

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование кафедры)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Прикладная информатика в социальной сфере

(направленность (профиль)/специализация)

## БАКАЛАВРСКАЯ РАБОТА

на тему «Разработка программного обеспечения для клиент-серверного  
мобильного приложения и web-приложения по истории ТГУ»

Студент

А.В. Мищенко

(И.О. Фамилия)

(личная подпись)

Руководитель

В.С. Климов

(И.О. Фамилия)

(личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

«\_\_» \_\_\_\_\_ 20\_\_ г.

Тольятти 2019

## АННОТАЦИЯ

Тема выпускной квалификационной работы: «Разработка программного обеспечения для клиент-серверного мобильного приложения и web-приложения по истории ТГУ».

Объект исследования – процесс организации доступа к контенту.

Предмет исследования – технологии организации доступа к контенту.

Цель работы: разработка программного обеспечения, обеспечивающего доступ к клиент-серверному приложению «Хроники ТГУ».

Задачи: Анализ и характеристика предметной области; Выбор технологии разработки программного обеспечения; Реализация и тестирование приложения.

Структура работы представлена тремя главами, посвященными, соответственно, изучению предметной области, выбору технологии реализации, описанию физической реализации. Иллюстративный материал – таблицы, схемы и экранные формы.

Первая глава выпускной квалификационной работы содержит анализ предметной области, обоснование выбора платформы и среды разработки.

Вторая глава – моделирование программного обеспечения.

Третья глава – физическая разработка программного обеспечения мобильного и клиент-серверного приложения с использованием выбранных способов, тестирование.

В заключении даны теоретические выводы по главам и отчет по итогам практической реализации.

Результатом выпускной квалификационной работы является программное обеспечение, реализующее доступ к клиент-серверному приложению.

Работа содержит три главы, 42 страницы, 24 рисунка и 2 таблицы.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
Глава 1 АНАЛИЗ ПРОБЛЕМЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ .....	6
1.1 Анализ и характеристика предметной области .....	6
1.2 Обзор существующих технологий реализации.....	8
1.3 Обоснование выбора технологии реализации .....	12
Выводы по первой главе.....	14
Глава 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	15
2.1 Моделирование бизнес-процессов .....	15
2.2 Моделирование технологии реализации .....	20
2.3 Описание технологии реализации.....	23
Выводы по второй главе.....	25
Глава 3 ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	26
3.1 Разработка интерфейса.....	26
3.2 Разработка базы данных .....	29
3.3 Разработка мобильного приложения .....	32
Выводы по третьей главе .....	38
ЗАКЛЮЧЕНИЕ .....	39
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	40

## ВВЕДЕНИЕ

Применение информационных технологий позволяет существенно расширить область распространения исторических знаний. В 2018 году кафедра «История и философия» ТГУ приступила к работе над коллективной монографией по истории Тольяттинского государственного университета. Одновременно было принято решение о разработке мобильного и web-приложений «Хроники ТГУ», реализующих возможности для более эффективного продвижения собранных в ходе работы над книгой материалов.

Объектом исследования является процесс организации доступа к контенту.

Предмет исследования – технологии организации доступа к контенту.

Новизна работы заключается в подходах к реализации онлайн-доступа к историческому контенту.

Методы, используемые при работе над ВКР – клиент-серверные технологии разработки приложений; объектно-ориентированное программирование на языке Java; диаграммы Entity Relationship, реализованные в Visual Paradigm for UML; диаграммы состояния процессов в нотациях IDEF0 и BPMN.

Цель работы. Разработка программного обеспечения, обеспечивающего доступ к клиент-серверному приложению «Хроники ТГУ».

Задачами выпускной квалификационной работы являются: анализ и характеристика предметной области; выбор технологии разработки программного обеспечения; реализация и тестирование приложения.

Актуальность заявленной темы обусловлена наличием заказа на разработку электронной версии книги. Задание ректора на создание электронной версии монографии по истории ТГУ было дано кафедре «История и философия» на заседании комиссии по финансово-хозяйственной деятельности Гуманитарно-педагогического института.

В настоящее время представление электронного варианта печатных материалов является обязательным для издания, ориентированного на широкую

аудиторию и представляющего информационную ценность. Это гарантирует доступность информации и ее широкое распространение.

В основу концепции разрабатываемого приложения заложена идея реконструкции истории: теоретический материал книги дополнен визуальными элементами, такими как галерея, документальные источники, исторические хроники, представляющие трехуровневый срез истории с 50-х гг. прошлого столетия до современности.

Область применения – кафедра «История и философия» ТГУ для продвижения информации по истории ВУЗа.

Выпускная квалификационная работа состоит из введения, трех глав, заключения и списка литературы.

Первая глава выпускной квалификационной работы посвящена анализу предметной области, изучению уровня автоматизации организации и ее структуры, а также выбору технологии разработки.

Вторая глава содержит моделирование бизнес-процессов, описание выбранной технологии реализации.

Третья глава – физическая разработка приложения с использованием выбранных способов, тестирование и этапы внедрения.

В заключении даны теоретические выводы по главам и отчет по итогам практической реализации. Результатом работы является готовая реализация программного обеспечения для мобильного и веб-приложения по истории ТГУ.

При работе над выпускной квалификационной работой была изучена литература и информационные ресурсы на русском и английском языках [1–30].

Работа содержит три главы, 24 рисунка, 2 таблицы.

# **Глава 1 АНАЛИЗ ПРОБЛЕМЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ**

## **1.1 Анализ и характеристика предметной области**

На начальном этапе разработки приложения проводится анализ предприятия и его структуры, а затем определение модели будущей системы [16].

Организационная структура Тольяттинского государственного университета включает 10 институтов, Научно-исследовательскую часть, институт Дополнительного образования «Жигулевская долина», СДО «Росдистант», а также ряд административных подразделений.

Кафедра «История и философия» является структурным подразделением Гуманитарно-педагогического института Тольяттинского государственного университета (рисунок 1).

Кафедра «История и философия» была создана в 1951 году как кафедра истории основ марксизма-ленинизма (в дальнейшем кафедра общественных наук) вечернего отделения Куйбышевского индустриального института, а позднее Ставропольского филиала ВУЗа. Первым преподавателем и руководителем стал доктор исторических наук, профессор А.Э. Лившиц.

В 1967 году филиал был преобразован в Тольяттинский политехнический институт, в котором выделились 24 кафедры. Кафедра общественных наук разделилась на кафедру политической экономии, истории КПСС, философии и научного коммунизма. Именно кафедра истории КПСС стала центром исторической науки города.

В настоящее время кафедра «История и философия» ведет обучение по направлению бакалавриата (очная и дистанционная формы обучения). Возглавляет кафедру кандидат исторических наук, доцент О.А. Безгина.

В 2018 году кафедра приступила к работе над научной монографией, отражающей историю Тольяттинского государственного университета, а также историю города через этапы становления и развития университета.

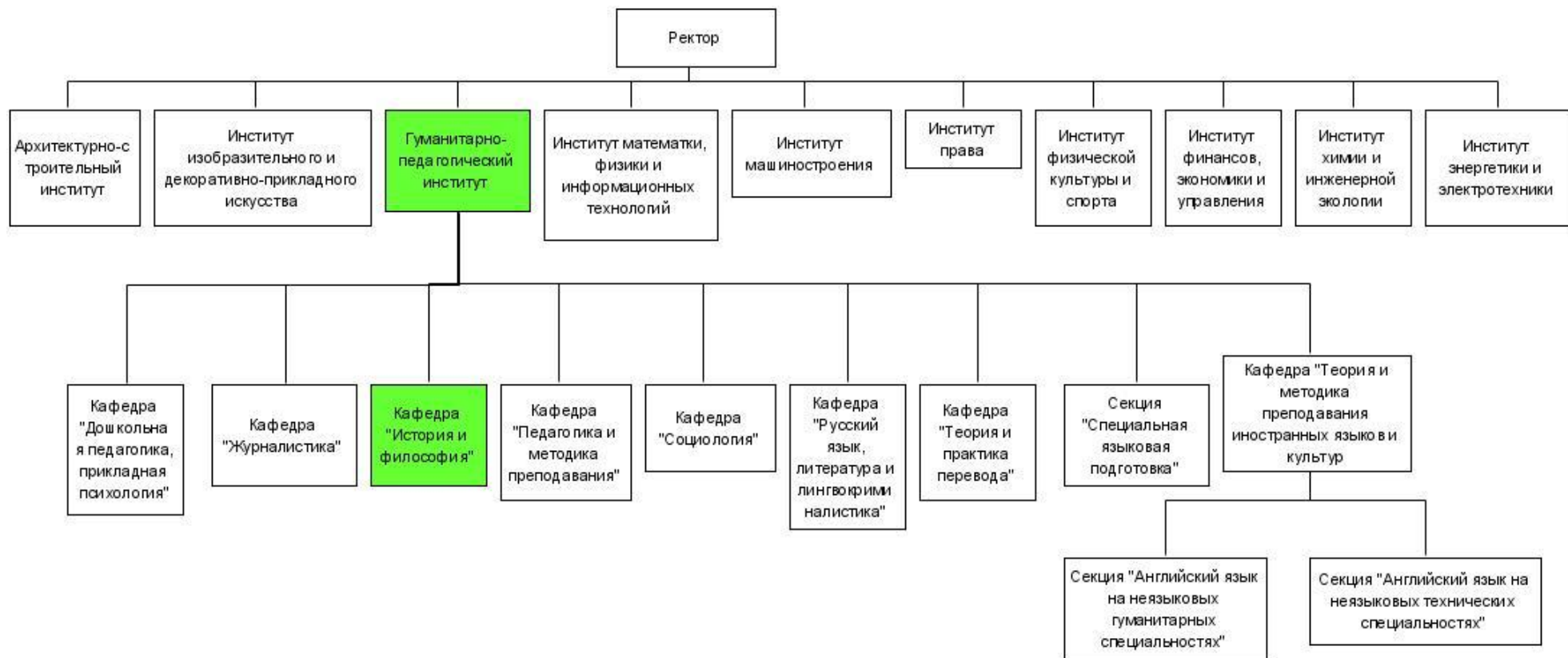


Рисунок 1 – Кафедра «История и философия» в структуре институтов Тольяттинского государственного университета

Оптимальным способом представления и распространения книги и собранных материалов является создание онлайн-приложения. Формат приложения позволяет разместить текст печатного издания и дополнить его интерактивными элементами. Таким образом, электронная версия книги приобретает самодостаточность полноценного тематического ресурса.

## **1.2 Обзор существующих технологий реализации**

Выбор способа разработки программного обеспечения основывается на анализе существующих технологий, выявлении плюсов и минусов рассматриваемых вариантов.

Приложение состоит из двух частей.

1. Клиентская часть – front-end (фронтэнд) – представляет собой видимый пользователю интерфейс, реализуется на компьютере пользователя.
2. Серверная часть – back-end (бекэнд) – пассивная, отвечает на пользовательский запрос, выступает хранилищем данных.

Для клиентской части используется язык разметки гипертекста HTML5 [8], стилевые таблицы CSS и язык программирования JavaScript [29, 30]. HTML5 создает структуру документа, определяет основные компоненты страницы. Внешний вид, местоположение и, отчасти, связь элементов (компонентов) определяется при помощи CSS. Для добавления динамичности и взаимодействия служит JavaScript.

Существует ряд подходов к разработке клиентской части приложения.

Стандартный проект, не требующий крупных временных затрат, может быть выполнен с использованием системы управления контентом (CMS), такими как австралийская Joomla или отечественная разработка – Битрикс. Работа в CMS позволяет минимизировать работу по созданию макета веб-приложения за счет использования шаблонов и стилей. При этом сохраняется возможность для самостоятельной разработки и подключения новых компонентов и модулей к существующему шаблону.

Использование CMS накладывает на разработчика определенные ограничения, связанные с необходимостью использовать структуру шаблонов



для построения макетов, также соответствующие выбранному шаблону стили. От шаблона зависит количество колонок, наличие эффектов в заголовке, адаптивность дизайна и пр.

Другой способ разработки интерфейса, позволяющий оптимизировать трудовые затраты разработчика – фреймворк (каркас, структура). Фреймворки – шаблоны, близкие по свойствам к специализированным библиотекам, написанные на разных языках и под разные нужды. Могут быть применены для создание одностраничных лендингов или сложных проектов. Эти шаблоны уже содержат готовые решения для создания стилей страницы, модульные сетки для построения структуры проекта, некоторые компоненты и связи между ними.

Таблица 1 – Сравнительная характеристика фреймворков

	NW.js	Django	Bootstrap	JQuery Mobile
Поддержка сервера	Node.js	Apache	Любой сервер	Поддержка сервера через Ajax
Поддерживаемые языки	JavaScript	Python	JavaScript, PHP	JavaScript
Платформы разработки	desktop- и web-приложения	web-приложения	desktop-, мобильные и web-приложения	Мобильные приложения
ОС	Windows, Linux, MacOS	Windows, Linux, MacOS	Windows, Linux, MacOS, Android	Android
Поддерживаемые СУБД	MongoDB	PostgreSQL, MySQL, SQLite	Все виды СУБД	SQLite
Поддержка web-технологий	+	+	+	+
Наличие адаптивной сетки дизайна	-	-	+	+

Еще одним методом разработки является применение библиотек JavaScript, таких как JQuery [26], ExtJS и др.

Разработка web-приложения осуществляется с использованием той или иной программной среды. Среди хорошо зарекомендовавших себя в настоящее время – последняя версия свободно распространяемой интегрированной среды разработки кроссплатформенных приложений Eclipse JavaScript Neon. Eclipse имеет собственный браузер для просмотра работы кода, но может запускать приложение также во внешние браузеры, например, Chrome или на сервере.

В качестве серверной технологии могут быть использованы бесплатные решения, такие как Open Server, Apache Tomcat или Node.js. Каждый из этих вариантов имеет свои особенности установки и развертывания, а также запуска и, что не менее важно, работает с определенной базой данных.

Сервер Node.js работает с файловой базой данных MongoDB, связь между сервером и базой данных осуществляется при помощи языка JavaScript. Node.js может взаимодействовать также с СУБД MySQL посредством подключения дополнительных модулей.

Web-сервер Open Server может быть развернут на переносном носителе (например, флеш-накопителе) и запускаться при необходимости. Open Server предоставляет возможность работы с СУБД MySQL, MongoDB, MariaDB и PostgreSQL используя набор приложений для их администрирования, таких как, например, phpMyAdmin или менеджер MySQL. Для работы с PostgreSQL используется также SQL Manager Lite for PostgreSQL.

Разработка мобильного приложения ведется на платформе Android с приоритетным использованием объектно-ориентированного языка программирования Java [10, 11, 22, 27]. Поэтому требуется наличие установленной виртуальной машины Java версии 7 и выше. Операционная система Android – система с открытым исходным кодом, основанная на Linux Kernel.

В настоящее время существует несколько сред разработки, различающихся по техническим характеристикам и функциональным

возможностям. Это Eclipse for Android Developers в пакете Eclipse Neon 2016 года (предшествующая версия – Eclipse JEE Oxygen, следующая Eclipse IDE 2019 – не включают инструменты для разработки Android приложений).

С 2015 года Google запустил Android Studio и прекратил официальную поддержку plugin для Eclipse. Таким образом, Android Studio [19] – инструмент программирования приложений, созданный специально для Android. До версии 2.3 включал в установочный файл все необходимые инструменты – Android SDK (software development kit) и Android Virtual Device (AVD). Последняя актуальная версия 3.3.2 требует отдельной загрузки и установки SDK.

Еще одной средой разработки является MIT App Inventor, доступный в первой версии и обновленной версии 2.x. Позволяет создавать приложения, используя веб-браузер и подключенное реальное устройство или его эмулятор. Данный веб-инструмент разработан в Массачусетском технологическом институте. Работа сохраняется на серверах среды, готовое приложение упаковывается в пакет .apk [12].

Таблица 2 – сравнительная характеристика сред разработки приложений на платформе Android

	Eclipse Neon for Android Developers (до 2015 года все версии Eclipse + plugin)	Android Studio IDE от IntelliJ IDEA	MIT App Inventor 2.x от Google labs
Технические характеристики			
Поддержка актуальных версий платформы	+	+	+
Интеграция с SDK	- (требуется установка)	+(до версии 2.3)	- (использует aiStarter)
Интеграция с AVD	- (требуется установка)	+	+
Работа с внешними устройствами для тестирования	+	+	+

приложений			
Поддерживаемые языки	Java, C++	Java, C++, Kotlin	Визуальный блочный язык на основе Scratch и StarLogoTNE

Таблица 1 демонстрирует возможности разных инструментов разработки мобильных приложений на платформе Android. В качестве сравнительных параметров выбраны технические особенности сред и поддерживаемые языки.

### 1.3 Обоснование выбора технологии реализации

Анализ подходящих технологий позволил выявить сильные и слабые стороны рассмотренных вариантов. И таким образом были определены подходы и технологии, максимально удовлетворяющие потребностям реализации.

В качестве технологии для интерфейса приложения были выбраны возможности фреймворка Bootstrap [7] и библиотеки JQuery. Главная особенность Bootstrap – гибкая система модульных сеток, основанная на макете из 12 колонок. Колонки могут быть вложены друг в друга, объединены, можно использовать смещение и так называемый «плавающий контейнер», являющийся основой адаптивного дизайна. Фреймворк позволяет изначально установить количество колонок для экранов разного размера. Параметры «col-md-3 col-sm-6 col-xs-12» ограничивают web-приложение по размеру экрана – md (монитор) 992 px и больше; sm (планшет) 768 px и больше; xs (смартфон) менее 768 px. Для мониторов с очень большим разрешением – 1200 px и больше – предусмотрен параметр lg.

При использовании Bootstrap разработка ориентирована в первую очередь на мобильные устройства. Тэг <meta name=«viewport» content=«width=device-width, initial-scale=1.0»> добавляет масштабируемость дизайна по ширине экрана [21].

Вся анимация и эффекты компонентов Bootstrap основаны на js-библиотеке JQuery. Одними из таких эффектов является слайдинг контента на

страницах, создание галерей и подвижных текстовых блоков, а также галерей с использованием модальных окон.

В качестве бекэнда для приложения выбран web-сервер Open Server. Его функциональные возможности (рисунок 2) в полной мере соответствуют требованиям разработки.

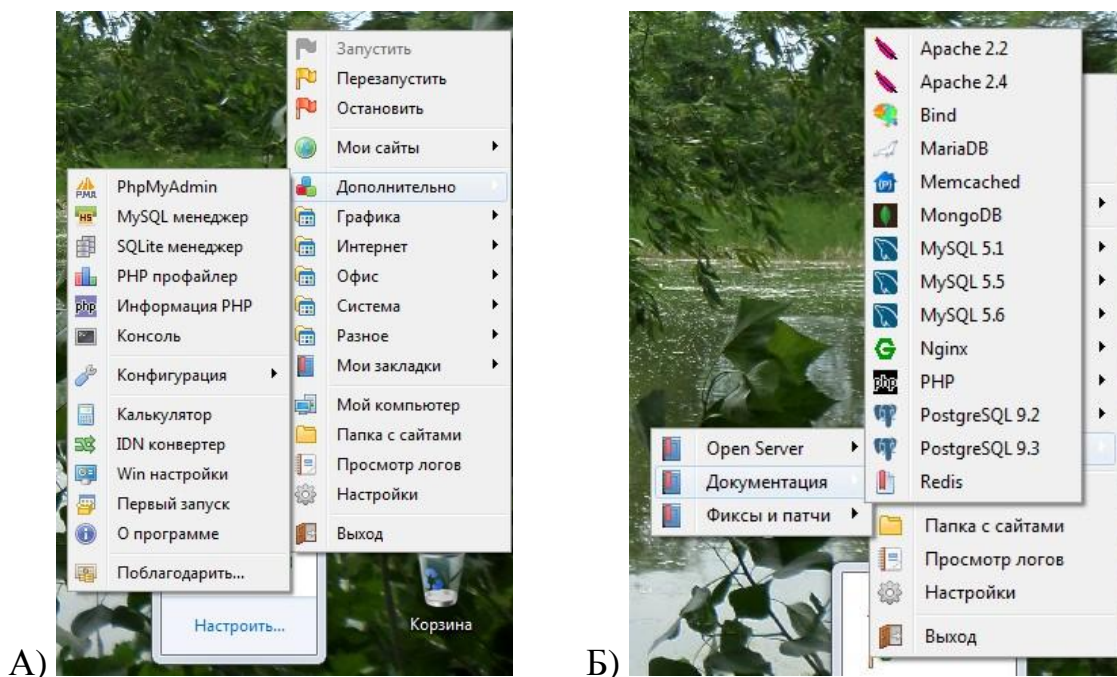


Рисунок 2 – Панель запуска Open Server: А) службы администрирования и Б) базы данных

Среди сильных сторон Open Server, прежде всего, – мобильность, возможности администрирования без использования CMS, а также работа со всеми современными базами данных – реляционными, такими как СУБД MySQL и PostgreSQL, и файловыми, как, например, MongoDB, без необходимости загрузки и установки дополнительных модулей.

В качестве СУБД выбор сделан в пользу PostgreSQL 10. Она входит в комплектацию Open Server, напрямую использует язык запросов SQL (Structure Query Language) [28] и является гибкой СУБД, поддерживающей большое количество разных типов таблиц. В качестве инструмента для работы с данной СУБД используется менеджер SQL Manager Lite for PostgreSQL – средство для регистрации базы данных на сервере PostgreSQL и создания, а также работы с таблицами базы данных.

PostgreSQL предоставляет все основные возможности реляционной базы данных, таких как работа с высокоуровневым языком запросов, целостность, отказоустойчивость и др. Кроме того, есть возможность ограничения доступа к данным.

Анализ предметной области и изучение проблемы выявили необходимость разработки программного обеспечения, обеспечивающего доступ к материалам по истории университета. Проведенный обзор технологий позволил определиться с наиболее подходящими для реализации поставленных задач методами и средами разработки.

### **Выводы по первой главе**

В качестве технологии для мобильной части приложения выбрана платформа Android, среда разработки Android Studio. Для интерфейса приложения – фреймворк Bootstrap, обеспечивающий гибкость разработки и возможность использовать JavaScript для добавления динамичности страницам; разработка с применением среды Eclipse Javascript Neon. База данных использует менеджер SQL Manager Lite for PostgreSQL для работы с таблицами и запросами в СУБД PostgreSQL. Объединение этих технологий позволяет создать полноценное программное обеспечение для онлайн-доступа к материалам монографии.

## Глава 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 2.1 Моделирование бизнес-процессов

Жизненный цикл информационной системы и программного обеспечения состоит из четырех основных этапов:

- Анализ;
- Дизайн;
- Программирование;
- Тестирование.

Этапу анализа предшествует определение «Осуществимость» проекта, куда входит выбор способа осуществления. На этапе «Анализ» проводится документирование при помощи графических методов представления потоков данных [16].

Проектирование информационной системы начинается с изучения бизнес-процессов организации. Эффективными способами исследования является технология структурного анализа и проектирования, опирающаяся на технологию IDEF. В первую очередь применяется методика функционального моделирования IDEF0 (ранее технология структурированного анализа и разработки).

В данной методике рассматривается процесс «Организация доступа к контенту» по истории ТГУ. Диаграмма состояния «Как есть» представляет процесс без применения приложений, с использованием только доступных кафедре методов продвижения информации и материалов.

Для типизации категорий информации используется четыре вида стрелок:

- Входящие (Input) – указывают то, что потребляется в ходе выполнения исследуемого процесса;
- Управляющие (Control) – указывают ограничения и инструкции, которые влияют на ход процесса;
- Выход (Output) – результат выполнения процесса;
- Исполняющий механизм (Mechanism) – исполнитель процесса и/или то, что используется для исполнения процесс [1, 6].



Рисунок 3 – Диаграмма состояния «Как есть» процесса «Организация доступа к контенту»

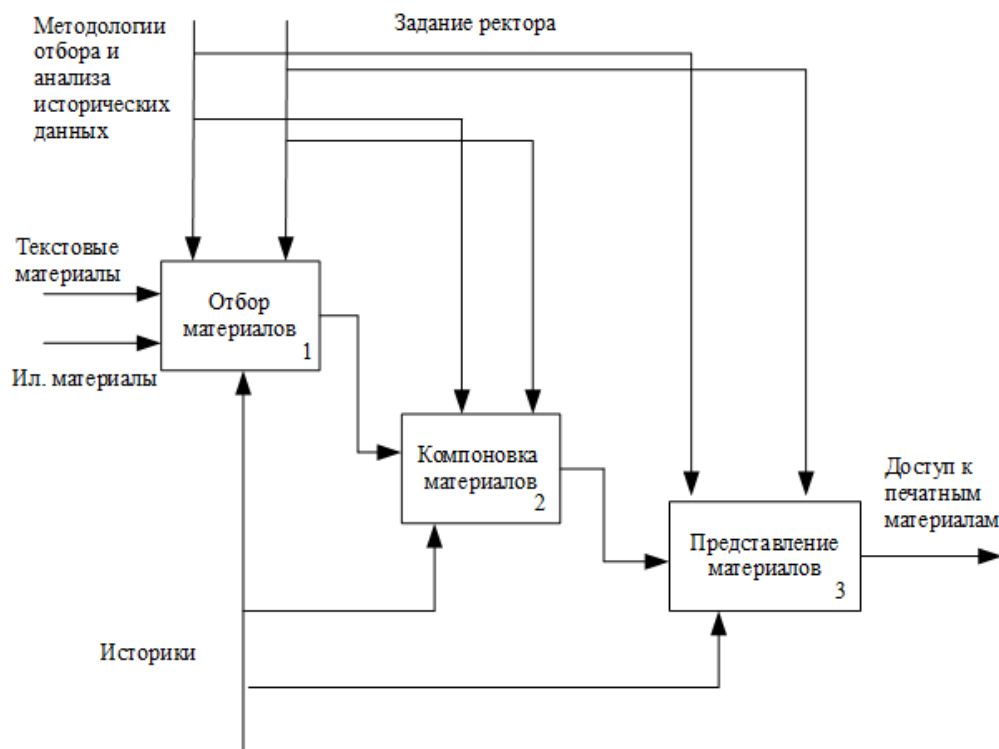


Рисунок 4 – Декомпозиция диаграммы-IDEF0, состояние «Как есть» процесса «Организация доступа к контенту»



Декомпозиция процесса разбивает основной процесс на три взаимосвязанных подпроцесса: «Отбор материалов», осуществляемый историками выбор текстовых и иллюстрационных материалов для дальнейшей обработки; «Компоновка материалов» для дальнейшего «Представления материалов» в формате печатной книги.

На входе используется прежние данные – это текстовые материалы и иллюстрации (фотографии, схемы, таблицы), ограничителями сверху служат – Методология (отбора, компоновки и предоставления исторического материала) и Распоряжение ректора, данное кафедре «Истории и философии» на заседании комиссии по плану финансово-хозяйственной деятельности (ФХД) ГумПИ; снизу – деятельность историков, задействованных в работе над книгой. На выходе – доступ к печатным данным, сформированные посредством обработанной информации.

Анализ процесса в состоянии «Как есть» выявил проблему в представлении выходных данных. Печатный вариант книги накладывает определенные ограничения на результат. Среди них основные: невозможность исправления опечаток (корректировки материала); статичность представленных данных (невозможность дополнять и обновлять информацию). В результате данные по истории ТГУ оказываются не устаревшими с течением времени с исторической точки зрения, но отстающими в плане актуальности последних исследований в данной области.

Таким образом, возникает необходимость не просто поиска оптимального способа решить данную проблему, но и сделать этот способ максимально эффективным. То есть информация может быть не только представлена в web-приложении, но также доступна на мобильном приложении. Администраторский доступ к серверу позволяет редактировать имеющуюся информацию, если/когда это потребуется, дополнять разделы новыми материалами (продолжающаяся хроника, новые факты истории, новые медиаматериалы).

Методика функционального моделирования IDEF0 позволяет взглянуть на изучаемый процесс в состоянии «Как будет». В этом случае на диаграмме будут представлены входящие потоки данных и ограничения, как в модели «Как есть», а механизм и выходные данные будут изменены в соответствии с вносимыми изменениями.

В данном случае, изменением станет использование приложений, которые решат проблему с предоставлением доступа и распространением информации по истории.

Модель «Как будет» представлена на рисунке 5.



Рисунок 5 – Диаграмма состояния «Как будет» процесса «Организация доступа к контенту»

Красные стрелки механизма исполнения и выходного сигнала отражают направления изменения в процессе. Разработка приложений ведется с сохранением существующих исполнителей процесса – при непосредственном участии историков кафедры, а за основу контента берется печатная версия издания.

В результате на последнем этапе (декомпозиция процесса представлена на рисунке 6) изменяется способ представления материалов – добавляется оцифровка, подготовка к электронной публикации текстов и таблиц.

Соответственно на выходе изменяется способ доступа к контенту. Приложения позволяют осуществлять поиск по представленным материалам, что существенно упрощает доступ к информации, организация хроник и глав издания во взаимосвязанных информационных блоках облегчает восприятие материалов разного типа.

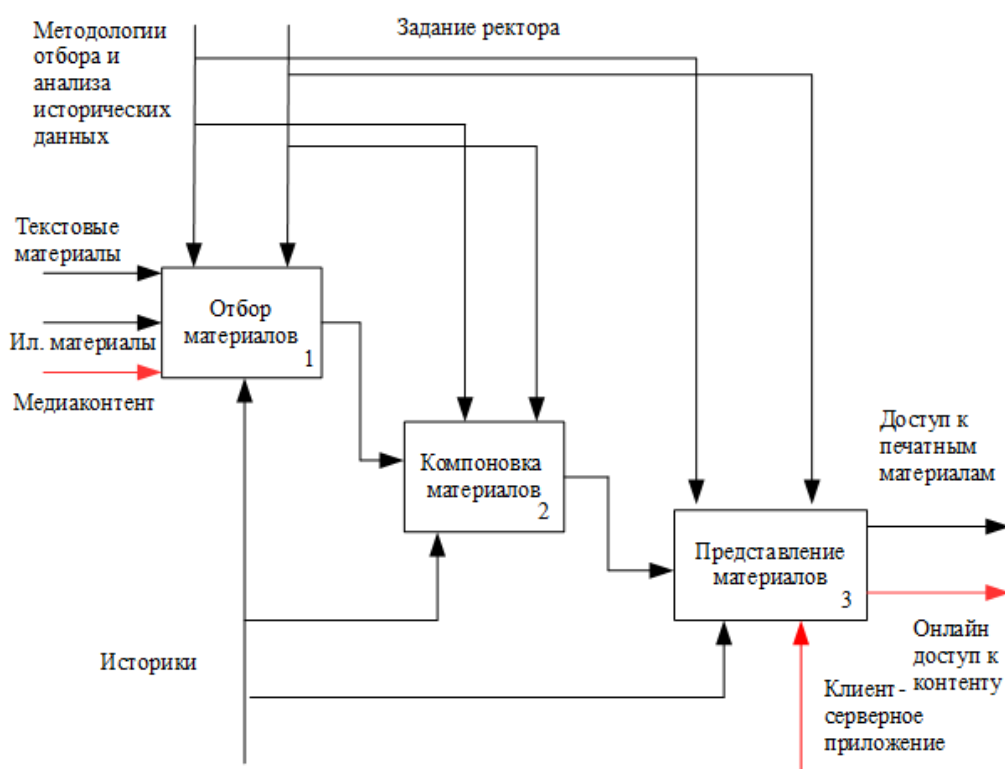


Рисунок 6 – Декомпозиция диаграммы-IDEF0, состояние «Как будет» процесса «Организация доступа к контенту»

После проведения анализа и функционального моделирования были выявлены слабые места существующего процесса продвижения информации, а именно невозможность влиять на изменения уже опубликованного, по сути, статичного, представления материала. Решением стала разработка приложений, систематизирующих собранные данные и предоставляющего возможности для их актуализации в процессе использования.

## 2.2 Моделирование технологии реализации

Этапы реализации программного обеспечения могут быть представлены в виде нескольких последовательных процессов, связанных линейно и взаимодействующих с другими элементами общей структуры разработки. Нотация BPMN (Business Process Model and Notation) позволяет наглядно проиллюстрировать весь процесс разработки в виде алгоритма (рисунок 7).

Участники, взаимодействующие в ходе разработки, это историки и разработчики. Они находятся в зоне ответственности обозначенной пулами или дорожками. Далее следуют элементы потока и соединяющие элементы. Началом, или отправной точкой череды процессов является заказ на разработку, после которого начинается первый этап – отбор материала. По сути здесь работают историки, которые компонуют информацию и определяют форму ее представления в приложении. Затем следует разработка экранных форм (дизайн-макета), основанная на техническом задании, далее утверждение представленных материалов стороной заказчика и наступает этап разработки, физической реализации. После тестирования готового программного обеспечения разработка завершается.

При разработке приложения использован реляционный тип данных. В базе данных этого типа данные организованы в виде таблиц, операции над данными – это операции над таблицами. Основными элементами реляционной модели данных являются: отношения (форма представления – таблица), кортеж (строка таблицы), сущность (описание свойств объекта), атрибут (заголовок столбца таблицы), первичный ключ (один или несколько атрибутов) [9].

Разработка базы данных начинается с создания концептуальной, логической и физической моделей, для проектирования которых используется ER-метод (от англ. entity – сущность, relationship – связь) и универсальный язык моделирования UML [3].

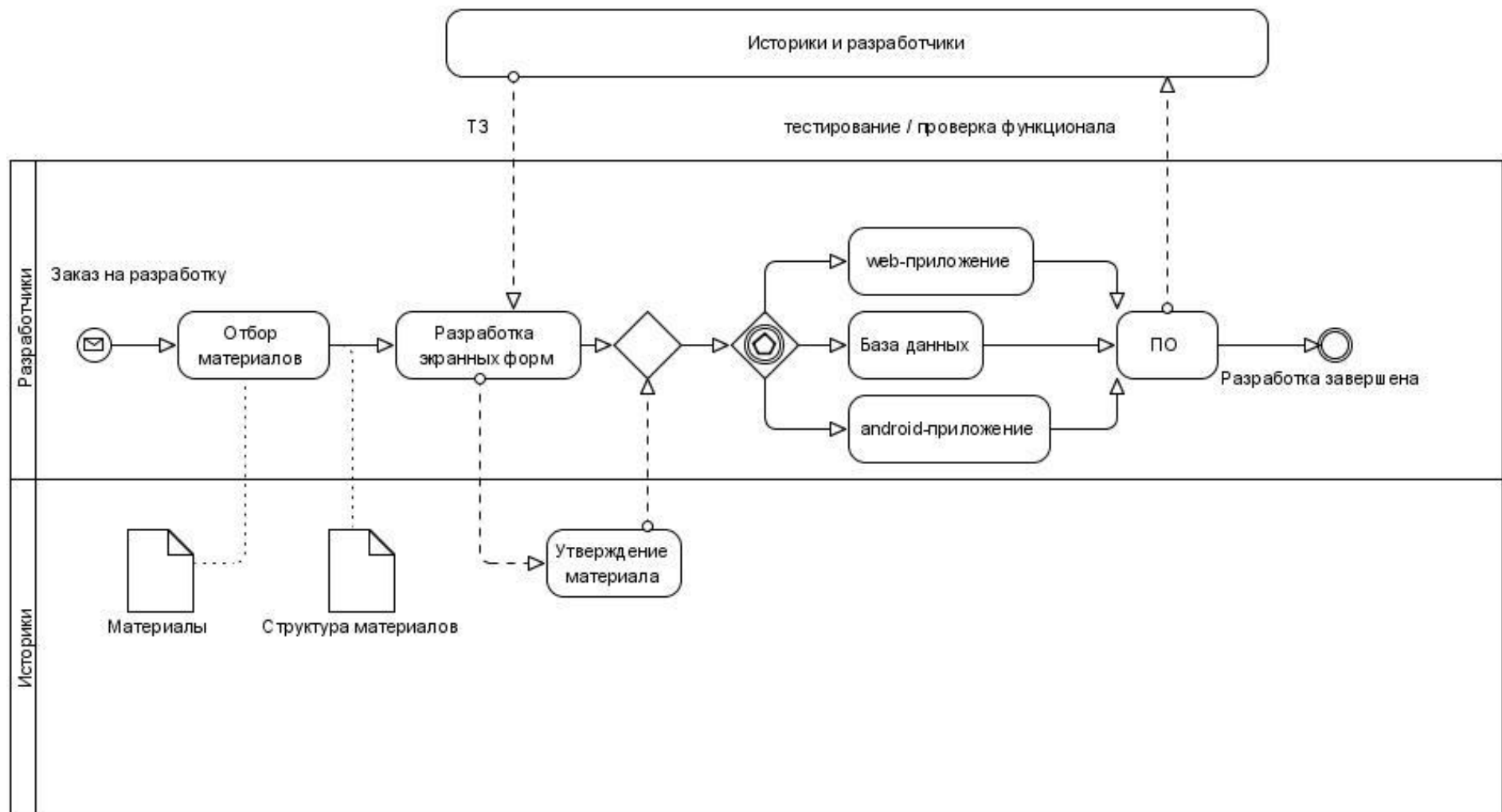


Рисунок 7 – Иллюстрация процесса разработки ПО в BPMN

Концептуальная модель базы данных содержит только понятия сущность-связь и отражает содержание таблиц. Логическое проектирование – следующий этап в работе с ER-моделью. На данном этапе таблицы, отображенные на концептуальной модели, дополняются связями.

Существует три вида связей: один-к-одному, один-ко-многим, многие-ко-многим. Первая устанавливается, когда одному атрибуту таблицы соответствует один атрибут другой таблицы. Отношения один-ко-многим указывает, что одному атрибуту соответствует несколько атрибутов другой таблицы. Многие-ко-многим – несколько атрибутов одной таблицы связаны с несколькими атрибутами другой. При этом сущности соединяются не напрямую, а посредством промежуточной (или линковочной) таблицы. Логическая модель базы данных «Хроники ТГУ» представлена на рисунке 8. На данном этапе элементы таблицы дополнены типом данных – типом значения элемента (integer, date, varchar).

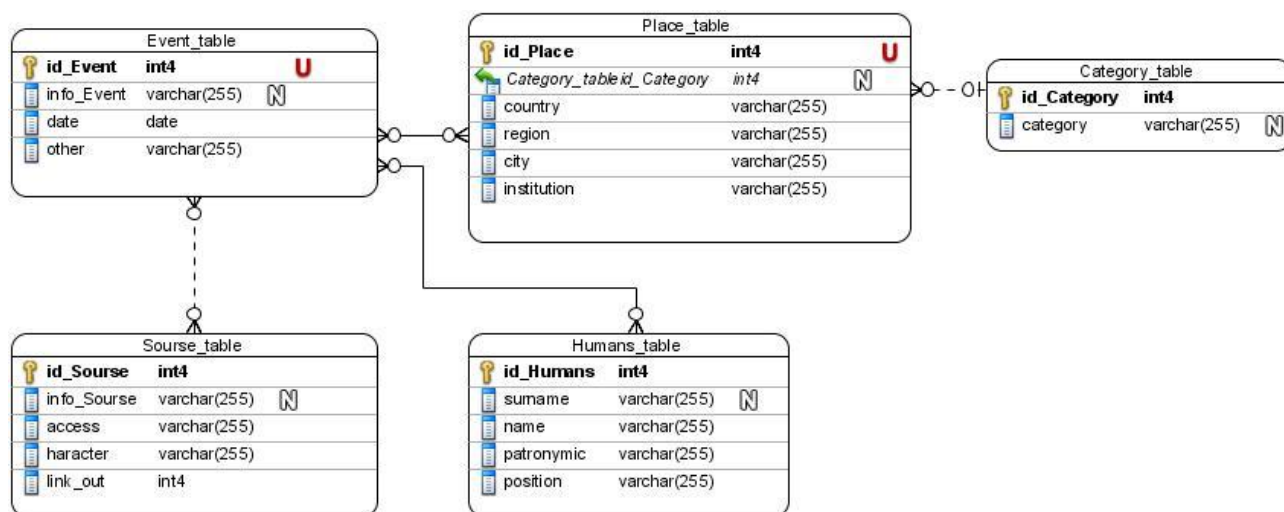


Рисунок 8 – Логическая модель базы данных «Хроники ТГУ»

Заключительный этап – построение физической модели. Добавляются линковочные таблицы, определены значения первичного и внешнего ключа. На этом этапе моделирования завершается. Следующим шагом является разработка базы данных посредством менеджера или напрямую, при помощи команд на SQL.

### 2.3 Описание технологии реализации

Фреймворк Bootstrap, на базе которого разрабатывается клиент-серверное приложение, отвечает современным требованиям адаптивного дизайна и функциональности макета.

Первоначально Bootstrap разрабатывался как внутренняя библиотека компании Twitter и до 2011 года назывался Twitter Bootstrap [7]. В настоящее время вышла уже четвертая версия фреймворка, однако наиболее стабильной и успешно совместимой с JavaScript-библиотекой JQuery по-прежнему остается Bootstrap 3. На основе JQuery работают все основные компоненты Bootstrap, такие как dropdown-меню, галереи-слайдеры и другие [20]. Именно по этой причине в скриптах и стилях соответствующие библиотеки JQuery (или ссылки на хранилище данных библиотек в сети) подключаются до .js и .css библиотек и ссылок Bootstrap. Так фреймворк запускается уже после запуска динамических компонентов.

Основным компонентом Bootstrap является система сеток, позволяющих моделировать дизайн web-приложения в соответствии с требованиями контента и при этом с учетом адаптивности. В зависимости от потребностей дизайна может быть использован стандартный «контейнер» сетки «.container», предназначенный для фиксированной ширины экрана, или «.container-fluid» – скользящий контейнер, учитывающий полную ширину экрана устройства.

В основного компонента для разработки мобильного приложения использован элемент WebView Android. В модуле, обеспечивающем доступ к клиент-серверному приложению, WebView ориентирован на полноэкранный режим.

WebView позволяет создавать свой браузер для просмотра web-страницы, с его помощью Android напрямую взаимодействует с HTML5. В общем смысле WebView отображает существующий контент как часть активности мобильного приложения, при этом сохраняя все возможности web-браузера. Позволяет перемещаться по страницам и возвращаться в приложение, а также обращаться к внутренним ссылкам и материалам приложения.

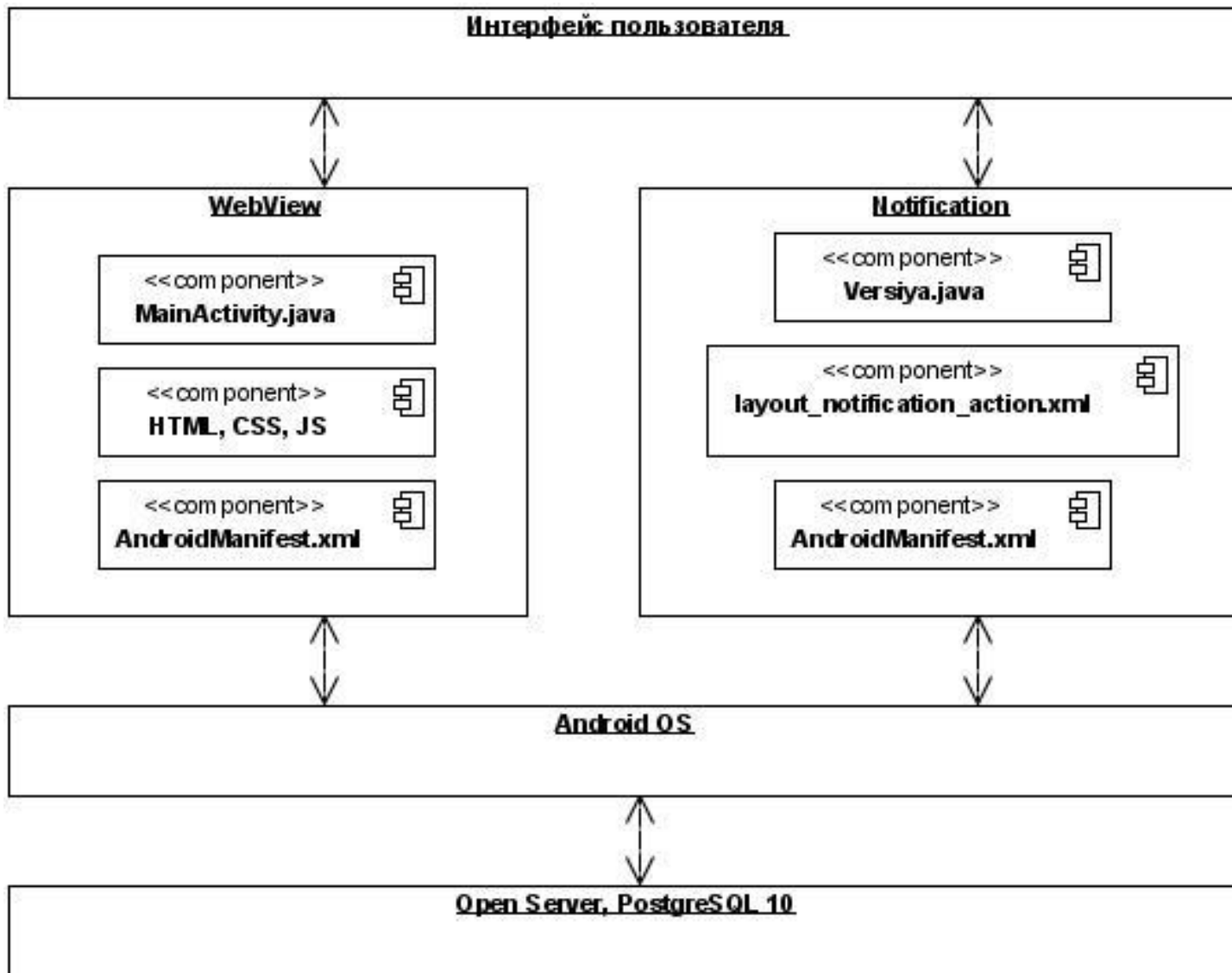


Рисунок 9 – Архитектура мобильного приложения



Данный компонент Java первоначально являлся частью платформы JavaFX 2.0 [13, 23-25], где функциональность встроенного браузера обеспечивалась при помощи WebView и WebEngine. Для работы с этим элементом в Eclipse есть plugin JavaFX Scene Builder, где можно визуальными методами установить предпочтительные размеры и параметры компонента.

WebView Android позволяет использовать различные подходы к отображению web-контента: это может быть полноэкранный режим встроенного браузера; адрес web-приложения может быть введен вручную в поисковой строке или запущен по нажатию кнопки. Компонент может обращаться к локальным ресурсам самого приложения, используя материалы внутри модуля. А также к прописанным в адресе ресурсам сервера.

На рисунке 9 представлена архитектура мобильного приложения, использующего элементы WebView и Android Notification – процесс доступа от пользовательского интерфейса к контенту.

В целом, данные технологии позволяют создать полноценное клиент-серверное приложение, объединяющее в себе как особенности мобильного приложения, так и web-технологии.

### **Выводы по второй главе**

Было проведено моделирование бизнес-процесса организации доступа к контенту и процесса разработки программного обеспечения. Отдельно представлена архитектура мобильного приложения.

Использование разных методов моделирования позволяет подробно рассмотреть задачу с разных ракурсов и определиться с подходами физической реализации.

# Глава 3 ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 3.1 Разработка интерфейса

В качестве среды разработки программного обеспечения было выбрано свободно распространяемое приложение Eclipse Javascript Neon, который позволяет создавать рабочий пакет, включающий все необходимые JavaScript- и HTML5-библиотеки для создания интерфейса.

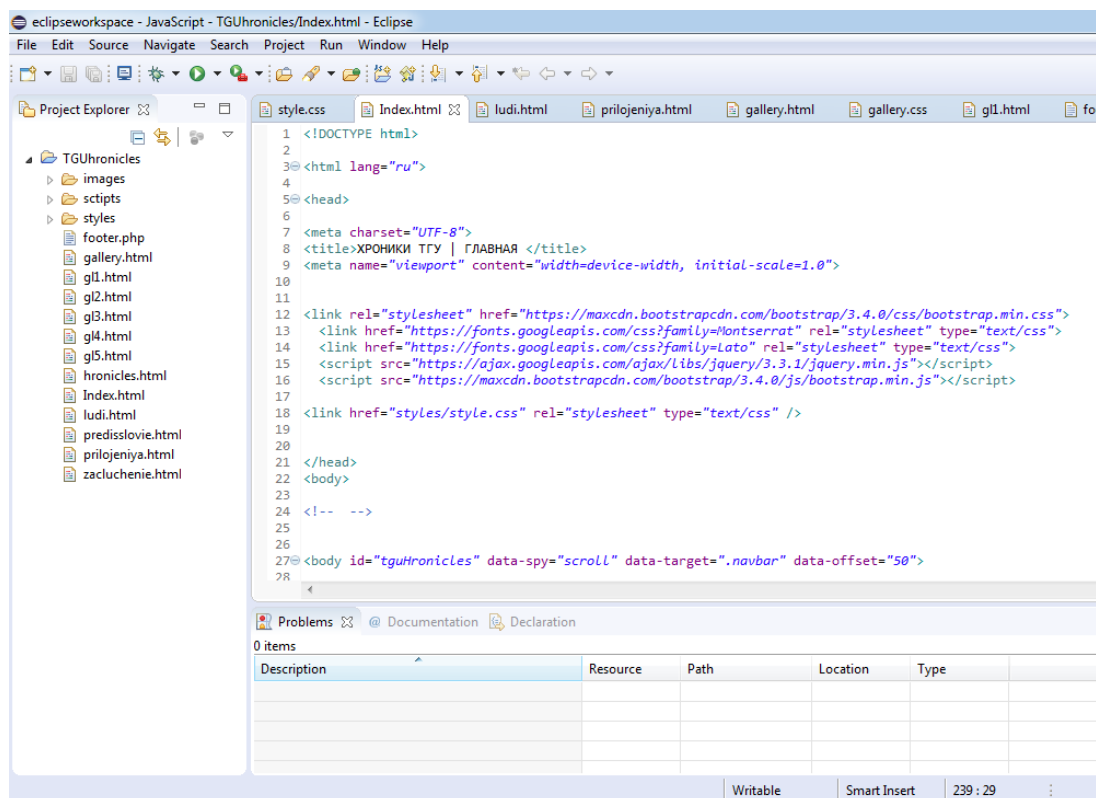


Рисунок 10 – Интерфейс и рабочее пространство среды Eclipse JavaScript Neon

Элементы фреймворка Bootstrap встраиваются в основной HTML5 код при помощи тегов <id> и <class>, которые создают контейнер и колонки модальной сетки:

```
<div id=«reading» class=«container-fluid»>  
<div class=«col-xs-8»> <div class=«row»>  
<div class=«col-md-3 col-sm-6 col-xs-12»>  
<div class=«panel panel-default text-center»>  
<div class=«panel-heading»>Предисловие</div>
```

```

<div class=«panel-body»>
<a class=«read» href=«predislovie.html» target=«_blank» style=«text-decoration: none; color: black»>Ректор Тольяттинского государственного университета
<br/> М.М. Криштал</a> </div></div></div>

```

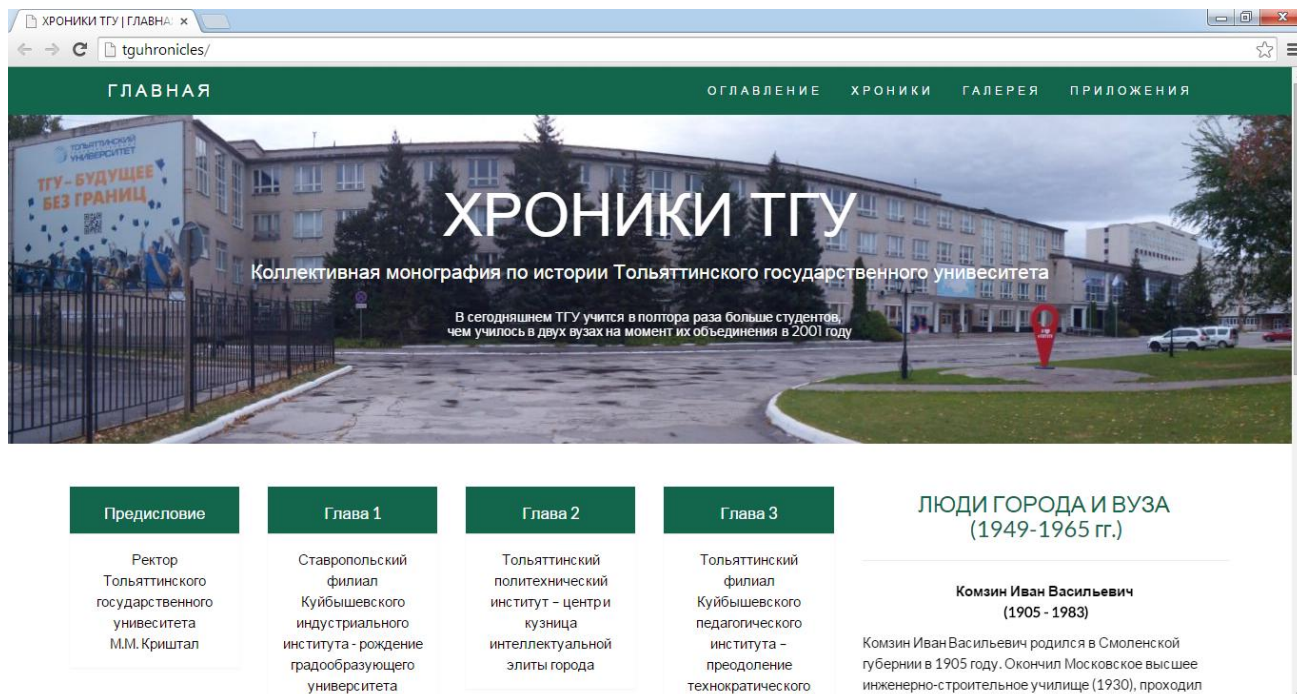


Рисунок 11 – Главная страница интерфейса приложения с примером реализации модульной сетки (главы и колонка «Люди города и ВУЗа»)

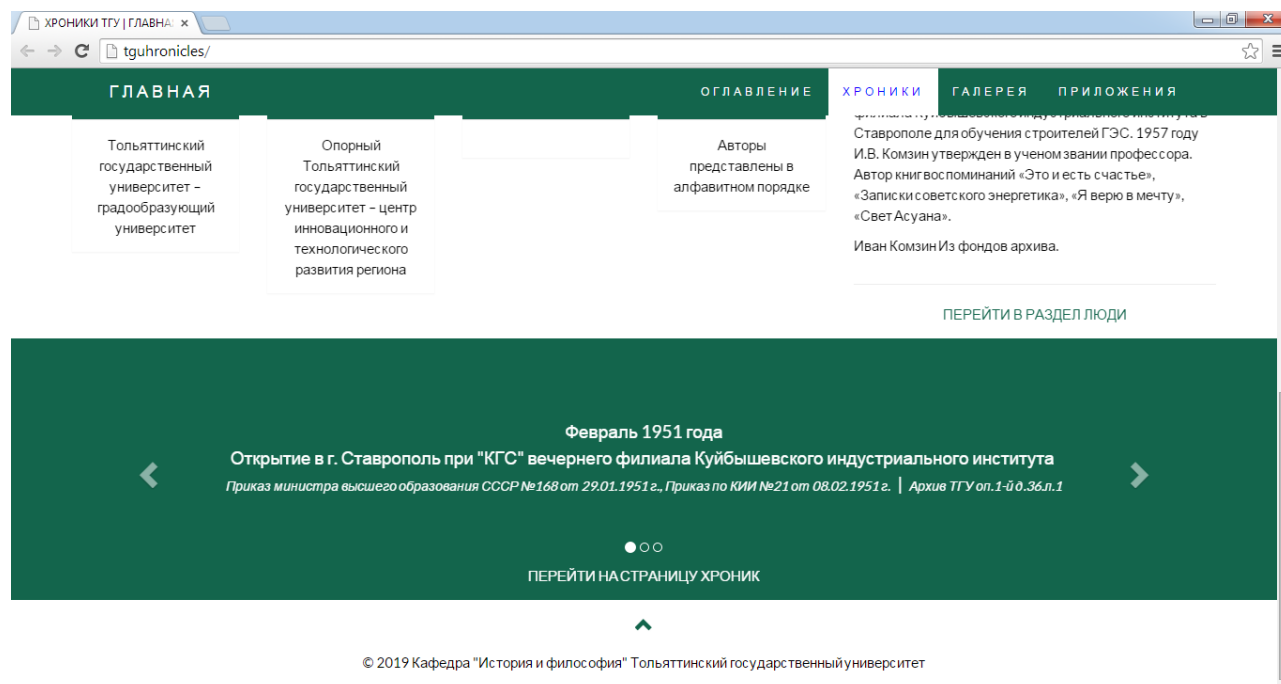


Рисунок 12 – Пример интерфейса приложения с реализацией мягкого скроллинга, фиксация по Id в панели навигации

На рисунке 12 представлена демонстрация возможностей Bootstrap – прокрутка по странице с привязкой разделов к элементам навигации в шапке фиксированного меню. Программная реализация данной функции на JavaScript:

```
body id=«tguHronicles» data-spy=«scroll» data-target=«.navbar» data-offset=«50»>
$(document).ready(function(){ $(«.navbar a, footer a[href='#tguHronicles']
»)on('click', function(event) { if (this.hash !== «») {event.preventDefault();
var hash = this.hash; $('html, body').animate({scrollTop: $(hash).offset().top}, 900,
function(){ window.location.hash = hash; }); }});
```

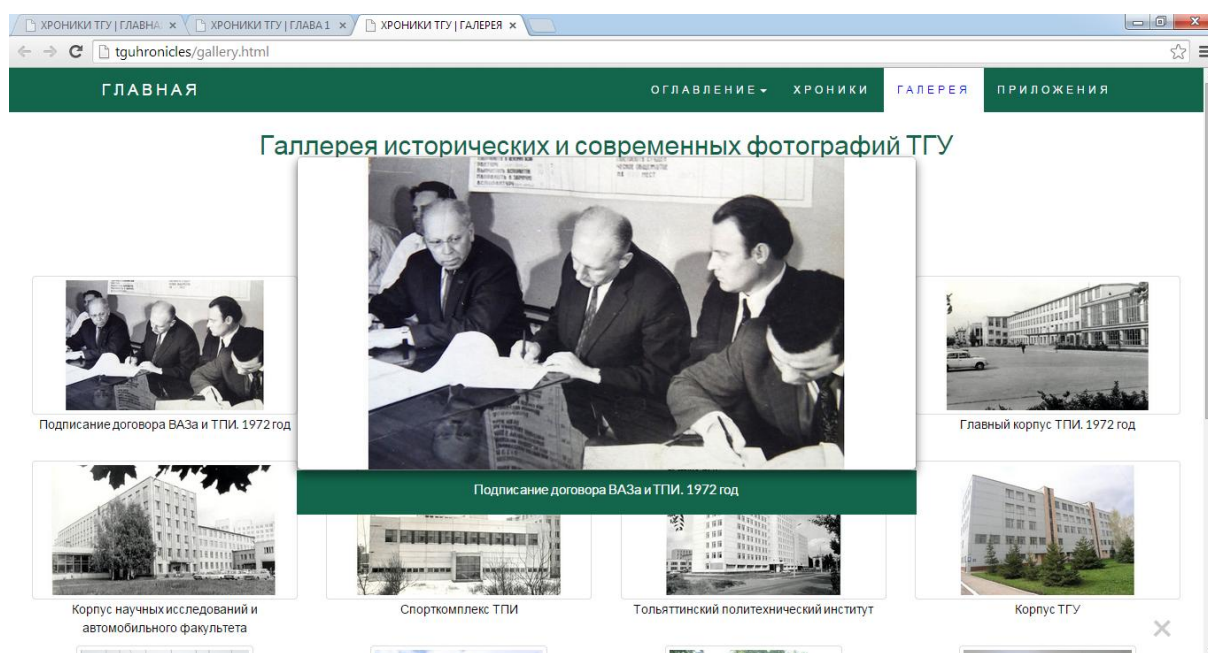


Рисунок 13 – Пример интерфейса веб-приложения с реализацией галереи Bootstrap и модального окна Bootstrap

В приложении реализована возможность использования модальных окон для создания окна просмотра изображения в галерее. Захват картинки и динамическая часть добавлены при помощи JavaScript:

```
var modal1 = document.getElementById('okno1');
var img1 = document.getElementById('img1');
var modalImg1 = document.getElementById(«1»);
var captionText1 = document.getElementById(«caption»);
img1.onclick = function(){ modal1.style.display = «block»;
modalImg1.src = this.src; captionText1.innerHTML = this.alt; }
var span = document.getElementsByClassName(«close»)[0];
```

```
span.onclick = function() { modal1.style.display = «none»;} }
```

Повторяющиеся элементы интерфейса добавлены при помощи `php include`.

### 3.2 Разработка базы данных

Регистрация базы данных в СУБД PostgreSQL осуществляется при помощи SQL Manager Lite for PostgreSQL. Генерация таблиц при помощи SQL происходит посредством запросов (рисунок 14):

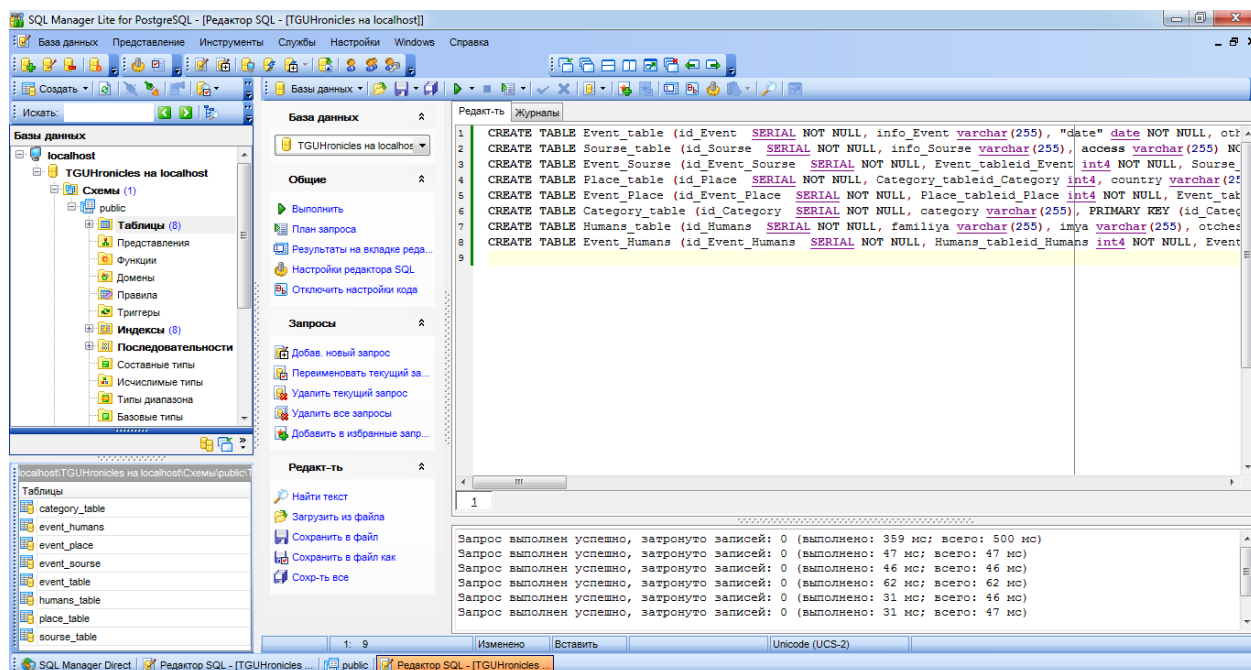


Рисунок 14 – Создание таблиц в базе данных «Хроники ТГУ» в интерфейсе SQL Manager Lite for PostgreSQL

Основные таблицы «Событие», «Источник», «Место», «Люди», «Категория» создаются с указанием уникального первичного ключа таблицы и внешних ключей от линковочных таблиц. Генерация происходит в соответствии с Физической моделью базы данных (рисунок 15). Для таблицы «Событие» первичный ключ указывается как: PRIMARY KEY (id\_Source). Линковочные таблицы создаются с указанием внешних ключей смежных таблицы. Пример создания промежуточной таблицы при помощи запроса SQL: CREATE TABLE Event\_Place (id\_Event\_Place SERIAL NOT NULL, Place\_tableid\_Place int4 NOT NULL, Event\_tableid\_Event int4, PRIMARY KEY (id\_Event\_Place)).

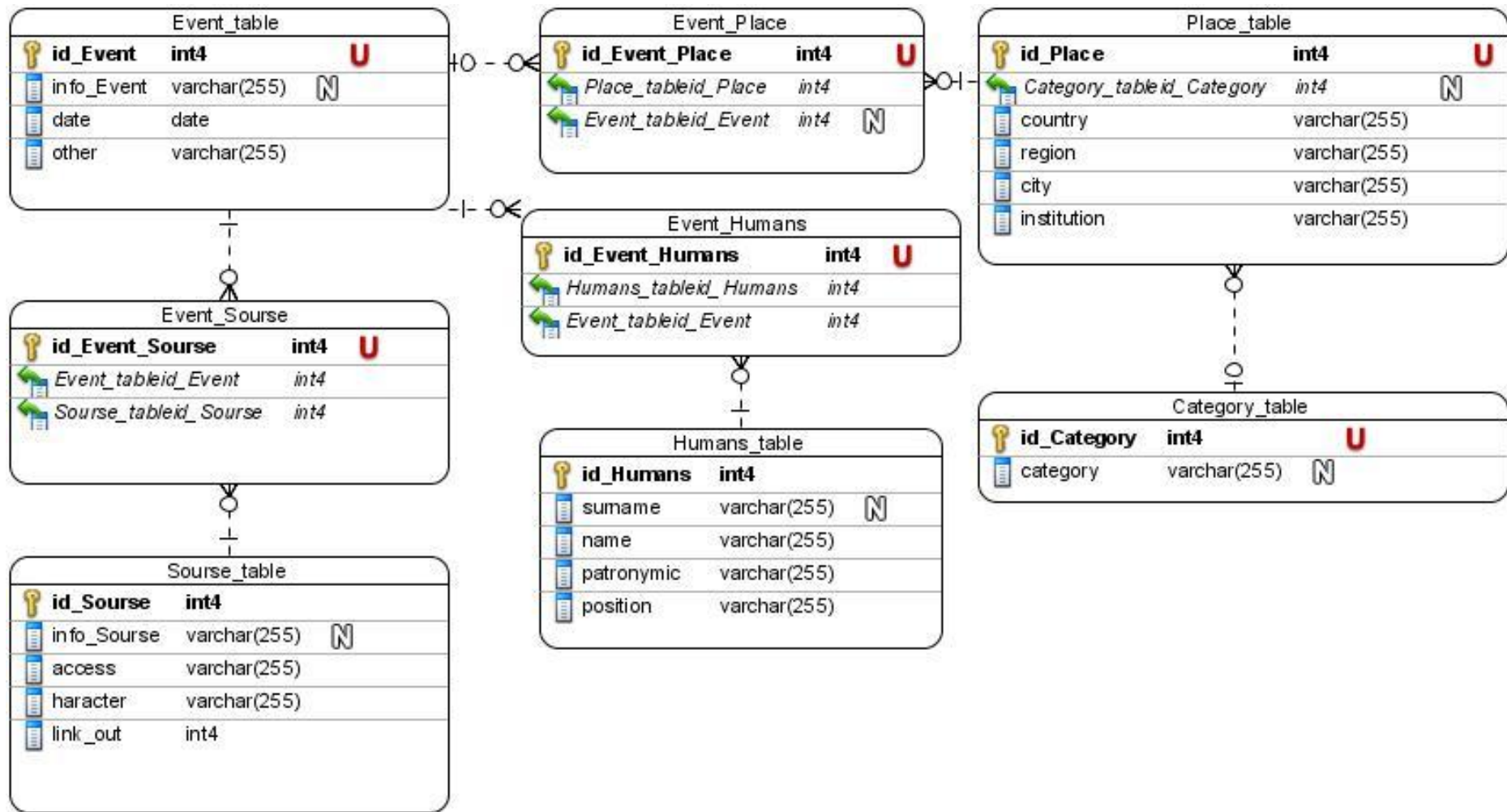


Рисунок 15 – Физическая модель базы данных «Хроники ТГУ»

После индексации таблиц связи устанавливаются посредством первичного ключа и внешнего ключа.

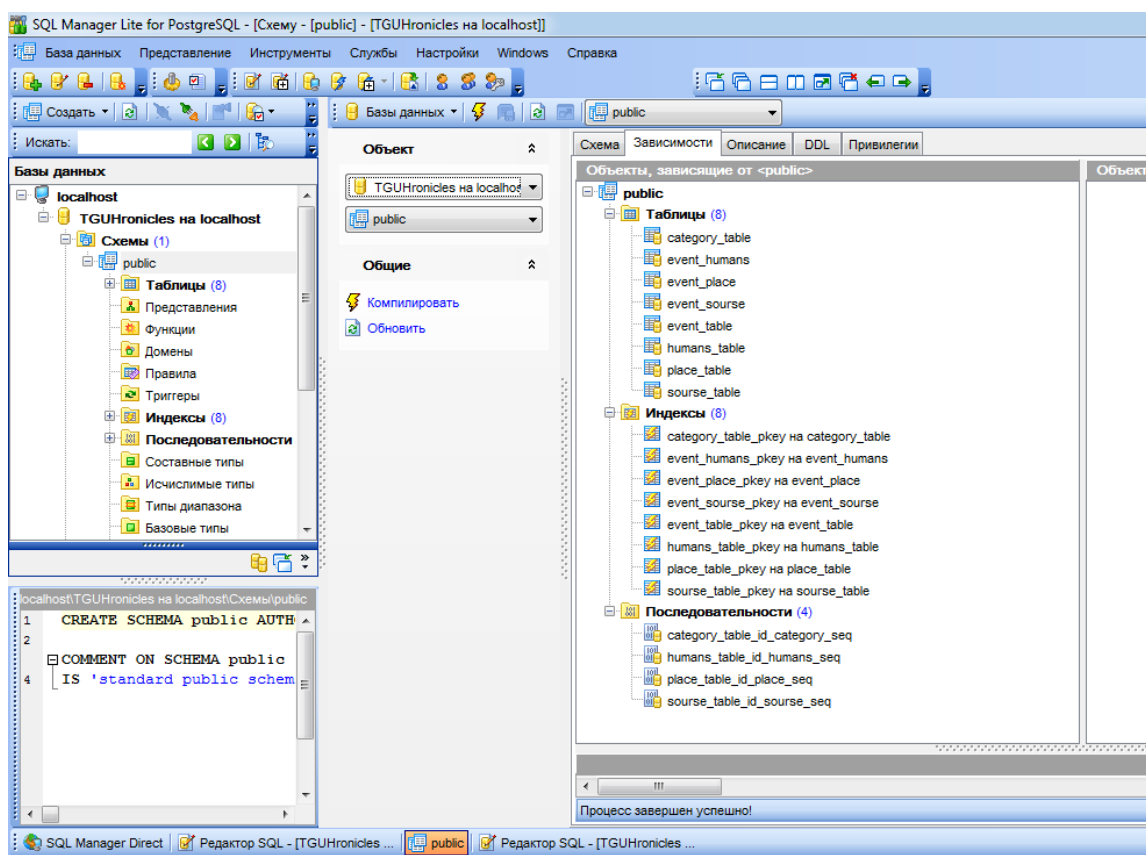


Рисунок 16 – Схема базы данных «Хроники ТГУ» в интерфейсе SQL Manager Lite for PostgreSQL

Наполнение базы данных посредством SQL запросов:

INSERT INTO event\_table (info\_event, date, other) VALUES ('Открытие в г. Ставрополь-на-Волге при «Куйбышевгидрострое» вечернего филиала Куйбышевского индустриального института им. В.В. Куйбышева (КИИ) с факультетами: механический, строительный и энергетический','8/02/1951','не позже');

INSERT INTO Source\_table (info\_Source, access) VALUES ('Приказ министра высшего образования СССР №168 от 29.01.1951 г., Приказ по КИИ №21 от 08.02.1951 г.','Архив ТГУ оп.1-й д.36.л.1');

INSERT INTO Category\_table (category) VALUES ('Тольяттинский государственный университет');

В результате создается возможность создания перекрестного запроса, с обращением к данным из нескольких таблиц: SELECT id\_event, info\_event, date

, info\_sourse FROM event\_table, sourse\_table WHERE event\_table.id\_event = sourse\_table.id\_sourse; где к информации по полям одной таблицы добавляется информация из одного поля другой, с пересечением по первичному ключу. Результат выполнения запроса представлен на рисунке 17.

id	info_event	date	info_source
1	Открытие в г. Ставрополь-на-Волге при «Куйбышевгидрострое» вечернего филиала Куйбышевского инд	08.02.1951	Приказ министра высшего образования С
2	Официальное сообщение об организации филиала Куйбышевского индустриального института в г. Став	16.04.1951	"Гидростроитель", №2 от 16.04.1951г.
3	Для поступления в филиал были организованы пятимесячные подготовительные курсы без отрыва от п	01.04.1951	Документы по истории ТПИ-ТГУ
4	Исполняющим обязанности заведующего вечернего филиала назначен Д.Е. Чуркин	16.08.1951	Приказ КИИ от 16.08.1951 г.
5	В филиал поступили 162 человека. Первым штатным преподавателем назначен А.Э. Лившиц.	01.01.1952	Документы по истории ТПИ-ТГУ
6	Заведующим вечерним филиалом назначен старший преподаватель кафедры графики КИИ В.К. Никола	26.08.1952	Документы по истории ТПИ-ТГУ
7	В филиал поступили 107 человек	01.01.1953	Приказ КИИ №181 от 26.08.1952г.
8	В источнике ВУЗ впервые упомянут как филиал вечернего отделения КИИ (заведующий - кандидат техн	28.09.1953	Документы по истории ТПИ-ТГУ

Рисунок 17 – Результат выполнения перекрестного SQL-запроса в интерфейсе SQL Manager Lite for PostgreSQL

Следующим этапом работы с базой данных является установление ее связи с приложением на Open Server при помощи php запроса:

```
<? Php
$db = pg_connect(«host=localhost port=5432 user=postgres
dbname=TGUHronicles»); $connectstring = $db; $connect =
pg_connect($connectstring) or die('Не удалось соединиться: ' . pg_last_error()); ?>
```

Для подключения использованы регистрационные параметры базы данных TGUHronicles на PostgreSQL Server 10.

### 3.3 Разработка мобильного приложения

Разработка программного обеспечения для доступа к клиент-серверному приложению на мобильном устройстве проводилась в среде Android Studio. Описание функций элемента WebView размещается в файле «MainActivity.java». Здесь же задаются компоненты активности – такие как загрузка URL, возврат назад постранично в приложении, чтение скрипта. Так же могут быть добавлены переходы между модулями и доступ к внутреннему контенту. Фрагмент кода представлен ниже:



```

package com.example.test;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;
public class MainActivity extends AppCompatActivity {
public WebView webView;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
webView = (WebView) findViewById(R.id.webView);
webView.getSettings().setJavaScriptEnabled(true);
webView.getSettings().setBuiltInZoomControls(true);
webView.getSettings().setDisplayZoomControls(false);
webView.setWebViewClient(new WebViewClient());
webView.loadUrl(«http://192.168.43.42»);  }
@Override
public void onBackPressed (){
if (webView.canGoBack()){
webView.goBack();
} else {super.onBackPressed();}}

```

Параметры компонента могут быть заданы Eclipse JavaScript Neon в файле «activity\_main.xml» или графически (рисунок 18):

```

<WebView
android:id=«@+id/webView»
android:layout_width=«match_parent»
android:layout_height=«match_parent»/>

```

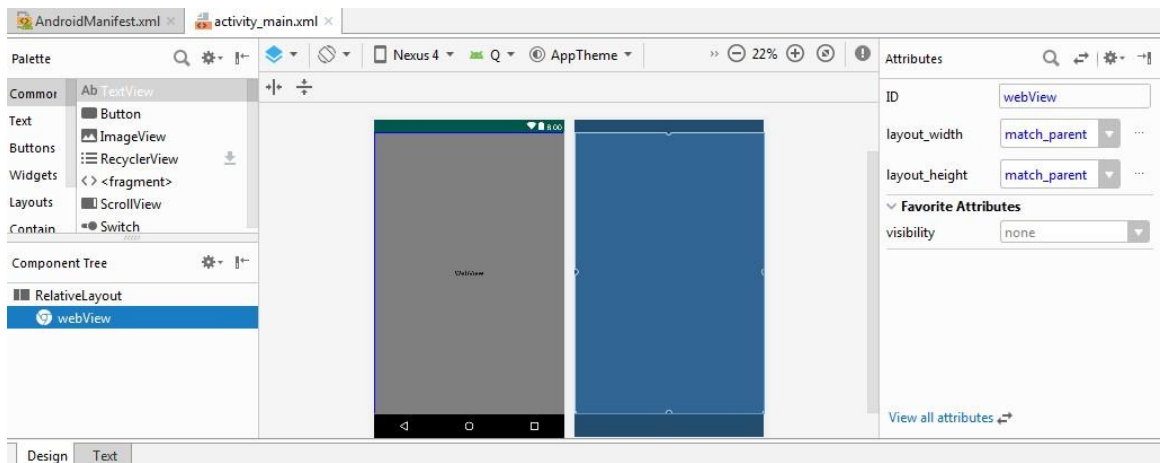


Рисунок 18 – Элемент WebView Android в графическом представлении среды разработки Android Studio.

В файле «AndroidManifest.xml» задаются допустимые разрешения приложения. Так как данный элемент использует доступ в сеть Интернет, то это отражается в файле манифеста в виде разрешения: `<uses-permission android:name=«android.permission.INTERNET»/>`. Здесь же указывается название приложения и загружается ссылка на логотип: `android:icon=«@mipmap/tgu_logo»`  
`android:label=«Хроники ТГУ»`

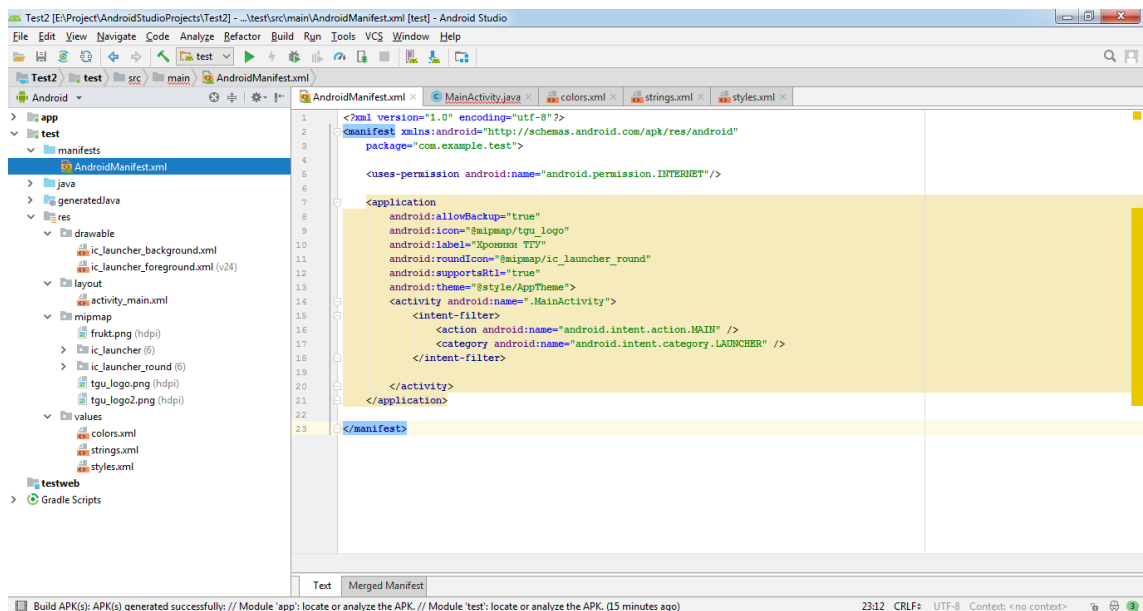


Рисунок 19 – Файл «AndroidManifest.xml» в среде Android Studio.

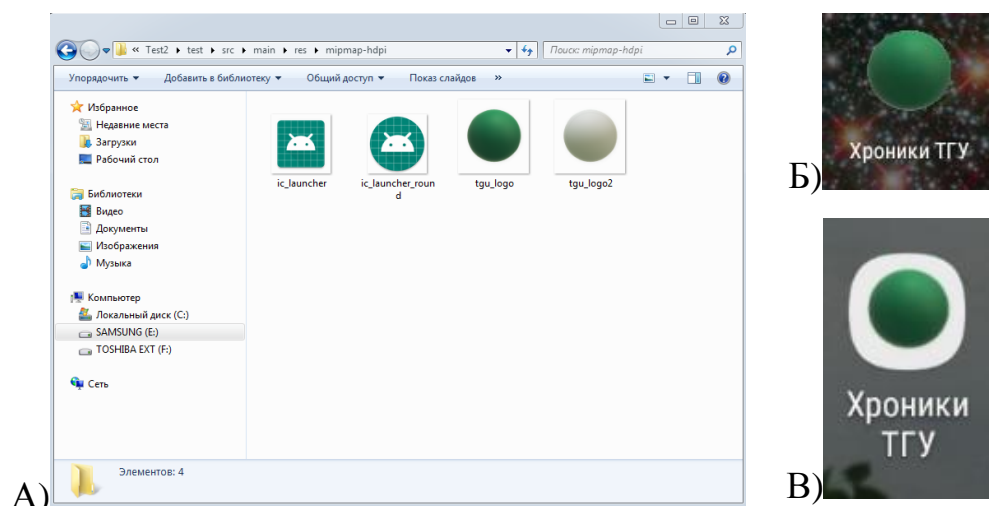


Рисунок 20 – Варианты логотипа: А) в папке mirpar проекта и Б), В) их отображение на устройствах

Стили, относящиеся к приложению в целом, записываются в файлы, размещенные в папке «values». В частности, скрипт файла «color.xml» позволяет установить основные цвета приложения, а также убрать отображение строки заголовка (тулбар), добавив параметр `<style name=«AppTheme» parent=«Theme.AppCompat.NoActionBar»>` и/или изменить цвет строки состояния.

Тестирование программного обеспечения проводилось на трех мобильных устройствах: планшете Digma Optima 10.4, смартфонах Lenovo A859 и Samsung Galaxy J4. Это позволило сравнить результаты работы элемента WebView Android и работу компонентов в операционных системах Android 5.1, 4.2.2 и 8.0.0 соответственно.

Ниже представлены скриншоты, сделанные на разных устройствах и иллюстрирующие работу приложения. Рисунок 21 демонстрирует процесс загрузки приложения с оповещением об успешной установке. На рисунке 22 изображен интерфейс главной страницы в горизонтальном положении устройства. Размещение объектов интерфейса соответствует структуре модульной сетки с двойным контейнером: общим для раздела и внутренним для глав и раздела «Люди города и ВУЗа».

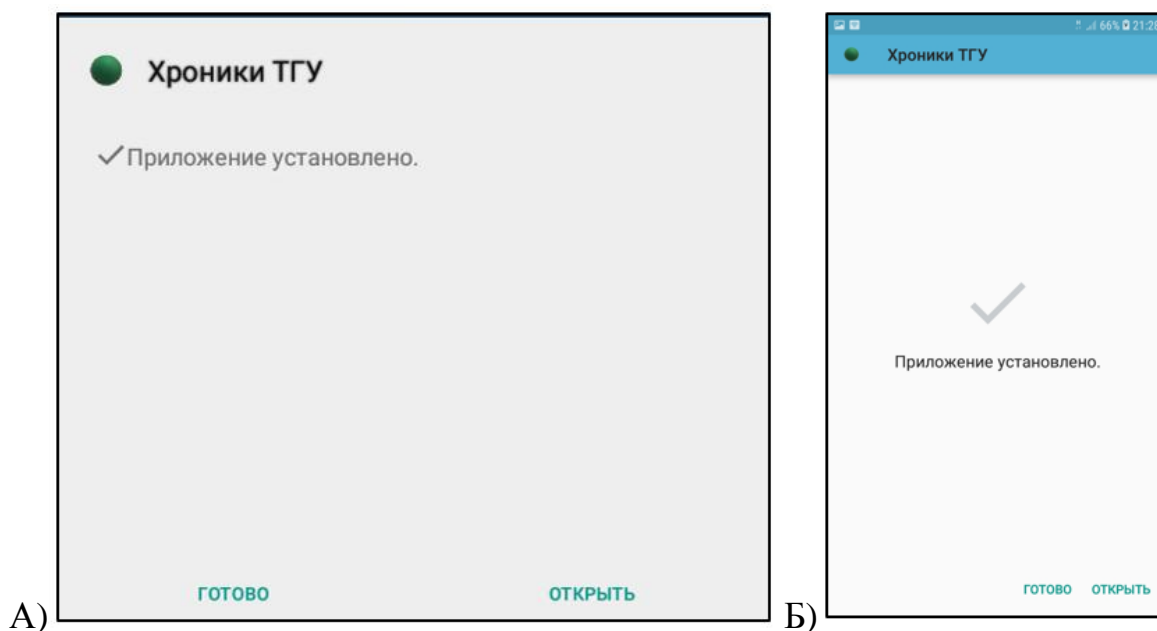


Рисунок 21 – Установка приложения на мобильных устройствах двух типов: А) планшет Digma Optima 10.4 и Б) смартфон Sumsung Galaxy J4.

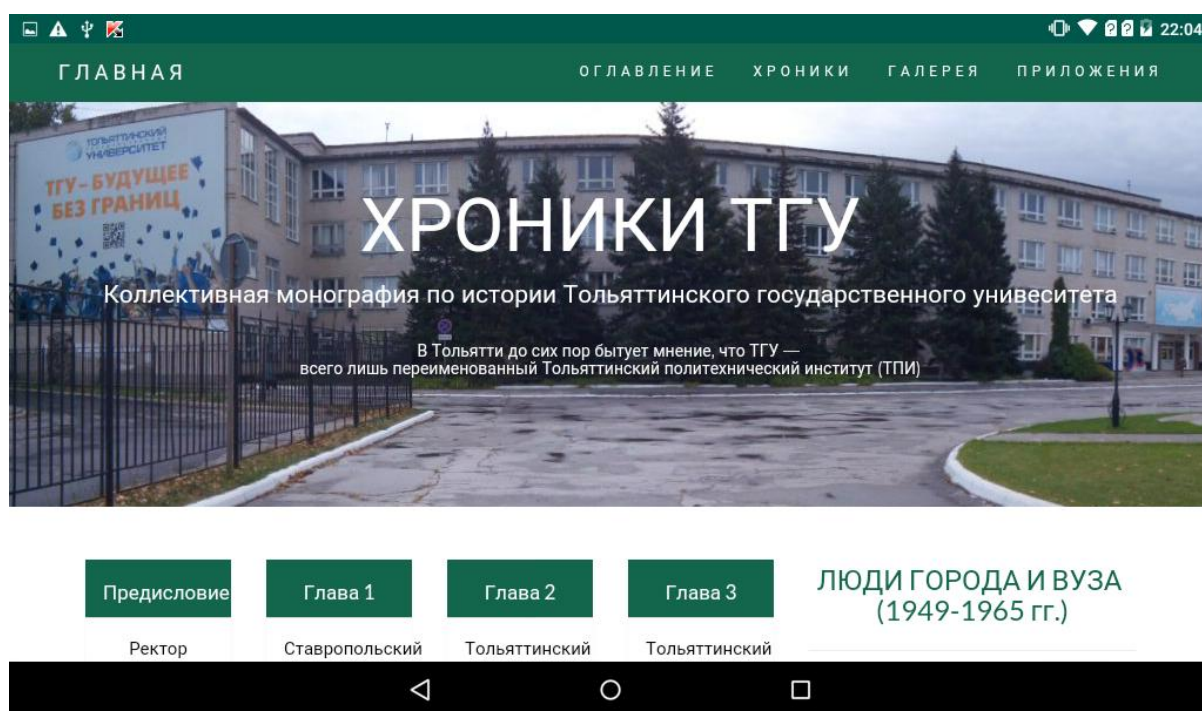


Рисунок 22 – Фрагмент интерфейса – отображение главной страницы веб-приложения на планшете Digma Optima 10.4.

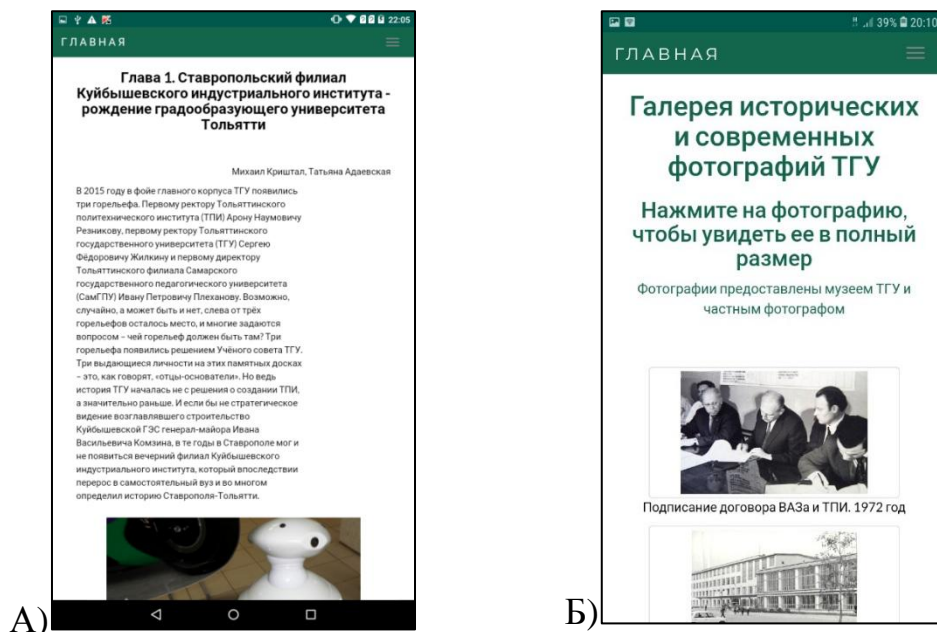


Рисунок 23 – Фрагменты интерфейса – отображение страниц веб-приложения через мобильное приложение на устройствах: А) планшет Digma Optima 10.4 и Б) смартфон Sumsung Galaxy J4

Для создания уведомлений о наличии обновлений или подключении новых разделов были изучены такие виды оповещений как Android Notification и Broadcast Receiver, создающий широкоформатные сообщение типа alarm. Для разрабатываемого приложения был выбран первый тип уведомлений private NotificationCompat.Builder tester, так он позволяет пользователю свободно работать с приложением, не отвлекая его внимание на перекрывающие экран или регулярно повторяющиеся сообщения, а создает значок в строке состояния .setSmallIcon(R.mipmap.ic\_launcher).

Код для создания уведомления и перехода на новую активность Versiya.class имеет следующий вид:

```

this.tester.setWhen(System.currentTimeMillis());
this.tester.setTitle(«ХРОНИКИ ТГУ»);
this.tester.setText(«Проверить обновления и доступные разделы»);
Intent intent = new Intent(this, Versiya.class);
PendingIntent pendingIntent = PendingIntent.getActivity(this, REQUEST_CODE,
intent, PendingIntent.FLAG_UPDATE_CURRENT);
this.tester.setContentIntent(pendingIntent);

```

```
NotificationManager notificationService =  
(NotificationManager)this.getSystemService(Context.NOTIFICATION_SERVICE);  
Notification notification = tester.build();  
notification.defaults = Notification.DEFAULT_VIBRATE;  
notificationService.notify(NOTIF_ID, notification);
```

Результат работы программы представлен на рисунке 24.

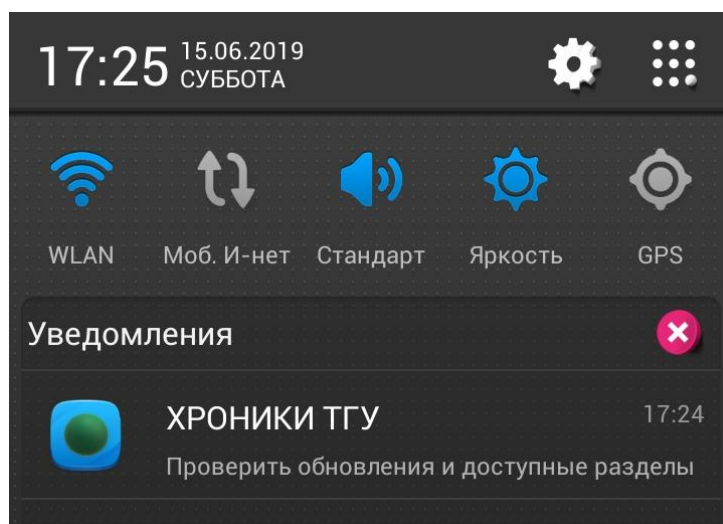


Рисунок 24 – Push-уведомление на экране телефона Lenovo A859

Для привлечения внимания к появлению уведомления в области строки состояния может быть использован звуковой сигнал – определенный по умолчанию или выбранный из установленных на устройстве, либо вибрация. Данный элемент требует внесения соответствующего разрешения в файл манифеста, для уведомления с сигналом в виде вибрации это будет: `<uses-permission android:name=«android.permission.VIBRATE»/>`.

### **Выводы по третьей главе**

Представлена технология реализации всех компонентов разрабатываемого программного обеспечения – мобильного- и web-приложений и базы данных.

В ходе тестирования проверялась корректность загрузки, установки и запуска программного обеспечения; отображение клиент-серверного приложения – интерфейс и контент – с учетом возможностей автоповорота вертикально и горизонтально, масштабирование элементов; работа переходов при помощи средств устройства, доставка корректная работа уведомлений.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы по теме «Разработка программного обеспечения для клиент-серверного мобильного приложения и web-приложения по истории ТГУ» изучались практические возможности реализации онлайн-доступа к монографии по истории ТГУ, подготовленной кафедрой «История и философия».

В ходе выполнения работы было разработано программное обеспечение, обеспечивающее доступ к материалам, хранящимся на сервере, и внешним источникам данных. В качестве среды разработки использованы Eclipse JavaScript Neon и Android Studio.

Был решен ряд задач:

1. Проведен анализ предметной области и моделирование бизнес-процесса, определена технология реализации;
2. Разработано клиент-серверное приложение с использованием фреймворка Bootstrap;
3. Разработана база данных с использованием СУБД PostgreSQL Server 10;
4. Разработано android-приложение с использованием элементов WebView и Android Notification;
5. Проведено тестирование приложения.

Дальнейшим этапом работы над приложением является наполнение разделов материалами и ведение базы данных хроник.

Разработанное приложение отвечает всем предъявляемым заказчиком требованиям – содержит материалы разного характера (текстовые, медиаматериалы, ссылки на внешние источники), имеет ресурсы для расширения возможностей применения, таких как отображение видеоконтента. Приложение может быть установлено на мобильное устройство любого типа с одинаковым качеством отображения и доступа к контенту.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Александров, Д. В. Инструментальные средства информационного менеджмента : CASE-технологии и распределенные информационные системы [Текст] : учебное пособие / Д. В. Александров. – М. : Финансы и статистика, 2011 – 224 с.
2. Баратов, И. В. Большой англо-русский и русско-английский компьютерный словарь [Текст] / И. В. Баратов, ред. Н. В. Морозов. – М. : Живой язык, 2010. – 512 с.
3. Буч, Г. Введение в UML от создателей языка. 2-е изд. [Текст] / Г. Буч, Д. Рамбо, И. Якобсон. – М. : ДМК Пресс, 2015. – 496 с.
4. Глухов, И. Н. Теория систем и системный анализ [Текст] : учебное пособие / И. Н. Глухов – М. : Проспект, 2017 - 117 с.
5. Голицына, О. Л. Информационные системы [Текст] : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 2-е изд. – М. : ФОРУМ ИНФРА-М, 2016. – 448 с.
6. Горбенко, А. О. Информационные системы в экономике [Текст] / А. О. Горбенко. – М. : БИНОМ. Лаборатория знаний, 2014. – 292 с.
7. Евдокимов, А. П. Создание сайтов своими руками на Bootstrap [Текст] / А. П. Евдокимов, М. В. Финков. – СПб. : Наука и Техника, 2017. – 240 с.
8. Кириченко, А. В. Динамические сайты на HTML, CSS, JavaScript и Bootstrap : Практика, практика и только практика [Текст] / А. В. Кириченко, Е. В. Дубовик. – СПб. : Наука и Техника, 2018. – 272 с.
9. Кумскова, И. А. Базы данных [Текст] : учебник / И. А. Кумскова. – 2-е изд., стер. – М. : КНОРУС», 2015. – 488 с.
10. Лигуори, Р. Java 8. Карманный справочник [Текст] : пер. с англ. / Р. Лигуори, П. Лигуори. – М. : ООО «И.Д.Вильямс», 2016. – 256 с.
11. МакГрат, М. Программирование на Java для начинающих [Текст] / М. МакГрат : пер. с англ. М. А. Райтман. – М. : Издательство «Э», 2016. – 192 с.



12. МакГрат, М. Программирование приложений на Android для начинающих [Текст] / М. МакГрат : пер. с англ. М. А. Райтман. – М. : Эксмо, 2016. – 192 с.
13. Машнин, Т. С. JavaFX 2.0 : Разработка RIA-приложений [Текст] / Т. С. Машнин. – СПб. : БХВ-Петербург, 2012. – 320 с.
14. Новиков, Б. А. Основы технологий баз данных [Текст] : учебное пособие / Б. А. Новиков, Е. А. Горшкова ; под. ред. Е. В. Рогова. – М. : ДМК Пресс, 2019. - 240 с.
15. Стратегическое управление информационными системами [Текст] : учебник / под. ред. Г. Н. Калянова. – М. : Институт Информационных Технологий: БИНОМ. Лаборатория знаний, 2017. – 510 с.
16. Сухомлинов, А. И. Разработка информационных систем [Текст] : учебное пособие / А. И. Сухомлинов – М. : Проспект, 2017. - 112 с.
17. Флэнаган, Д. JavaScript [Текст] : карманный справочник, 3-е изд. / Д. Флэнаган. – М. : ООО «И.Д. Вильямс», 2017 – 320 с.
18. Darwen, Н. An Introduction to Relational Database Theory : [Электронный ресурс] / Н. Darwen. - UK. : [Б.и.] - 2014, 235 p. – Режим доступа: bookboon.com (дата обращения 30.03.2019)
19. Android developers : на англ. яз. : [Электронный ресурс] / Android developer – Режим доступа: <https://developer.android.com> (дата обращения 30.03.2019)
20. Bootstrap : на англ. яз. : [Электронный ресурс] / Bootstrap – Режим доступа: <https://getbootstrap.com> (дата обращения 11.03.2019)
21. Bootstrap 3 Tutorial : на англ. яз. : [Электронный ресурс] / [www.w3schools.com](http://www.w3schools.com) : THE WORLD'S LARGEST WEB DEVELOPER SITE – Режим доступа: <https://www.w3schools.com/bootstrap/default.asp> (дата обращения 11.03.2019)
22. Java SE Technologies : на англ. яз. : [Электронный ресурс] // ORACLE – Режим доступа:

<https://www.oracle.com/technetwork/java/javase/tech/index.html> (дата обращения 11.03.2019)

23. JavaFX 2.2 : All Classes : на англ. яз. : [Электронный ресурс] / ORACLE – Режим доступа: <https://docs.oracle.com/javafx/2/api/index.html> (дата обращения 11.03.2019)

24. JavaFX Scene Builder: User Guide : на англ. яз. : [Электронный ресурс] / ORACLE – Режим доступа: <https://docs.oracle.com/javase/8/scene-builder-2/user-guide/> (дата обращения 11.03.2019)

25. JavaFX: Getting Started with JavaFX : на англ. яз. : [Электронный ресурс] / ORACLE – Режим доступа: <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784> (дата обращения 11.03.2019)

26. JQuery Tutorial : на англ. яз. : [Электронный ресурс] / [www.w3schools.com](http://www.w3schools.com) : THE WORLD'S LARGEST WEB DEVELOPER SITE – Режим доступа: <https://www.w3schools.com/jquery/default.asp> (дата обращения 11.03.2019)

27. The Java™ Tutorials : на англ. яз. : [Электронный ресурс] / ORACLE – Режим доступа: <https://docs.oracle.com/javase/tutorial/index.html> (дата обращения 11.03.2019)

28. SQL Tutorial : на англ. яз. : [Электронный ресурс] / [www.w3schools.com](http://www.w3schools.com) : THE WORLD'S LARGEST WEB DEVELOPER SITE – Режим доступа: <https://www.w3schools.com/sql/default.asp> (дата обращения 11.03.2019)

29. JavaScript Tutorial : на англ. яз. : [Электронный ресурс] / [www.w3schools.com](http://www.w3schools.com) : THE WORLD'S LARGEST WEB DEVELOPER SITE – Режим доступа: <https://www.w3schools.com/js/default.asp> (дата обращения 11.03.2019)

30. A re-introduction to JavaScript (JS tutorial) : на англ. яз. : [Электронный ресурс] / MDN web docs : [moz://a](https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript) – Режим доступа: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/A\\_re-introduction\\_to\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript) (дата обращения 11.03.2019)