

Выбранное средство разработки программного продукта (платформа «1С», решение «1С:Управление небольшой фирмой 8») полностью себя оправдало. В ходе разработки основные цели проекта были достигнуты, и результатом является комплексный программный продукт, позволяющий эффективно управлять деятельностью клуба: составлять расписание занятий, отслеживать посещения, управлять абонементом и инвентарем, хранить и оперативно изменять информацию о клиентах, вести бухгалтерский и управленческий учет.

ЛИТЕРАТУРА

1. Селищев Н. 1С. Управление небольшой фирмой 8.2. Автоматизация малого бизнеса. – М: Рид Групп — 2011. – 304 с.
2. Мокина Е.Е., Марухина О.В., Фисоченко О.Н., Берестнева Е.В. Информационная система поддержки принятия решений для выпускников бакалавриата. // Информационное общество — 2014. №3 — С. 20-24.
3. Хрусталева Е.Ю. 101 Совет начинающим разработчикам в системе "1С:Предприятие 8. – М: Рид Групп — 2014. – 213 с.
4. Черемных С.В., Семенов И.О., Ручкин В.С. Моделирование и анализ систем. IDEF-технологии – М: Финансы и статистика. — 2006. — 188 с.

ПРОЕКТИРОВАНИЕ ANDROID-ПРИЛОЖЕНИЯ ДЛЯ ЗАКАЗА УСЛУГ В МНОГОФУНКЦИОНАЛЬНОМ КОМПЛЕКСЕ

О.И. Минаева

*Научный руководитель: В.С. Шерстнев, к.т.н., доцент кафедры ВТ ИК ТПУ
(г. Томск, Томский политехнический университет)
e-mail: olua94@mail.ru*

DESIGNING OF ANDROID APPLICATION FOR SERVICES BOOKING IN MULTIFUNCTIONAL COMPLEX

O.I.Minaeva

(Tomsk, Tomsk Polytechnic University)

This article describes a process of Android application designing, which includes the definition of the app's key functions. Also the article includes MVVM pattern's description, which is used in Android apps.

Keywords: Android OS, Java, mobile application development, MVVM pattern, services booking.

Введение. Мобильные устройства уже давно стали неотъемлемой частью нашей жизни. Но сейчас это уже не просто телефоны, предназначенные для совершения звонков и отправки текстовых сообщений. Сейчас смартфоны должны включать в себя все необходимое для общения, образования, работы и развлечения. Именно поэтому на данный момент существует большое количество различных приложений, игр и прочих программ, разработанных специально для таких целей. Популярность мобильных приложений неизменно растет, ведь пользоваться ими удобно и быстро.

Одним из обширных направлений, для которого создаются мобильные приложения, является сфера заказа услуг. Сюда можно включить всевозможные сервисы заказа билетов, такси, автомойки и т.д. Основным преимуществом таких программ является более быстрое оформление бронирования по сравнению со звонком оператору или заказом на сайте.

В данной статье будет рассмотрен процесс проектирования Android-приложения, позволяющего совершить заказ услуг на автомойке комплекса, а также отображающего актуальную информацию обо всех направлениях, новостях и акциях данного комплекса.

Обзор аналогов. Прежде чем перейти непосредственно к проектированию, необходимо было найти и проанализировать существующие решения в интересующей области. В процессе анализа приложений были найдены и рассмотрены следующие аналоги: мобильные приложения для записи на автомойку «Freewash», «МойкаМойка», «Где мойка?», «Автомойки Онлайн», «Новая мойка» [1-5]. После анализа были выделены общие достоинства и недостатки, встречающиеся в рассмотренных приложениях. К достоинствам можно отнести:

- наличие авторизации пользователя;
- отображение общей информации об автомойке;
- возможность просмотра истории моек;
- возможность оценивания мойки.

К недостаткам некоторых приложений можно отнести:

- невозможность выбора отдельных услуг, только пакетом услуг;
- нет информации по автомойкам г. Томска;
- нет возможности редактирования существующей записи.

На основании исследования аналогов были выявлены следующие функции, которые должны быть реализованы в приложении:

1. авторизация пользователя;
2. выбор определенных услуг автомойки;
3. отправка заказа;
4. просмотр существующих заказов;
5. оценивание совершенного заказа;
6. редактирование и удаление заказа.

При наличии всех приведенных выше функций, приложение будет не только удобно в использовании, но и даст возможность пользователю выбирать услуги на основе их оценки.

Проектирование. В общем случае, Android-приложение состоит из:

- Java-классов, являющихся подклассами основных классов из Android SDK (View, Activity, ContentProvider, Service, BroadcastReceiver, Intent) и Java-классов, у которых нет родителей в Android SDK;
- манифеста приложения;
- ресурсов наподобие строк, изображений и т.п [6].

В плане обработки взаимодействия между пользовательским интерфейсом и его логикой Android следует архитектурному шаблону «Model-View-ViewModel» (MVVM). Структура шаблона представлена на рисунке 1.

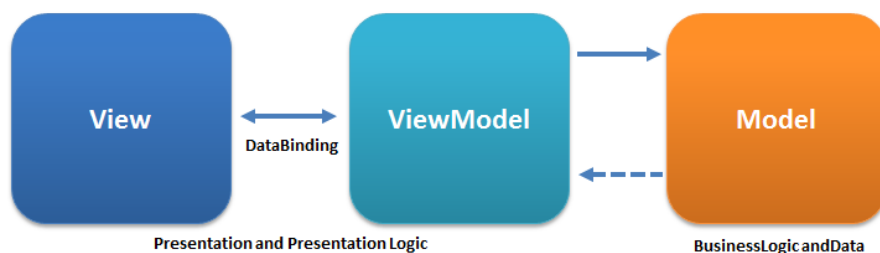


Рисунок 1. Структура шаблона MVVM

Разработка пользовательского интерфейса совершается с помощью технологии XML (файлы слоев, layout files). Логика пользовательского интерфейса реализуется как компонент ViewModel. Функциональные связи между пользовательским интерфейсом и ViewModel реализуются через привязки обработчика (bindings), которые, по сути, являются правилами типа «если кнопка А была нажата, должен быть вызван метод onButtonAClick() из

ViewModel». Биндинги могут быть написаны в коде или определены декларативным путём [7].

В начале работы необходимо определить основные пользовательские экраны (View) приложения. На основе выявленных в предыдущем пункте функций были выбраны следующие:

1. экран авторизации;
2. меню;
3. экран выбора времени мойки;
4. экран выбора автомобиля;
5. экран выбора услуг;
6. экран подтверждения заказа;
7. экран просмотра существующих заказов.

Общую структуру экранов можно увидеть на рисунке 2.

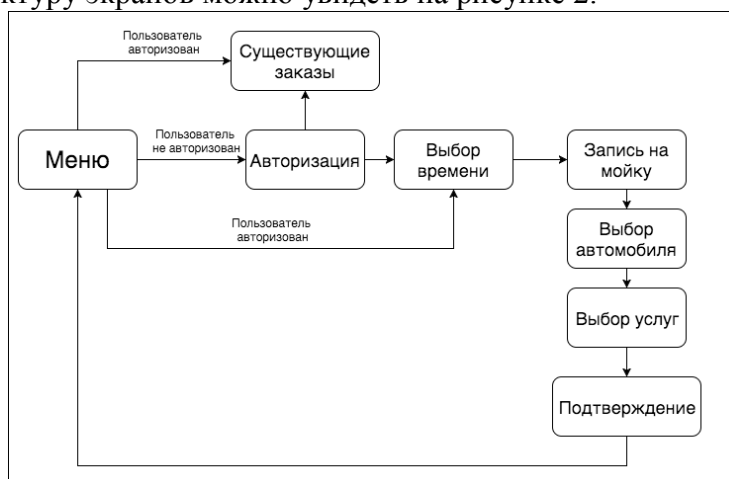


Рис. 2. Структура основных экранов приложения

Экраны выбора автомобиля, услуг и подтверждения заказа для удобства навигации было решено реализовать вкладками в рамках одного окна (рисунок 3). Каждый из экранов представляется в проекте в виде файла слоев.



Рис. 3. Прототип окна «Запись на мойку»

Все перечисленные ранее файлы слоев привязываются к классам, унаследованным от класса Activity. Класс Activity и его подклассы содержат логику, лежащую за пользовательским интерфейсом. При ближайшем рассмотрении этот класс соответствует ViewModel в архитектурном шаблоне Model-View-ViewModel (MVVM). Отношение между подклассом Activity и пользовательским интерфейсом – это отношение один к одному; обычно каждый подкласс Activity имеет только один связанный с ним слой пользовательского интерфейса, и наоборот.

Activity имеет жизненный цикл. В течении жизненного цикла Activity может находиться в одном из трёх состояний:

- Активно и выполняется – этот пользовательский интерфейс находится на переднем плане (т.е. на вершине стека Activity).
- Приостановлено – если данный интерфейс пользователя потерял фокус, но всё ещё видим. В таком состоянии никакой код не выполняется.
- Завершено – если интерфейс пользователя невидим. В таком состоянии код не выполняется [6].

В Java-приложении должен быть как минимум один класс, который наследуется от родительского класса Activity [8].

Средства разработки. Для разработки описываемого приложения была выбрана Android Studio – среда разработки приложений с инструментами Android SDK, базирующаяся на платформе IntelliJ IDEA компании JetBrains. Большим плюсом Android Studio является встроенная поддержка всех основных систем контроля версий (VCS). Также для разработки используется язык Java – объектно-ориентированный язык программирования.

Кроме того, в приложении будет использована библиотека для подгрузки изображений AndroidImageSlider и библиотека для возможности выбора пользователем нужной даты Material DateTime Picker.

Заключение. В результате была спроектирована структура основных экранов приложения и выбраны средства для разработки Android-приложений. Кроме того, был разобран шаблон MVVM для дальнейшего применения в проекте.

ЛИТЕРАТУРА

1. Приложение «Freewash» [Электронный ресурс]. – Режим доступа: <http://free-wash.ru/>, свободный. Дата обращения: 30 марта 2016 г.
2. Приложение «МойкаМойка» [Электронный ресурс]. – Режим доступа: <http://moikamoika.ru/>, свободный. Дата обращения: 30 марта 2016 г.
3. Сервис «Где мойка?» [Электронный ресурс]. – Режим доступа: <http://www.gdemoyka.ru/>, свободный. Дата обращения: 30 марта 2016 г.
4. Приложение «Автомойки Онлайн» [Электронный ресурс]. – Режим доступа: <http://xn--80aesdfknb7a.xn--80asehdb/>, свободный. Дата обращения: 30 марта 2016 г.
5. Приложение «Новая мойка» [Электронный ресурс]. – Режим доступа: <https://play.google.com/store/apps/details?id=com.loyaltyplant.partner.avtomoikanizhniinov>, свободный. Дата обращения: 30 марта 2016 г.
6. Архитектура Android-приложений. Часть III — основные части приложения [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/141201/>, свободный. Дата обращения: 30 марта 2016 г.
7. Архитектура Android-приложений. Часть II — архитектурные стили и шаблоны [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/140655/>, свободный. Дата обращения: 30 марта 2016 г.
8. Класс Activity [Электронный ресурс]. – Режим доступа: <http://microsin.net/programming/android/class-activity.html>, свободный. Дата обращения: 30 марта 2016 г.