммерческие аспекты разработки» профессионального модуля ПМ.12 «Разработка программных решений» специальности 09.02.07 «Информационные системы и программирование» является освоение обучающимися вида профессиональной деятельности «Разработка программных решений» и соответствующих профессиональных и общих компетенций:

|  |  |
| --- | --- |
| ПК 12.2 | Создавать приложения со сложной логикой переходов |
| ОК 1. | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам |
| ОК 2. | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |
| ОК 3. | Планировать и реализовывать собственное профессиональное и личностное развитие. |
| ОК 4. | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |
| ОК 5. | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |
| ОК 6. | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей |
| ОК 7. | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. |
| ОК 8. | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности |
| ОК 9. | Использовать информационные технологии в профессиональной деятельности. |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языке |
| ОК 11 | Планировать предпринимательскую деятельность в профессиональной сфере |

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающиеся в ходе выполнения заданий практики ПП.12.01 «Коммерческие аспект разработки» должны приобрести следующий практический опыт:

* разработки мобильных приложений;
* проведения тестирования и отладки приложения;
* использования инструментальных средств на этапе тестирования программного продукта;
* использования системы контроля версий.

В результате освоения программы учебной практики студент будет уметь:

* использовать технологии для разработки мобильных приложений;
* использовать технологии для работы с различными протоколами обмена данными;
* строить приложения со сложной логикой переходов;
* оформлять документацию на программные средства.

За время практики должны быть выполнены следующие виды работ:

* разработка, тестирование приложений для различных устройств под управлением ОС Android;
* подготовка приложений для публикации;
* публикация приложения в Play Market
* продвижение приложения;
* запуск Play Store из приложения;
* презентация решений.

Производственная практика ПП.12.01 «Коммерческие аспекты разработки» проводится на базе профессиональной мастерской «Разработка мобильных приложений» в БПОУ «Омский авиационный колледж имени Н.Е. Жуковского»

# ДНЕВНИК ПРАКТИКИ

|  |  |  |
| --- | --- | --- |
| ***Дата*** | ***Виды выполненных работ*** | ***Подпись руководителя практики*** |
| 21.03.22 | Создание графического интерфейса пользователя и файлов ресурсов. |  |
| 22.03.22 | Реализация запросов к API для получения данных. |  |
| 23.03.22 | Обработка ошибок при работе с веб-сервисом. |  |
| 24.03.22 | Организация хранения данных в памяти устройства. |  |
| 25.03.22 | Реализация логики работы приложения. |  |
| 26.03.22 | Тестирование работы приложения. |  |
| 28.03.22 | Описание продукта. |  |
| 29.03.22 | Создание файла сертификата. |  |
| 30.03.22 | Получение подписанного apk. |  |
| 31.03.22 | Загрузка подписанного apk и тестирование. |  |
| 01.03.22 | Документирование результатов. |  |
| 02.03.22 | Подготовка и защита портфолио. |  |

# ПОСТАНОВКА ЗАДАЧИ

Необходимый функционал приложения для смартфона:

* Иконка приложения должна соответствовать макету
* Экран Launch Screen. Текст на экране должен быть отдельным элементом. Логотип приложения должен быть расположен по центру экрана.
* Экран SignUp Screen. Реализована кнопка регистрации. При нажатии на кнопку “У меня уже есть аккаунт” открывается окно с авторизацией.
* Экран SignIn Screen. Реализована кнопка “Войти”. При нажатии на кнопку “Регистрация” открывается окно регистрации.
* Экран Main Screen. На экране отображаются обложки фильмов из подборки “new”. Обложки отображаются в виде карусели, листаются с помощью жеста swipe. При нажатии на обложку открывается экран Chat Screen для соответствующего фильма.
* Экран Profile Screen. Данные о пользователе запрашиваются с сервера. При нажатии на кнопку “изменить” реализовано изменение аватара пользователя. При нажатии на “выход” открывается экран авторизации. При нажатии на кнопку “обсуждения” открывается соответствующий экран.
* Экран Chat List Screen. Необходимо отобразить постер фильма, название чата, последнее сообщение. При нажатии на элемент списка открывается чат соответствующего фильма.
* Экран Chat Screen. Сообщения в чате должны отображаться согласно макету. Реализовано поле отправки сообщения.
* Экран Collection Screen. На экране отображаются созданные коллекции.
* Экран Create Collection Screen. При открытии экрана выбирается случайная иконка. При нажатии на кнопку “выбрать иконку” открывается окно со списком иконок.

Необходимый функционал приложения для смартфона:

* Иконка приложения настроена согласно макету
* Экран SignIn Screen. При нажатии на кнопку “войти” отправляется запрос на сервер.
* Экран Main Screen. При нажатии на иконку осуществляется переход на соответствующий экран. При нажатии на кнопку “cancel” происходит возврат на SignIn Screen.
* Экран Chat List Screen. Данные запрашиваются с сервера. При нажатии на кнопку “cancel” происходит возврат на экран Main Screen.
* Экран Movies Screen. На данном экране отображаются фильмы из подборки “new” в виде списка с вертикальной прокруткой.

# РЕАЛИЗАЦИЯ ФУНКЦИЙ ПРИЛОЖЕНИЯ

Описываем лишь некоторые функции приложения:

1. регистрация, авторизация, сохранение токиена в памяти устройства – приводим скан(Ы) экранов + программный код с комментариями c обязательно обработка ошибок при работе в Интернет
2. разработка экрана с чатом – скан экрана + программный код (и компоновка, и

коды классов)

1. подготовка приложенИЙ (их у вас 2) для публикации – подробно по этапам

*порядок создания цифровой подписи и создание apk файла.*

*представьте разработанные маркетинговые материалы*

Экран авторизации

В листинг 1 представлен код авторизации пользователя. Авторизация происходит средствами API, которое возвращает токен пользователя. Токен пользователя сохраняется в системе, с помощью объекта SharedPreference. Макет экрана можно рассмотреть на рисунке 1.

Листинг 1 - Авторизация пользователя

public void doLogin() { //авторизация  
 AsyncTask.*execute*(()->{ //запускаем в отдельном потоке  
 IApi.login(GetParametrs()).enqueue(new Callback<LoginResponce>() { //вызываем метод login (указан в интерфейсе), передаем тело запроса  
 @Override  
 public void onResponse(Call<LoginResponce> call, Response<LoginResponce> response) { //в случае успешного подключения  
 if(response.isSuccessful()){  
 sp = getApplicationContext().getSharedPreferences("settings", Context.*MODE\_PRIVATE*); //создаем файл с настройками приложения  
 SharedPreferences.Editor editor = sp.edit(); //создаем объект для редактирования настроек  
 editor.putString("token", response.body().getToken()); //записываем токен в настройки  
 editor.apply();//сохраняем токен пользователя  
 Intent intent = new Intent(getApplicationContext(), Main.class);  
 startActivity(intent);  
 }  
 else{  
 Toast.*makeText*(getApplicationContext(), "Ошибка: " + response.code(), Toast.*LENGTH\_LONG*);  
 }  
 }  
 @Override  
 public void onFailure(Call<LoginResponce> call, Throwable t) {  
 Toast.*makeText*(getApplicationContext(), t.getLocalizedMessage(), Toast.*LENGTH\_LONG*);  
 }  
 });  
 });  
}

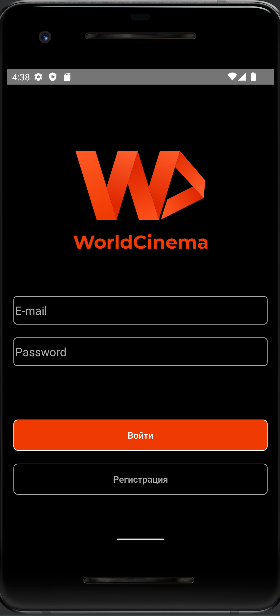


Рисунок - Экран авторизации

Экран регистрации

В листинг 2 представлен код регистрации пользователя. При нажатии на кнопку регистрация происходит вызов метода doRegister(), который отправляет тело запроса с данными пользователя на сервер. Макет экрана регистрации представлен на рисунке 2.

Листинг 2 - Регистрация пользователя

public void doRegister() {  
 AsyncTask.*execute*(()->{  
 iApi.register(GetParametrs()).enqueue(new Callback<RegistrationResponse>() {//вызываем метод регистрации  
 @Override  
 public void onResponse(Call<RegistrationResponse> call, Response<RegistrationResponse> response) {  
 if(response.isSuccessful()){  
 Toast.*makeText*(getApplicationContext(), "Успешно", Toast.*LENGTH\_LONG*);  
 Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivity(intent);  
 }  
 else{  
 Toast.*makeText*(getApplicationContext(),”Ошибка: " + response.code(), Toast.*LENGTH\_LONG*);  
 }  
 }  
 @Override  
 public void onFailure(Call<RegistrationResponse> call, Throwable t) {  
 Toast.*makeText*(getApplicationContext(), t.getLocalizedMessage(), Toast.*LENGTH\_LONG*);  
 }  
 });  
 });  
}

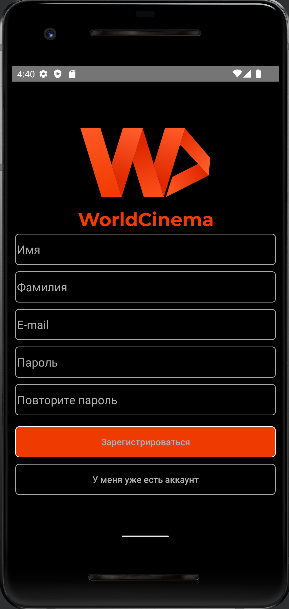


Рисунок - Экран регистрации

# ЗАКЛЮЧЕНИЕ

*что было сделано за практику, чему научились и какие цели достигли и какие задачи выполнили.*

# СПИСОК ЛИТЕРАТУРЫ

*Не менее 10 источников по работе в среде разработки Android Studio.*