

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и
информационных технологий

**Реализация мобильного приложения для сервиса оповещения о
социальной активности с элементами геолокации**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 421 группы
направления 09.03.01 «Информатика и вычислительная техника»
факультета компьютерных наук и информационных технологий
Филатовой Анастасии Сергеевны

Научный руководитель
ассистент _____

_____ М.В. Белоконь

Заведующий кафедрой
к. ф.-м.н., доцент _____

_____ Л.Б. Тяпаев

Саратов 2018 год

Введение. Всего несколько десятилетий назад компьютеры были огромными машинами, размером с целую комнату. Сегодня же для повседневных задач мы используем небольшие портативные устройства, такие как смартфоны или планшеты, которые не требуют постоянного подключения к электросети и помещаются в карман. Такие устройства можно использовать в любом месте, где бы вы ни находились, дома, на улице, в транспорте – использовать установленные программы и получать доступ к необходимой информации можно из любой точки земного шара.

Социальная жизнь во все века играла очень важную роль в жизни общества. Люди встречаются, общаются, делятся знаниями и впечатлениями, смотрят фильмы или ходят на выставки – в XXI веке каждый сможет найти, как наиболее интересно провести свое свободное время.

Целью данной выпускной квалификационной работы является написание мобильного приложения для сервиса оповещения о социальной активности с элементами геолокации. Для достижения поставленной цели необходимо выполнить следующие задачи:

- выбрать платформу, для которой будет реализовано приложение;
- ознакомиться с особенностями разработки приложений под выбранную платформу;
- выбрать средства разработки;
- разработать и реализовать базу данных для приложения;
- разработать и реализовать серверную часть для связи приложения с базой данных;
- разработать и реализовать приложение-клиент.

В разделе 1 «Microsoft SQL Server» рассматриваются особенности популярной системы управления базами данных Microsoft SQL Server. Дается определение базы данных и описывается реляционная модель данных.

Раздел 2 «Hibernate и JPA» описывает преимущества использования технологии Hibernate и основные аннотации JPA.

В разделе 3 «Maven» рассмотрены отличительные особенности инструмента для автоматизации сборки Maven, названы основные фазы жизненного цикла проекта, описаны параметры, необходимые для добавления плагинов и сторонних зависимостей.

В разделе 4 «Android» кратко описана история возникновения смартфонов на платформе Android, назван основной функционал таких устройств, приведена статистика изменения распределения версий системы за период с января по май 2018 года. В этой главе рассмотрены особенности разработки под операционную систему Android, такие как использование файла манифеста, разрешений и компонента Activity. Также здесь кратко охарактеризованы используемые виртуальные системы Dalvik и Android RunTime.

В разделе 5 «Gradle» приведено описание системы автоматической сборки Gradle и используемого в ней языка Groovy, а также вида блоков операций для каждой фазы жизненного цикла основного компонента сборки – задачи.

Раздел 6 «Трехзвенная архитектура “клиент-сервер”» рассматривает основные особенности трехзвенной клиент-серверной архитектуры и отличие ее от двухзвенной модели.

В разделе 7 «Реализация приложения» описаны принципы взаимодействия клиента с сервером и сервера с базой данных, а также пользователей приложения друг с другом. Показан интерфейс программы и возможности взаимодействия пользователя с ней. Рассмотрены основные имеющиеся аналоги и их отличия.

1 Microsoft SQL Server. Microsoft SQL Server – это одна из наиболее популярных систем управления реляционными базами данных в мире.

База данных – это организованное определенным образом хранилище данных. Реляционная база данных реализует реляционную модель данных, предполагающую, что все данные в таблице представляют собой набор взаимосвязанных таблиц, каждая из которых содержит множество объектов одного и того же типа. Каждая строка в таблице содержит набор атрибутов – параметров, характеризующих один объект. Все объекты одной таблицы имеют одинаковый набор атрибутов. Каждая таблица базы должна иметь первичный ключ – уникальный идентификатор, позволяющий однозначно идентифицировать конкретную строку.

Таблицы реляционной базы данных должны быть приведены в соответствие с требованиями нормальных форм. Нормализация отношений необходима для обеспечения непротиворечивости и целостности данных, исключение дублирования и приближения базы данных к структуре реального мира.

SQL Server обладает высокой скоростью работы, возможностью шифрования данных, собственной системой аутентификации и большим набором дополнительных средств от производителя.

Работа с базой данных производится на языке SQL. SQL Server использует расширение языка SQL собственного производства – Transact-SQL, предоставляющий некоторые дополнительные возможности, такие как использование локальных и глобальных переменных, расширенный набор функции для работы со строками, датами и другие [1–3].

2 Hibernate и JPA. JPA – это технология, обеспечивающая объектно-реляционное отображение простых JAVA объектов и предоставляющая API для сохранения, получения и управления такими объектами.

Существует несколько реализаций этого интерфейса, одна из самых популярных использует для этого Hibernate.

Связь классов Java и таблиц базы данных может осуществляться как посредством маппинга в .xml-файле настроек, так и с помощью аннотаций JPA.

Аннотации делятся на аннотации классов и аннотации атрибутов. Аннотации классов устанавливаются на сам класс, аннотации атрибутов – на соответствующие поля класса.

Из аннотаций классов главными являются @Entity и @Table. @Entity указывает на то, что данный класс является сущностью модели данных. @Table ставит в соответствие данной сущности таблицу базы данных.

Аннотации атрибутов позволяют установить набор свойств любому полю класса, например, определить соответствующий столбец таблицы в базе данных, указать длину и тип значения, тип связи между таблицами [4–7].

3 Maven. Maven – это инструмент для автоматизации сборки Java-проекта, позволяющий компилировать проект, создавать его jar-файл, дистрибутив программы и генерировать документацию.

В число основных преимуществ можно отнести независимость от операционной системы, простоту управления зависимостями, возможность сборки из консоли, интеграцию со всеми основными Java IDE.

Одним из отличий Maven'a от аналогов является уже определенный жизненный цикл проекта – список фаз, выполняющийся в строгом порядке.

Главным отличием Maven является декларативный подход к построению скрипта, в то время как его основной конкурент – Ant, использует императивный подход [8].

Maven неразрывно связан с таким понятием, как архетип – это шаблон, по которому создается структура проекта в зависимости от его типа [9, 10].

4 Android

Первым смартфоном, использующим операционную систему Android, стал HTC Dream, выпущенный в октябре 2008 года, а уже в 2017 году доля мобильных устройств, работающих на этой системе, превысила 85%.

Операционная система Android построена на основе Linux, вследствие чего обладает высокой производительностью и открытым исходным кодом.

Устройства, работающие на платформе Android, обладают широким функционалом, по сути делающим их портативными компьютерами. В дополнение к имеющимся возможностям пользователь может устанавливать любые дополнительные приложения.

На сегодняшний день на платформе Android работают уже не только смартфоны и планшетные компьютеры, но также и некоторые стационарные компьютеры, электронные книги, наручные часы, телевизоры, игровые консоли, аудио- и видеоплееры.

Регулярно выходят новые версии системы, исправляются старые ошибки и добавляются новые возможности. Каждая новая версия обладает преимуществами, недоступными в предыдущих [11,12].

4.1 Особенности разработки под платформу Android

Основная информация о приложении содержится в файле AndroidManifest.xml. В нем прописываются следующие данные:

- уникальный идентификатор приложения;
- объявление и описание всех используемых компонент;
- список всех необходимых внутренних и внешних разрешений [13].

Одним из основных компонентов Android-приложения является Activity – активность или деятельность. Активность – это инструмент приложения, который отвечает за графический интерфейс конкретного окна,

а также за все работу пользователя с этим окном. Для каждой активности необходимо прописать файл разметки (возможно, не один) – визуальный шаблон пользовательского интерфейса, настройка и управление которым осуществляется из класса соответствующей активности.

Каждая активность имеет свой жизненный цикл, состоящий из следующих возможных состояний:

- активная;
- приостановленная;
- остановленная.

Отличительной особенностью приложений, написанных под Android, от настольных является отсутствие единой точки входа и привилегированности программ [14–16].

Приложения под Android пишутся для установленной на платформе виртуальной машины – Dalvik или Android RunTime.

Dalvik использует тип компиляции Just-In-Time, то есть перевод кода в машинный язык производится в момент выполнения программы при каждом запуске. ART использует схему Ahead-Of-Time, производя интерпретацию кода один раз – в момент установки приложения [17–20].

5 Gradle. Gradle — система автоматической сборки, построенная на принципах систем Ant и Maven.

Система Gradle использует объектно-ориентированный язык программирования Groovy, являющийся, по сути, языком Java с дополнительными возможностями, взятыми из языков Ruby, Python и Smalltalk. Gradle использует скрипт, написанный на Groovy, с набором готовых функций для сборки проектов.

Основные настройки проекта находятся в файле с названием build.gradle. Это программа на Groovy, позволяющая настроить конфигурацию самого Android-проекта, номера используемых версий и SDK,

версию приложения. Остальные настройки позволяют определить используемую версию JDK, зависимости и многое другое.

Основным компонентом сборки проекта является `task` – задача, представляющая собой набор инструкций, которые необходимо выполнить во время сборки проекта. Операции для задачи можно разделить на отдельные функциональные блоки. Эти блоки будут выполняться последовательно по фазам жизненного цикла. Каждая задача имеет свой жизненный цикл, состоящий из трех состояний – инициализации, конфигурации и выполнения [21–24].

6 Трехзвенная архитектура "клиент-сервер". "Клиент-сервер"

– это архитектура, в которой происходит разделение участников взаимодействия на две группы:

- серверы – поставщики сервисных функций;
- клиенты – заказчики услуг у серверов.

Двухзвенная модель предполагает, что функции бизнес-логики выполняются на стороне клиента, а сервер является только поставщиком запрашиваемой информации.

На смену двухзвенной модели пришла усовершенствованная трехзвенная архитектура, предполагающая, что множество клиентов обращается к одному или нескольким серверам, и обращение к базе данных происходит уже с сервера. Разделение функций приложения происходит следующим образом:

- программа-клиент реализует функции компонента представления, здесь происходит процесс ввода-вывода данных и взаимодействие с пользователем;
- программа-сервер представляет собой прикладной компонент и выполняет функции бизнес-логики;

- сервер базы данных представляет собой реализацию компонента доступа к ресурсам, предоставляющего по запросу необходимую информацию из базы данных [25-27].

7 Реализация приложения. В соответствии с задачей практики было реализовано мобильное приложение, сохраняющее информацию в единую базу данных.

Для связи с базой данных на языке Java было написано серверное приложение, использующее для доступа к данным технологии JPA (Java Persistence API) и Hibernate. Связь клиента с сервером осуществляется посредством сокетов. Клиент отправляет серверу номер функции, которую необходимо выполнить, а также некоторые необходимые для ее выполнения данные. Сервер на основе полученных данных генерирует SQL-запрос и пересылает пользователю полученные блоки данных.

Программа-клиент написана в IDE Android Studio и протестирована на встроенном эмуляторе на модели телефона LG Nexus 5.

Для сохранения пользовательских данных было создано хранилище данных SharedPreferences, в которое можно сохранить данные в виде пары ключ-значение.

Для предотвращения загрузки сервера добавлять мероприятия могут только пользователи, обладающие статусом "Суперпользователь". Данный статус предоставляется модератором активным пользователям, посещающим мероприятия и оставляющим комментарии к ним. Также в целях безопасности все добавленные или измененные мероприятия перед выводом должны быть проверены модератором на соответствие российскому законодательству и действительность.

В системе присутствуют четыре вида ролей пользователя:

- незарегистрированный пользователь;
- пользователь;

- суперпользователь;
- модератор.

При первом запуске приложение автоматически определяет местоположение пользователя и прорисовывает карту города, в котором он находится. После выбора города перед пользователем открывается карта со всеми добавленными и подтвержденными мероприятиями, обозначенными в виде маркеров разных цветов. Цвет маркера отвечает за тип соответствующего мероприятия. В настройках программы пользователь может установить, мероприятия каких типов его интересуют изменить город, а также увидеть, мероприятиям каких типов соответствуют разные цвета маркеров.

При нажатии на маркер появляется информационное окно с основными данными о событии. Нажатие на само информационное окно откроет страницу с полными данными о мероприятии. Зарегистрированные пользователи увидят внизу кнопку с надписью "Я пойду" для сохранения статистики посещений в своем личном кабинете.

В личном кабинете отображаются введенные при регистрации данные, а также все добавленные им мероприятия и мероприятия, которые он посещал или собирается посетить. Все мероприятия разделены на соответствующие группы. К прошедшим событиям можно оставлять комментарии с оценками от 0 до 5.

7.1 Сравнение с имеющимися аналогами. Разработанное приложение не является принципиально новым, на рынке имеются различные аналоги, обладающие своими особенностями, преимуществами и недостатками. Основными являются:

- Афиша — расписания и билеты в кино;
- Eventr – куда пойти? Мероприятия и акции, афиша;
- KudaGo – афиша Москвы, Петербурга;

- **Мой гид.**

7.1.1 Афиша — расписания и билеты в кино. Приложение "Афиша" работает со всеми городами России, поддерживает покупку билетов, поддерживает возможность комментирования оценивания событий. Зарегистрированные пользователи могут добавлять в избранное понравившиеся им места и хранить историю посещений.

7.1.2 Eventr – куда пойти? Мероприятия и акции, афиша. "Eventr" предоставляет удобные возможности поиска, в личном кабинете можно собирать статистику о своих добавленных и интересных событиях, увидеть информацию о подписках.

7.1.3 KudaGo – афиша Москвы, Петербурга. Приложение "KudaGo" позволяет пользователям возможности поиска интересных их событий по разным фильтрам или по карте Google Maps. Присутствует возможность добавления мероприятия в "Избранное" и синхронизация с сервисом "ВКонтакте", предоставляющая возможность следить за обновлениями друзей.

7.1.4 Мой гид. Приложение "Мой гид" предоставляет доступ к событиям пяти крупнейших городов Саратовской области. Оно поддерживает работу с картографическим сервисом Google Maps и указывает погоду в городе.

Заключение. В результате написания выпускной квалификационной работы было разработано и реализовано мобильное приложение, предоставляющее пользователям возможности поиска событий города по интересам, ведения учета посещенных мероприятий и создания собственных. Для приложения были разработаны и написаны база данных и серверная программа. В процессе были изучены основы и особенности разработки под мобильную операционную систему Android, основы работы с сервисом Google Maps и его веб-службами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 METANIT.COM – Сайт о программировании. Введение в MS SQL Server и T-SQL [Электронный ресурс]. URL: <https://metanit.com/sql/sqlserver/1.1.php> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 2 ЯКласс. Реляционная база данных и ее особенности. Виды связей между реляционными таблицами [Электронный ресурс]. URL: <http://www.yaklass.ru/materiali?ctid=511&mode=cht> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 3 Руководства HOSTINGER. Реляционные СУБД – сравнение MySQL и SQL сервер [Электронный ресурс]. URL: <https://www.hostinger.ru/rukovodstva/reljacionnye-subd-sravnenie-mysql-i-sql-server/> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 4 Java онлайн для разработчиков. Hibernate в вопросах и ответах [Электронный ресурс]. URL: <http://java-online.ru/hibernate-faq.xhtml#proxy> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 5 JavaStudy – руководство для web-разработчика. Собеседование по Java EE — Hibernate Framework [Электронный ресурс]. URL: <http://javastudy.ru/interview/jee-hibernate-questions-answers/> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 6 JavaStudy – руководство для web-разработчика. Аннотации Java для работы с базой данных [Электронный ресурс]. URL: <http://javastudy.ru/interview/jee-hibernate-questions-answers/> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 7 CUBA Platform. Developer’s Manual. Аннотации сущностей [Электронный ресурс]. URL: https://docs.cuba-platform.com/cuba/5.6/manual/ru/webhelp/entity_annotations.html (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.

- 8 IT SPHERE. Gradle Tutorial часть 1 (Теория) – Gradle vs Maven vs Ant [Электронный ресурс]. URL: <http://it-channel.ru/2015/07/21/gradle-vs-mvn-vs-ant/> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 9 Quizful – тесты онлайн. Автоматизированные сборки в Java [Электронный ресурс]. URL: <http://www.quizful.net/post/automated-builds-java> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 10 Java онлайн для разработчиков. Фреймворк Apache Maven [Электронный ресурс]. URL: <http://java-online.ru/maven-pom.xhtml> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 11 Тарифкин.ру – универсальный интернет-помощник в мобильном мире. Что такое андроид и зачем он нужен [Электронный ресурс]. URL: <https://tarifkin.ru/mobilnye-os/chto-takoe-android-i-zachem-nuzhen> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 12 SETPHONE – смартфоны и сотовые телефоны: обзоры, новости, статьи. Самые первые смартфоны: какими они были [Электронный ресурс]. URL: <https://setphone.ru/stati/samye-pervye-smartfony-kakimi-oni-byli/> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 13 Сайт Александра Климова – Освой программирование играючи. Файл манифеста AndroidManifest.xml [Электронный ресурс]. URL: <http://developer.alexanderklimov.ru/android/theory/AndroidManifestXML.php> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 14 Сайт Александра Климова – Освой программирование играючи. Activity (Активность, Деятельность) [Электронный ресурс]. URL: <http://developer.alexanderklimov.ru/android/theory/activity-theory.php> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 15 Сайт Александра Климова – Освой программирование играючи. Жизненный цикл приложения на Android [Электронный ресурс]. URL: <http://developer.alexanderklimov.ru/android/theory/lifecycle.php> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.

- 16 Вычислительные сети. Теория и практика. Особенности разработки ПО для ОС Android [Электронный ресурс]. URL: <http://network-journal.mpei.ac.ru/cgi-bin/main.pl?ar=1&l=ru&n=21&pa=3> (Дата обращения: 02.06.2018). — Загл. с экрана. — Яз. рус.
- 17 UPGRADE-ANDROID.RU. Что такое Виртуальная машина Dalvik (Dalvik VM) [Электронный ресурс]. URL: <http://upgrade-android.ru/stati/stati/507-что-такое-virtualnaya-mashina-dalvik-dalvik-vm.html> (Дата обращения: 08.06.2018). — Загл. с экрана. — Яз. рус.
- 18 4DPA. Эра Dalvik в системе Android подходит к концу [Электронный ресурс]. URL: <http://4pda.ru/2013/11/09/123502> (Дата обращения: 08.06.2018). — Загл. с экрана. — Яз. рус.
- 19 Компьютерная грамотность, помощь и ремонт. Разбираемся В Тонкостях Программ Art И Dalvik [Электронный ресурс]. URL: <http://pcyk.ru/program/razbiraemsa-v-tonkostyax-programm-art-i-dalvik/> (Дата обращения: 08.06.2018). — Загл. с экрана. — Яз. рус.
- 20 AndroidInsider.ru – лучшее из мира Android. ART и Dalvik: Как оно работает [Электронный ресурс]. URL: <https://androidinsider.ru/os/art-i-dalvik-kak-ono-rabotaet.html> (Дата обращения: 08.06.2018). — Загл. с экрана. — Яз. рус.
- 21 Блог Java программиста. Про Gradle для любопытных [Электронный ресурс]. URL: <http://programador.ru/in-the-gradle> (Дата обращения: 05.06.2018). — Загл. с экрана. — Яз. рус.
- 22 habr.com. Gradle: управляя зависимостями [Электронный ресурс]. URL: <https://habr.com/company/redmadrobot/blog/275515/> (Дата обращения: 05.06.2018). — Загл. с экрана. — Яз. рус.
- 23 habr.com. Подробно о задачах Gradle [Электронный ресурс]. URL: <https://habr.com/post/167227/> (Дата обращения: 05.06.2018). — Загл. с экрана. — Яз. рус.
- 24 Сайт Александра Климова – Освой программирование играючи. Задачи Gradle [Электронный ресурс]. URL:

http://developer.alexanderklimov.ru/android/theory/gradle_task.php (Дата обращения: 05.06.2018). — Загл. с экрана. — Яз. рус.

25 Национальный открытый университет "ИНТУИТ". Академия Microsoft: Базы данных. Лекция 3: Различные архитектурные решения, используемые при реализации многопользовательских СУБД. Краткий обзор СУБД [Электронный ресурс]. URL: <https://www.intuit.ru/studies/courses/508/364/lecture/8643?page=2> (Дата обращения: 09.06.2018). — Загл. с экрана. — Яз. рус.

26 Национальный открытый университет "ИНТУИТ". Академия Microsoft: Современные веб-технологии. Лекция 4: Архитектурные особенности проектирования и разработки Веб-приложений [Электронный ресурс]. URL: <https://www.intuit.ru/studies/courses/611/467/lecture/28784?page=2> (Дата обращения: 09.06.2018). — Загл. с экрана. — Яз. рус.

27 Все о цифровом мире. Что такое Трехзвенная (многозвенная) архитектура "клиент – сервер" [Электронный ресурс]. URL: <http://prokomputer.ru/forum/threads/chto-takoe-trexyzvennaja-mnogozvennaja-arxitektura-klient-server.118/> (Дата обращения: 09.06.2018). — Загл. с экрана. — Яз. рус.