

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ВЕДЕНИЯ ТОРГОВ НА ИНТЕРНЕТ-АУКЦИОНЕ EBAY ДЛЯ ПЛАТФОРМЫ GOOGLE ANDROID

А. П. Полищук¹, С. А. Семериков¹, И. А. Теплицкий¹,
Е. П. Линник², И. И. Линник²

¹ Украина, г. Кривой Рог, Криворожский национальный университет

² Украина, г. Киев, Национальный авиационный университет

Операционная система Google Android открывает большие возможности для разработки программного обеспечения мобильных устройств. Android имеет потенциал для устранения препятствий на пути к успеху в деле разработки и реализации нового поколения программного обеспечения для мобильных устройств. Точно так же, как стандарты платформ PC и Macintosh создали рынки для настольного и серверного программного обеспечения, Android, предоставляя стандартную среду для мобильных приложений, позволит создать рынок мобильных приложений, предоставив разработчикам возможность получать большую прибыль от реализации своих программных продуктов.

Учитывая, что мобильные устройства в настоящее время становятся самым распространенным средством для коммуникации и осуществления доступа в сеть Internet, а ОС Google Android является современной перспективной платформой для таких мобильных устройств, представляет интерес возможность использования мобильных устройств на платформе Google Android как средство для обучения программированию.

Корпорацией Google разработан ряд средств разработки для Google Android (в т.ч. для начинающих – Google App Inventor); в основе всех их лежит Android SDK (<http://developer.android.com>). Рекомендуемой интегрированной средой разработки приложений для Google Android является Eclipse с установленным плагином The Android Development Tools (ADT).

Разработчику приложений доступны следующие утилиты из набора Android SDK: Android Emulator, Android Virtual Devices (AVDs), Hierarchy Viewer, Draw 9-patch, Dalvik Debug Monitor Service (ddms), Android Debug Bridge (adb), Android Asset Packaging Tool (aapt), Sqlite3, Traceview, mksdcard, dx, UI/Application Exerciser Monkey. Наиболее важной утилитой из набора SDK является эмулятор, представляющий собой QEMU-приложение, эмулирующее виртуальное ARM мобильное устройство, в котором можно запускать приложения Android. Эмулятор запускает полный стек Android системы, вплоть до уровня ядра, вклю-

чающий в себя набор предустановленных приложений (таких, как Dialer), к которым можно получить доступ из вашего приложения. Вы можете выбрать, какая версия Android системы запустится в эмуляторе с помощью настроек AVDs. Также можно настроить внешний вид мобильного устройства и параметры экрана. При запуске эмулятора и во время выполнения, можно использовать целый ряд команд и опций для контроля за его поведением.

Дополнительно к пользовательским возможностям, Android эмулятор поддерживает различные аппаратные функции, которые могут быть присущи мобильному устройству, в том числе: процессор ARM v5; 16-ти битный LCD дисплей; одна или несколько клавиатур; звуковой чип, поддерживающий ввод и вывод звука; флеш-карта, эмулируемая через файл образа на компьютере разработчика.

Центральным свойством Google Android является то, что одно приложение может использовать элементы другого приложения (если другое приложение позволяет это). Для того, чтобы это работало, система должна иметь возможность запускать процесс, если какая-либо часть этого процесса необходима, и инстанцировать Java-объекты этой части. Поэтому, в отличие от большинства других систем, приложение Google Android не имеет единой точки входа (такой как функция main(), например). Можно сказать, что приложение состоит из связанных компонентов которые система инстанцирует и запускает по мере необходимости.

Для разработки Android-приложений предоставляется четыре типа элементов, которые можно использовать в сочетании друг с другом. Информация о том, какие из этих элементов будут использованы в конкретном приложении, хранится в файле манифеста AndroidManifest.xml.

1. Activity (Деятельность) представляет собой отдельный экран в приложении, реализованный в виде экземпляра класса, наследуемого от базового класса Activity. Деятельности позволяют отображать пользовательские элементы интерфейса на экране и обрабатывать возникающие события. В большинстве случаев приложение имеет несколько экранов, которые могут взаимодействовать между собой при помощи возвращаемых ими значений.

2. Для перемещения между экранами используется специальный класс Intent (Намерение), который описывает, какие действия хочет выполнить приложение. Наиболее важными составными частями намерения являются действие и данные, над которыми это действие выполняется. Например, чтобы посмотреть контактную информацию конкретного человека, вам нужно создать намерение, действием для которого будет VIEW (Просмотр), а данными – идентификатор этого человека, называемый URI (Uniform Resource Identifier – унифицированный иден-

тификатор ресурса).

Как было сказано выше, Intent нужно для указания действий, которые хочет выполнить приложение. А для того, чтобы Activity знало, как реагировать на эти действия, используется класс IntentFilter (фильтр намерений). Например, чтобы на экран можно было вывести контактную информацию о человеке, нужно опубликовать IntentFilter, который сообщит экрану, как обрабатывать действием VIEW связанные с ним данные о человеке. Все деятельности публикуют свои фильтры намерений в файле манифеста.

3. Service (Служба) – это компонент приложения, который позволяет работать ему в фоновом режиме без использования интерфейса пользователя. Одним из ярких примеров использования сервиса является проигрывание музыкальных файлов, когда пользователь выбирает файл для проигрывания или редактирует плейлист в одной или нескольких Activity, но воспроизведением музыки занимается не сама Activity, а Service, который продолжает воспроизводить музыку даже после того, как пользователь покинет Activity.

Можно подключиться (bind to) к работающей службе, или стартовать службу, если она еще не работает. При подключении вы можете взаимодействовать со службой через интерфейс который она предоставляет. Для музыкальной службы, возможно, этот интерфейс позволяет пользователям выбрать паузу, перемотку, остановку и перезапуск воспроизведения

4. Content Provider (Контент-провайдер). Программы могут хранить свои данные в файлах, базах данных SQLite и т.п. Контент-провайдеры служат удобным механизмом, посредством которого данные ваших приложений могут совместно использоваться другими приложениями. Класс ContentProvider реализует множество методов для работы с данными (сохранение, поиск и т.п.).

В большинстве случаев каждое Android-приложение запускается в своем собственном Linux-процессе. Процесс создается, когда приложению требуются выполнить какой-либо код, и остается запущенным, пока приложение используется. По завершению процесса система освобождает свою память для использованию другими программами.

Аукцион eBay – крупнейший в мире аукцион в сети Интернет. Доступ к сайту eBay.com для поиска и просмотра информации о товарах открыт для всех. После регистрации открывается возможность размещать свои товары для продажи, а также делать ставки на товары, размещенные на аукционе.

Основным интерфейсом доступа пользователей к аукциону со стационарного компьютера является доступ через любой браузер. Для раз-

работчиков приложений и информационных систем существует возможность доступа к информации сайта eBay через ряд API Web-служб. Таким образом, возможно разрабатывать собственные приложения без привязки к Web-интерфейсу сайта eBay.com. Приложение может представить данные таким способом, который наилучшим образом соответствует потребностям пользователей.

Сервисы eBay поддерживают протокол XML/HTTPS. В XML-RPC API приложение создает запрос в виде строки XML, устанавливает некоторое количество параметров заголовка HTML, и посылает запрос на eBay используя протокол HTTPS.

После обработки запроса, eBay сервер посылает ответ обратно в приложение, также через HTTPS. Этот ответ состоит из XML документа, содержащего данные, которые являются результатом первоначального запроса. Затем приложение анализирует XML для извлечения данных.

Вся совокупность вызовов к сервисам eBay поделена на несколько библиотек API: eBay Trading API; eBay Feedback API; eBay Shopping API; eBay Finding API; eBay Merchandising API.

Рассмотрим задачу: необходимо разработать приложение для операционной системы Google Android, позволяющее пользователю работать в качестве покупателя с Интернет-аукционом eBay.com.

В приложении должны быть реализованы следующие функциональные возможности:

- ведение на мобильном устройстве локальной базы интересующих нас лотов, за процессом торгов по которым мы будем наблюдать, т.е. локальный список наблюдения (Watch List);

- подгрузка лотов в локальный Watch List должна осуществляться путем запроса к разделу My eBay Watch List сервиса eBay.com. Подгружаться и храниться должна информация о наименовании лота, фотографии, типе аукциона, начальной цене, текущей ставки, времени начала и окончания аукциона, возможности осуществления немедленной покупки, информация о продавце и способах доставки товара;

- поиск и загрузка в список наблюдения лота по его уникальному идентификатору;

- процедура предложения ставки для аукционов с типом Chinese;

- процедура осуществления немедленной покупки для аукционов с типом Fixed Price, и Chinese with Fixed Price;

- ведение локального списка лотов по которым пользователь сделал ставки, локальный список ставок (Bid List). Подгружаемая информация также, что и для лотов хранящихся в Watch List;

- перенос лотов из локального списка Bid List в списки выигранных или проигранных аукционов по окончании срока действия аукциона,

- Won List и Loss List соответственно;
- просмотр истории сделанных ставок;
 - удаление лотов из всех локальных списков;
 - периодический, в фоновом режиме, с задаваемым в настройках интервалом, опрос состояния лотов находящихся в локальных списках Watch List и Bid List;
 - оповещение пользователя при наличии изменений по интересующим лотам;
 - контроль пользователем срока окончания действия уникального идентификатора доступа к сервисам eBay.com, получение нового идентификатора при окончании срока действия текущего.

Общая структура мобильного клиента доступа к Web-сервисам Интернет-аукциона eBay.com приведена на рис. 1.

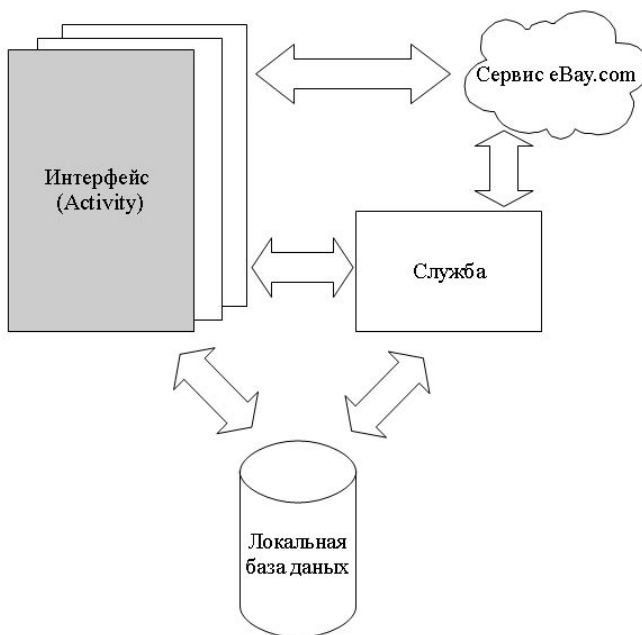


Рис. 1. Общая структура приложения

Для хранения локальных списков лотов применяется SQLite3, являющаяся составным элементом платформы Google Android.

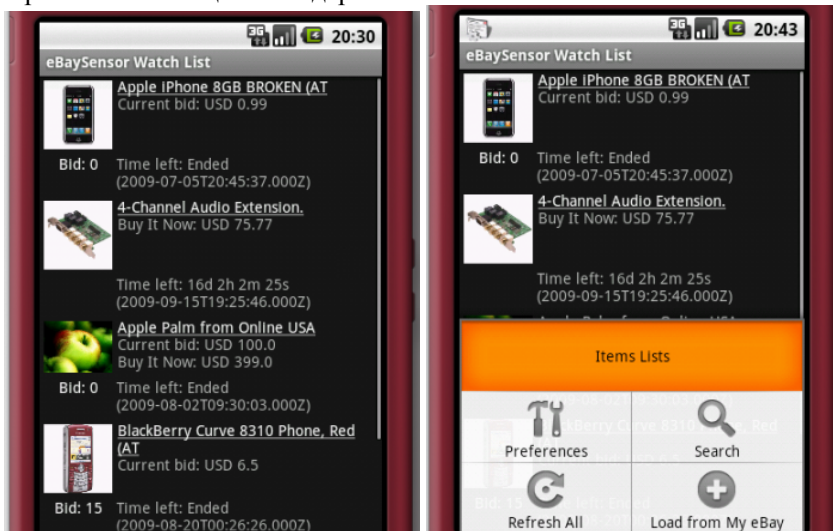
Доступ к различным API сервисов eBay (например Trading API, Shopping API) инкапсулируется в соответствующих классах.

Периодический опрос состояния наблюдаемых лотов осуществляется объектом класса Service платформы Google Android.

При обнаружении изменений, произошедших с наблюдаемыми лотами с момента последнего опроса их состояния, производятся изменения в локальной базе данных и выдается сообщение в статусной строке.

Интерфейс пользователя реализуется с помощью набора наследников класса Activity, используя как созданные на этапе разработки схемы размещения графических элементов управления, так и создаваемые динамически в процессе работы программы.

Настраиваемые параметры приложения сохраняются между сеансами работы с помощью стандартного механизма SharedPreferences.



a

б

Рис. 2. Таблица лотов

Внешний вид заполненной данными из Watch List таблицы представлен на рис. 2а. Обработчик для строки таблицы запускает деятельность, в которой отображаются детальная информация о данном лоте (рис. 2б).

Одной из опций доступных, в меню является возможность загрузки в локальный список данных о наблюдаемых лотах из соответствующего раздела My eBay сайта eBay.com.

В зависимости от того, к какому типу аукциона относится данный лот и в каком состоянии он находится, над ним могут быть выполнены те или иные действия, выбираемые из меню.

Для активного лота Chinese аукциона, с возможностью оплаты по фиксированной цене, перечень возможных действий представлен на рис. 3а.

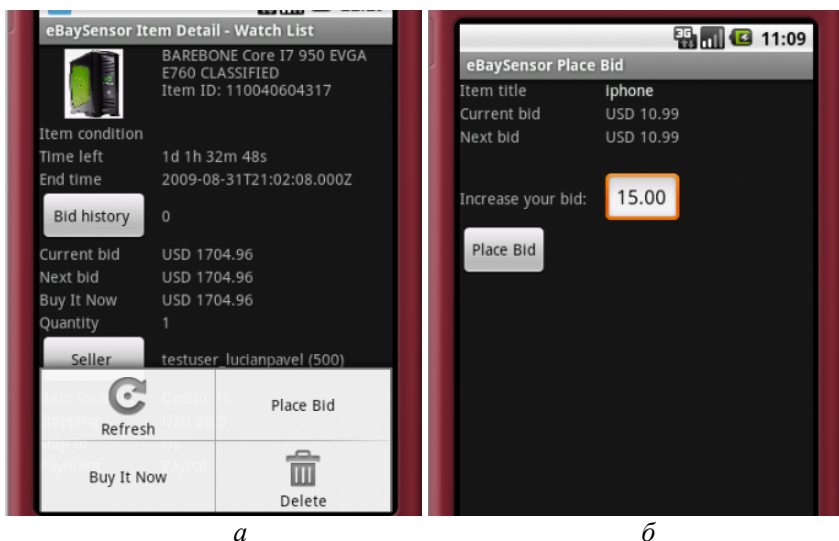


Рис. 3. Участие в аукционе

Таким образом, мы имеем возможность обновить информацию о лоте, сделать ставку, купить по фиксированной цене, либо удалить лот из списка просмотра. Для того, чтобы сделать ставку, выбираем пункт меню «Place Bid». Этот пункт меню вызывает деятельность PlaceBid.java, внешний вид которой изображен на рис. 3б.

При наличии ошибки сообщение о ней выводится на экране в диалоговом окне, в противном случае ставка считается принятой и информация о данном лоте помещается в список лотов, по которым сделаны ставки (Bid List).

Таким образом, в конечном итоге любой из лотов, находящийся в списке наблюдения (Watch List), либо в списке лотов, по которым сделаны ставки (BidList) должен попасть в список приобретенных (Won List) либо проигранных (Lost List) лотов.

Комплексный характер данной задачи позволяет проиллюстрировать как основные возможности платформы Google Android для создания мобильных приложений, так и использование Web-служб. Последнее направление является перспективным для создания на платформе Google Android мобильных приложений для доступа к облачным интерфейсам. Так, для студентов специальности «Прикладная математика» можно предложить задачу создания мобильного интерфейса к одной из Web-СКМ (например, Sage).