

УДК 004

## **БАЗИРОВАНИЕ ЖИЗНЕННОГО ЦИКЛА ANDROID ПРИЛОЖЕНИЯ НА ПРИМЕРЕ РАЗРАБОТКИ ДОСКИ ОНЛАЙН ОБЪЯВЛЕНИЙ**

Мерман Д.Г., студент гр. ПИБ-171, III курс

Павлов М.В., студент гр. ПИБ-171, III курс

Архандеев Д.К., студент гр. ПИБ-171, III курс

Научный руководитель: Тайлакова А.А., старший преподаватель

Кузбасский государственный технический университет

имени Т.Ф. Горбачева

г. Кемерово

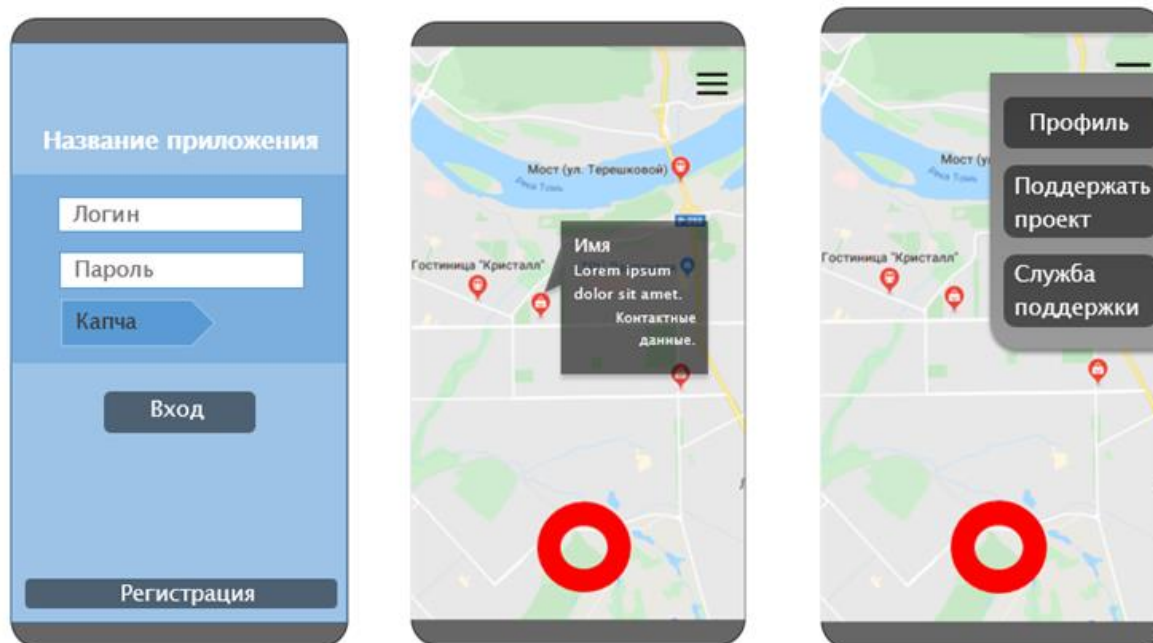
В данной работе пойдет речь о создании мобильного приложения, предоставляющего услуги онлайн доски объявлений «Рука помощи». До широкого распространения интернета, люди в основном пользовались печатными изданиями для публикации объявлений или расклеивали их в общедоступных местах. Сегодня эффективность таких способов крайне мала, поиск нужной услуги или нужного товара при таком подходе крайне затруднен. Приложение «Рука помощи» предоставляет пользователям следующие возможности:

- добавление заявок на оказание услуги;
- отображение заявки на карте;
- рассылка уведомлений;
- консультация со службой поддержки;
- добавление отзывов;
- доступ к личному кабинету;
- осуществление платежей.

Применение данного мобильного приложения сделает процесс публикации объявлений проще. Достаточно ввести минимально необходимую информацию о товаре, который вы хотите продать или услуге, которую можете предоставить, и нажать пару кнопок для того, чтобы ваше объявление увидели все пользователи приложения. В случае если вы ищете какой-то определенный товар или услугу, вам достаточно ввести в поисковую строку запрос. Таким образом, онлайн доска объявлений позволяет во много раз ускорить и упростить работу с объявлениями. Интерфейс приложения представлен на рисунке (рис. 1)

Процесс разработки приложений для мобильных устройств во многом аналогичен разработке десктопных приложений для ПК. Однако, существуют работы операционных системам для мобильных устройств. Мобильные приложения, в сравнении с десктопными платформами, имеют ограничения на использование источника питания, гораздо меньший технический потенциал и ограниченный экран. Основная техническая разница между операционными системами мобильных устройств и операционными систем настольных компьютеров заключается в том, что мобильная операционная система не является по-

настоящему многозадачной системой – на мобильных устройствах только одно приложение может быть активным в одно и то же время. Когда другое приложение запускается, работающее приложение, переводится в фоновый режим и остается в этом режиме, пока пользователь не обратится к нему снова. Если приложение остается в фоновом режиме слишком долго, или если доступной оперативной памяти становится слишком мало, операционная система может его завершить. Такая последовательность переходов между различными состояниями приложения называется его жизненным циклом.



*Рисунок 1 – Интерфейс приложения*

Жизненный цикл приложения базируется на взаимодействии с пользователем и потребности операционной системы в памяти и других вычислительных ресурсах. Когда пользователь взаимодействует с устройством, он может переключаться между различными приложениями или различными внешними представлениями одного и того же приложения. Когда это происходит, приложение проходит через различные состояния, требуя от разработчика, чтобы он управлял этими переключениями без потерь данных и без лишних задержек в реализации задач. Глубокое понимание жизненного цикла приложения и его тщательное проектирование чрезвычайно важны для разработки успешных мобильных программ.

Чтобы понять жизненный цикл приложения Android сначала полезно осмыслить состояния, в которых может находиться приложение Android, с точки зрения пользователя:

1. **Исполнение.** Когда пользователь прикасается к значку приложения, приложение запускается и становится видимым для пользователя.

2. **Пауза.** При взаимодействии с приложением пользователь может отвлечься и не прикасаться к экрану в течение какого-то периода времени, то есть пользователь прекращает работу с приложением на какой-то более-менее заметный период времени. В этом случае приложение затемняется и переходит в

состояние паузы.

3. Остановка. Если пользователь не притрагивается к экрану в течение более длительного периода времени, в результате чего экран становится черным, или же пользователь запускает другое приложение, так что начальный приложение становится невидимым – происходит переход в состояние остановки.

4. Завершение. Если устройство перезагружается, или пользователь запускает ряд других приложений – исходное приложение может быть завершено операционной системой, чтобы высвободить ресурсы для других приложений, с которыми пользователь взаимодействует в настоящий момент времени.

Разработчик должен хорошо понимать, что происходит, когда приложение переходит из одного состояния в другое, как обрабатывается каждое из этих состояний. Для этого требуется понимание жизненного цикла приложений для Android.

Жизненный цикл начинается с момента касания значка приложения. В этот момент вызывается на выполнение метод `onCreate()` – режим ожидания приложения. Он содержит код для отображения на экране всех элементов, включенных в разметку этой активности. Разработчик должен дополнить этот метод кодом, который инициализирует переменные и объекты разметки до значений, необходимых, чтобы пользователь мог начать взаимодействовать с приложением.

После того, как активность стартовала, выполняется метод `onResume()`. Этот метод не обязательно реализовывать, но это очень полезно делать, чтобы вернуть приложение в состояние выполнения, в котором оно было до перехода в состояние паузы. Это относится к системным службам, которые используются приложением, перезапуску анимации, а также к любым другим настройкам, необходимым, чтобы восстановить работу пользователя с приложением.

Когда пользователь окончательно останавливает взаимодействие с приложением, начинается ветвь его завершения. Рассмотрим несколько основных методов реализации завершения:

- `onPause()`. Этот метод должен использоваться для остановки используемых приложением служб, для остановки анимации, для сохранения важной информации о состоянии с тем, чтобы пользователь мог возобновить использование приложения из того состояния, в котором он его оставил.

- `onStop()`. Этот метод должен гарантировать, что все важные данные приложения переданы на постоянное хранение для того, чтобы они не потерялись, когда системные ресурсы данного приложения будут перераспределены между прочими приложениям.

- `onDestroy()`. Этот метод вызывается для «увольнения» всех ресурсов приложения. Это последний шанс для сохранения важных данных программы перед их уничтожением.

Правильное кодирование методов, связанных с жизненным циклом приложения, гарантирует эффективное распределение ресурсов девайса и верное восприятие пользователями итогового продукта.

**Список литературы:**

1. Habr [Электронный источник] – URL: <https://habr.com/ru>. – (Дата обращения: 18.01.2020).
2. Жизненный цикл андроид приложения [Электронный источник] – URL: <https://developer.android.com/guide/components/activities?hl=ru>. – (Дата обращения: 18.01.2020).