|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**КУРСОВАЯ РАБОТА**

по дисциплине «Функциональное и логическое программирование»

**Тема курсовой работы** Анализ предметной области и создание мобильного решения согласно теме варианта.

|  |  |
| --- | --- |
| **Студент группы** ИКБО-11-18 | Володин Андрей Алексеевич |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  |  |
| **Руководитель курсовой работы** | к.т.н., доцент Смольянинова В.А. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) |
|  |  |
| Работа представлена к защите | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  |  |
| Допущен к защите | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

2020

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Утверждаю

Заведующий кафедрой МОСИТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Головин С.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2020 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

«Функциональное и логическое программирование»

|  |  |
| --- | --- |
| Студент Володин Андрей Алексеевич | Группа ИКБО-11-18 |

**Тема работы:** Анализ предметной области и создание мобильного решения согласно теме варианта.

**Исходные данные:** Вариант № 4. Предметная область – Книжный магазин.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

1. Анализ предметной области для нахождения в ней проблем и «болей рынка»
2. Изучения проблематики предметной области на конкретном бизнес процессе в данной предметной области
3. Подготовка минимального функционала по проанализированному бизнес процессу;
4. Разработка схем IDEF0 и DFD для анализа внедрения приложения в работающий бизнес процесс и нагрузки системы данными.
5. Разработка и проектирование архитектуры создаваемой информационной системы.
6. Создание портретов пользователей разрабатываемого приложения
7. Создание информационной схемы мобильного приложения
8. Создание интерактивного макета разрабатываемого мобильного приложения
9. Разработка мобильного приложения;
10. Разработка или подключение остальных элементов разрабатываемой информационной системы.

**Срок представления к защите курсовой работы:** до «\_\_\_»\_\_\_\_\_\_\_ 2020 г.

**Задание на курсовую работу выдал** \_\_\_\_\_\_\_\_\_\_\_\_ (Смольянинова В.А.)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Задание на курсовую работу получил**  \_\_\_\_\_\_\_\_\_\_\_\_ (Володин А.А.)

Оглавление

[Введение 5](#_Toc59631655)

[1 Актуальность темы 5](#_Toc59631656)

[2 Цель и задачи 5](#_Toc59631657)

[3 Структура и объем работы 5](#_Toc59631658)

[3.Теоретическое введение 6](#_Toc59631659)

[4. Проектная часть 10](#_Toc59631660)

[4.1 Анализ поставленной задачи. 10](#_Toc59631661)

[4.2 Функциональные требования 11](#_Toc59631662)

[4.3 Нефункциональные требования 12](#_Toc59631663)

[4.4 Описание реализуемого функционала 12](#_Toc59631664)

[4.5 Архитектура информационной системы 14](#_Toc59631665)

[4.6 Анализ внедрения приложения 16](#_Toc59631666)

[4.7 Портрет пользователя 19](#_Toc59631667)

[4.8 Схема сценариев использования 20](#_Toc59631668)

[4.9 Интерактивная модель 20](#_Toc59631669)

[5. Экспериментальная часть 30](#_Toc59631670)

[5.1 Проектирование и разработка мобильного приложения 30](#_Toc59631671)

[5.2 Файловая структура приложения 52](#_Toc59631672)

[5.3 Подключение Firebase к проекту. 57](#_Toc59631673)

[Заключение 59](#_Toc59631674)

[Список используемой литературы 60](#_Toc59631675)

[Приложения 63](#_Toc59631676)

[Ссылки на проект 63](#_Toc59631677)

# Введение

## 1 Актуальность темы

Библиотеки и книжные магазины чрезвычайно популярны в наше время, даже несмотря на множество цифровых аналогов. Многие люди предпочитают держать в руках именно бумажную версию книги во время чтения. Если же сравнивать библиотеки и книжные магазины между собой, то более распространены именно магазины. Всего в Москве находится порядка 400 библиотек по всему городу, и только магазинов сети «Читай Город» порядка 480, а ведь есть и другие сети или даже отдельные магазины. Из этого можно сделать вывод, что выбранная тема является актуальной в настоящее время.

## 2 Цель и задачи

Целью работы является разработка мобильного приложения "Book shop" для обеспечения доступа ко всей информации по книгам в магазине сотруднику в любой момент времени и в любом месте магазина, а так же для создания запросов на новые книги и получения информации о зарплате.

Для достижения цели работы должны быть решены следующие задачи:

1) осуществить постановку задачи, выделить требования к приложению;

2) произвести обзор существующих решений для хранения и получения информации;

3) изучить современные средства разработки мобильных приложений для Android;

4) определить требования и спроектировать мобильное приложение;

5) реализовать мобильное приложение.

## 3 Структура и объем работы

Работа состоит из введения, четырех глав, заключения и библиографического списка. Объем работы составляет 63 страницы, объем библиографии – 20 источников, объем приложений – 1 страница.

В главе «Теоретическое введение» представлен теоретический материал, необходимый для выполнения курсовой работы.

Глава «Практическая часть» посвящена анализу задания, разработке подходящего решения, описанию реализуемого функционала приложения, созданию портретов пользователей и анализу бизнес процесса.

В главе «Экспериментальная часть» представлено создание приложения и реализация его функций.

В заключении сделаны выводы о проделанной работе и сформулированы перспективы дальнейшей разработки.

# 3.Теоретическое введение

Традиционное распространение книг на протяжении ряда последних лет показывает положительную динамику роста рублёвых продаж. Не стал исключением и 2019 год: оборот книжной отрасли по печатной книге в России по отношению к 2018 г. вырос почти на 3,4% (против 1,4% годом ранее); при этом в ведущих каналах распространения (книжные магазины, интернет магазины, FMCG) этот рост составил почти 3,2% (против 2% в 2018-м). Основным драйвером рынка печатной книги, как и прежде, остаётся интернет-канал, показавший годовой рост оборота на уровне 19% (против 11% роста годом ранее). В то же время в очередной раз отмечаем сохранение отрицательной динамики выручки в несетевой книжной рознице, которая наблюдается на протяжении ряда последних нескольких лет. Причина этого во многом заключается в усилившемся «оттоке» покупателей в интернет-каналы, а также росте популярности электронных книг. Примерно по 20% продаж на российском рынке приходится на детскую, учебную и художественную литературу.

Однако обычные книжные магазины все равно остаются востребованными и количество посетителей в них неизменно высокое. Как правило, книжные магазины обладают либо маленькими размерами, порядка 40-60кв. метров, либо большими, которые включают в себя либо просторный зал, либо несколько небольших залов. Во втором случае в таких магазинах работает порядка 4-7 человек, один из которых постоянно находится за кассой магазина, а один на складе. Посетители таких магазинов часто спрашивают у продавцов, где они могут найти ту или иную книгу, и делают они это по трем причинам:

1. У посетителя мало времени,
2. Посетитель не смог сам найти книгу,
3. Посетитель не уверен в названии книги.

В любом из случаев работник должен в короткие сроки найти требуемую книгу и продать ее посетителю. Однако в настоящее время часто складывается ситуация, что сам работник не знает есть ли такая книга или где она находится. Для того чтобы узнать это ему необходимо пройти к кассе, на стоящем там компьютере найти необходимую книгу и узнать ее местоположение, пройти к нужному месту и отдать ее посетителю. В среднем данный порядок действий занимает от двух до четырех минут. Посетитель может испортить себе настроение или же вовсе покинуть магазин. Таких ситуаций в день возникает порядка 40 в ненагруженный день и около 150 в нагруженный, если брать магазин, находящийся в центре города. Таким образом работник теряет от часа до двух в день на поиски книг. На данный момент нет мобильного приложения (на момент 20.12.2020 в открытом доступе), которое бы решало такие проблемы персонала магазина.

Из-за перехода в цифровой формат есть большая потребность магазинов иметь собственные сайты или же собственные приложения по продаже книг. Из года в год сохраняется рост продаж книг именно через интернет, а не через сам магазин, следовательно, это важный пункт для любого магазина. Однако на данный момент недостатка сайтов или приложений по продаже книг нет на рынке, а собственный сайт сделать можно практически без знаний за несколько дней через конструктора сайтов, поэтому на мой взгляд данная проблематика уступает по важности проблематике отсутствия у продавцов портативной системы навигации по магазину.

Главным образом, приложение для магазина должно иметь следующие функции:

1. Поиск книги в магазине,
2. Запрос на покупку новой книги,
3. Получение информации о сотруднике,
4. Регистрация продажи книги.

Для корректной реализации всех перечисленных функций лучше всего подойдет клиент-серверная архитектура приложения. Данные должны храниться на сервере и при запросе выдаваться пользователю. Так же программа должна корректно отображать данные для нескольких одновременно работающих пользователей. Хранение данных на одном устройстве с помощью базы данных SQLlite не подходит для данного приложения ввиду отсутствия доступа к ней других пользователей и постоянного изменения данных.

Приложение будет служить звеном, соединяющим пользователя продавца и базу данных, в которой хранится вся нужная информация. Пользователю достаточно лишь сделать запрос, и он получит всю необходимую ему информацию.

Для реализации серверной части приложения была использована база данных в реальном времени Firebase от компании Google. Firebase служит базой данных, которая изменяется в реальном времени и хранит данные в JSON. Любые изменения в базе данных тут же синхронизируются между всеми клиентами, или девайсами, которые используют одну и ту же базу данных. Другими словами, обновление в Firebase происходят мгновенно.

Для реализации приложения использовался язык программирования Java версии 12. Java— строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Разработка ведётся сообществом, организованным через Java Community Process; язык и основные реализующие его технологии распространяются по лицензии GPL. Права на торговую марку принадлежат корпорации Oracle.

# 4. Проектная часть

## 4.1 Анализ поставленной задачи.

Разработка мобильных приложений под Android на сегодняшний день очень востребована ввиду высокой популярности данной ОС. Работники книжных магазинов средних размеров не обладают достаточным мобильным программным обеспечением, чтобы сразу отвечать на запросы покупателей, из-за чего им приходится тратить время на перемещение к стационарному компьютеру и поиску информации с него.

На представленном ниже бизнес-процессе в нотации BPMN 2.0 изображен процесс приема заявки на книгу от посетителя. Мы видим, что время работника и посетителя тратится на поход к компьютеру и ожидание результатов.

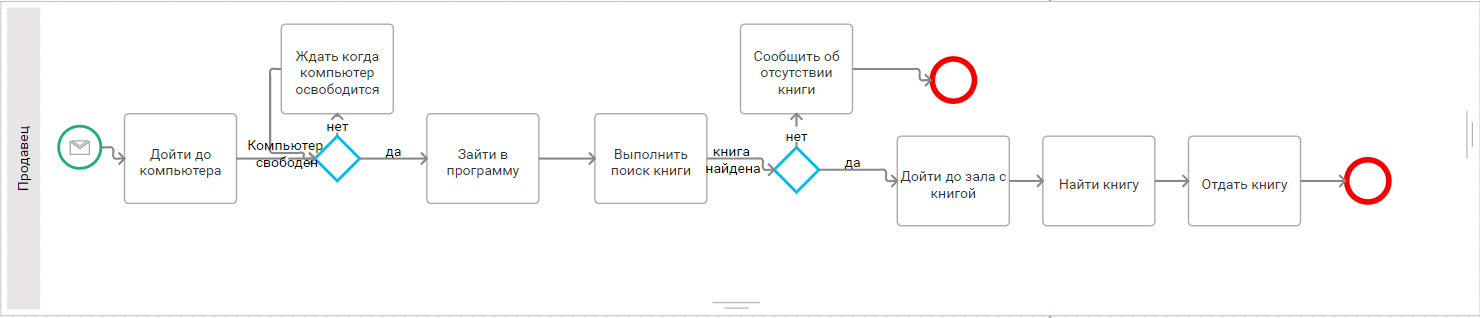


Рисунок 1- Бизнес-процесс поиска книги.

Book shop – мобильное приложение, с помощью которого можно в любой момент времени получить нужную информацию по какой либо области в книжном магазине. Каждый работник магазина устанавливает данное приложение и регистрируется в нем через менеджера магазина. После установки и получения логина и пароля сотрудник может зайти в приложение и приступить к работе. Сотрудник может найти книгу по ее названию или части названия, узнать ее местоположение, а также продать ее, за что получит бонус к зарплате. Так же он может увидеть весь список книг, доступных в магазине и узнать о них полную информацию. Сотрудник имеет доступ к вкладке с полной информации о нем и историей его продаж. Также сотрудник может отправить менеджеру запрос на покупку конкретной книги для магазина.

Так же есть режим менеджера, который может добавлять новые книги и работников, удалять их, получать от них запросы и имеет доступ к полной информации о пользователях.

На представленном ниже бизнес-процессе приема заявки на книгу от посетителя видно, как он упростился по сравнению с предыдущим и времени как работника, так и посетителя тратится гораздо меньше.

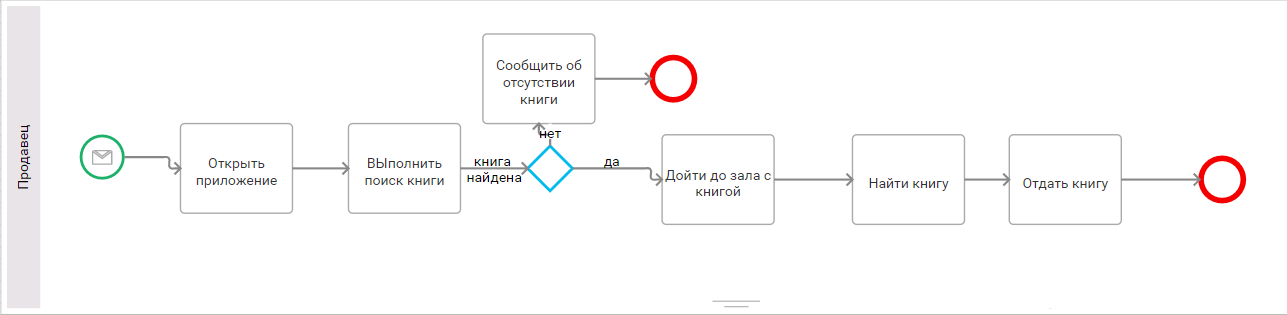


Рисунок 2- Бизнес-процесс поиска книги с использованием приложения.

## 4.2 Функциональные требования

Исходя из исследования предметной области были выделены следующие функциональные требования к приложению:

1) пользователь должен иметь возможность войти в приложение используя логин и пароль;

2) пользователь должен иметь доступ к информации о книгах и их местоположению в магазине;

3) пользователь должен иметь возможность войти в режим менеджера;

4) режим менеджера должен предоставлять возможность добавления и удаления книг и сотрудников, а также получать запросы от сотрудников;

5) пользователь должен иметь возможность отправить запрос на приобретение новой книги в магазин;

6) пользователь должен иметь доступ к информации о себе и своей зарплате;

7) пользователь должен иметь возможность изменить положение книги в приложении.

## 4.3 Нефункциональные требования

Исходя из исследования предметной области и опыта эксплуатации мобильных устройств и приложений были выделены следующие нефункциональные требования:

1) приложение должно быть разработано для ОС Android (версия

Lollipop и более поздние);

2) приложение должно работать в автономном режиме (без подключения к сети Интернет);

3) интерфейс приложения должен быть выполнен в соответствии с принципами;

4) интерфейс приложения должен быть на русском языке

## 4.4 Описание реализуемого функционала

В первую очередь, приложение создается для доступа к информации, которая поможет в продаже книг покупателям. Главной функцией приложения является предоставление доступа к базе данных магазина. Первой функцией будет являться поиск книги по ее названию или части названия. Пользователь вводит название или его часть в строку и нажимает кнопку поиска. После чего на экране появляются все элементы базы данных, которые являются книгами и имеют в своем названии указанную строку. Для этого создается запрос к базе данных, который возвращает подходящие элементы из базы данных. После чего пользователь может нажать на интересующую его книгу и выбрать один трех вариантов действия: вернуться на предыдущий экран, изменить местоположение книги, или оформить ее продажу. Продажа книги данным продавцом прибавляет определенную сумму к его зарплате.

Второй функцией будет являться предоставление полного списка литературы в магазине. При нажатии на определенный элемент будет выводиться информация о нем.

Третьей функцией будет являться создание запроса на покупку новой книги. Данный запрос после его создания будет отправлен менеджеру. Пользователь вводит название книги, автора и издателя и нажимает кнопку «отправить».

Четвертой функцией будет являться получение информации о текущем сотруднике. Перейдя на вкладку с информацией, сотрудник будет видеть свои данные, текущую зарплату и список его продаж.

Так же приложение будет иметь альтернативный режим, режим менеджера, которому доступны другие функции. Для попадания в данный режим необходимо при входе в приложение ввести в поля логина и пароля «admin».

Первой функцией режима менеджера будет являться добавление новых сотрудников или же удаление уже существующих. Перейдя во вкладку добавления сотрудника, пользователь указывает его данные, включающие в себя имя, фамилию, электронную почту, логин, пароль, дату и размер зарплаты.

Второй функцией режима менеджера будет являться добавление или удаление книг. Перейдя во вкладку добавления книг, пользователь вводит всю информацию о книге и заносит ее в базу данных.

Третьей функцией режима менеджера будет являться получение запросов от сотрудников. Перейдя во вкладку запросов, пользователь будет видеть запросы от сотрудников. При нажатии на запрос пользователь может либо оставить его, либо удалить.

Однако данное приложение не может функционировать в полной степени самостоятельно, так как не учитывает продажи книг покупателями, которые сами взяли книги. Для этого требуется разработать приложение для ПК, которое будет каждую проданную книгу удалять из базы данных.

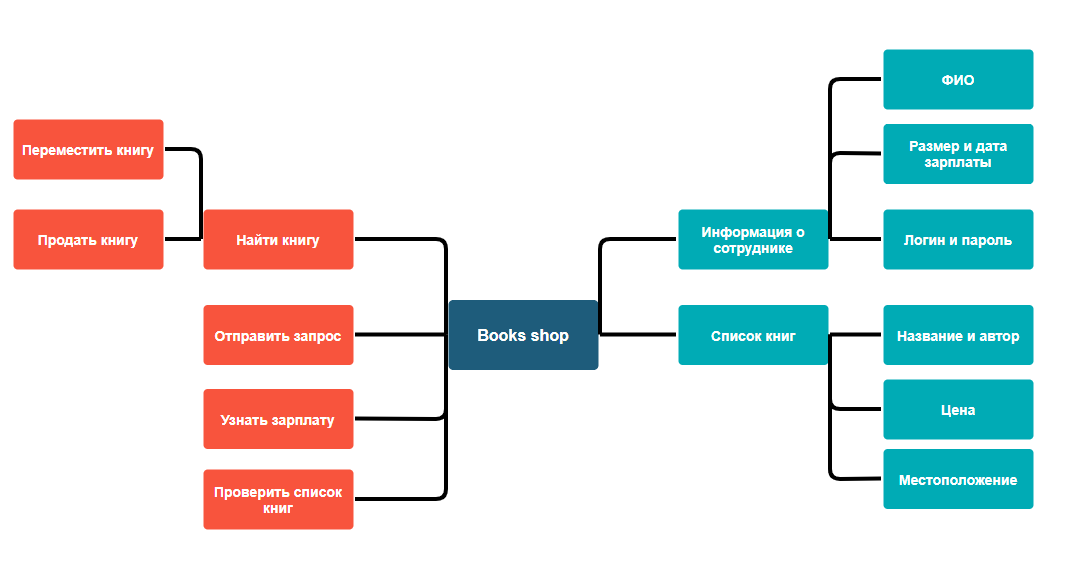


Рисунок 3- Информационная схема приложения.

## 4.5 Архитектура информационной системы

Данное приложение будет обладать клиент-серверной архитектурой. Все данные будут храниться на удаленном сервере базы данных, доступ и обработка данных будут производиться с клиентского оборудования. Данное приложение не совершает сложных операций с данными, осуществляя только получение данных, добавление, изменение и удаление данных. Для таких целей была выбрана Realtime Database Firebase.

Firebase – это одно из BaaS-решений (Backend as a Service), которое дает разработчику массу возможностей.

Это и сервер, и база данных, и хостинг, и аутентификация в одной платформе. Так, Firebase Realtime Database предоставляет разработчикам API, который синхронизирует данные приложения между клиентами и хранит их в облачном хранилище.Приложение подключается к базе данных через WebSocket, который отвечает за синхронизацию данных в течение всего сеанса.

Для разработки приложения была выбрана среда разработки Android Studio- официальная IDE для Android, в основе проекта лежит Intellij IDEA. В Android Studio есть автодополнение кода, удобный debug-режим, в качестве сборщика проектов используется Gradle. В начале разработки можно выбрать один из нескольких готовых шаблонов мобильного приложения. IDE поддерживает NDK и позволяет редактировать проектные файлы на C/C++, а также обладает для них встроенным дебаггером на основе LLDB. Android Studio обладает очень удобным редактором макетов с drag-and-drop интерфейсом, позволяющим быстро вносить правки в файлы разметки. Начиная с версии 2.3, была добавлена функция Instant Run, значительно ускоряющая процесс внесения изменений и запуска их на тестируемом устройстве. Платформа является результатом сотрудничества Google и JetBrains, рекомендована сообществом Android-разработчиков для разработки мобильных приложений.

Данная среда разработки является самой популярной и доступной средой среди разработчиков мобильных приложений под операционную систему Android.

Клиент-серверная архитектура позволяет разгрузить сеть и поддерживать непротиворечивость данных за счет их централизованной обработки. Однако, языки хранимых процедур не приспособлены для полноценной реализации бизнес-логики. Поэтому бизнес-логика в клиент-серверных ИС по-прежнему реализуется на клиентских компьютерах. Такой подход имеет следующие недостатки:

* любые изменения в бизнес-логике требуют обновления на клиентском компьютере;
* клиентские компьютеры должны быть достаточно производительными;
* слабая защита данных от взломов.



Рисунок 4- Клиент серверная архитектура.

## 4.6 Анализ внедрения приложения

Программа, разработанная в данной работе, помогает сотрудникам экономить время, исключая из их действий походы к стационарному компьютеру и обратно, и, следовательно, увеличивает их продуктивность в течение всего дня.

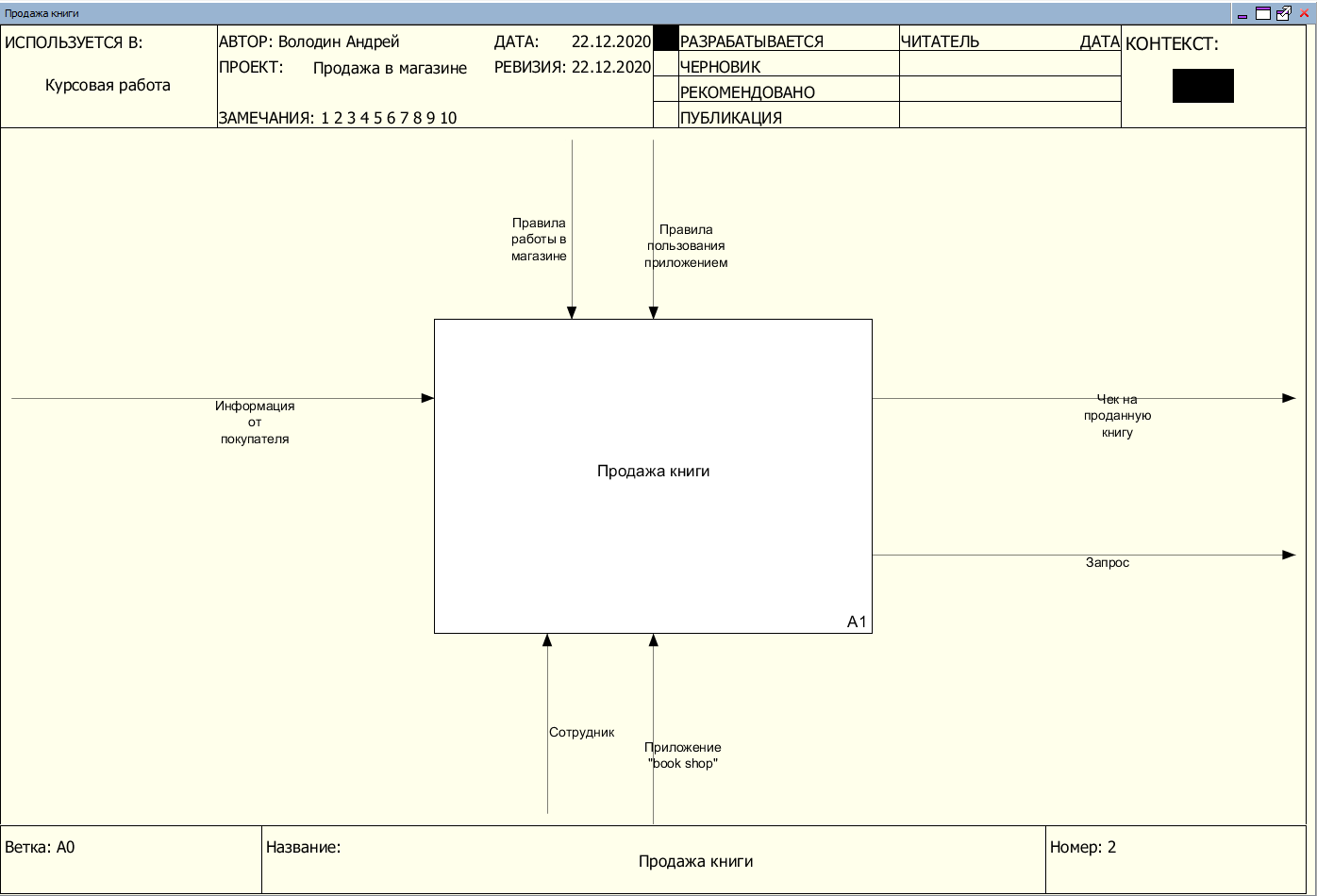


Рисунок 5- IDEF0 диаграмма продажи книги

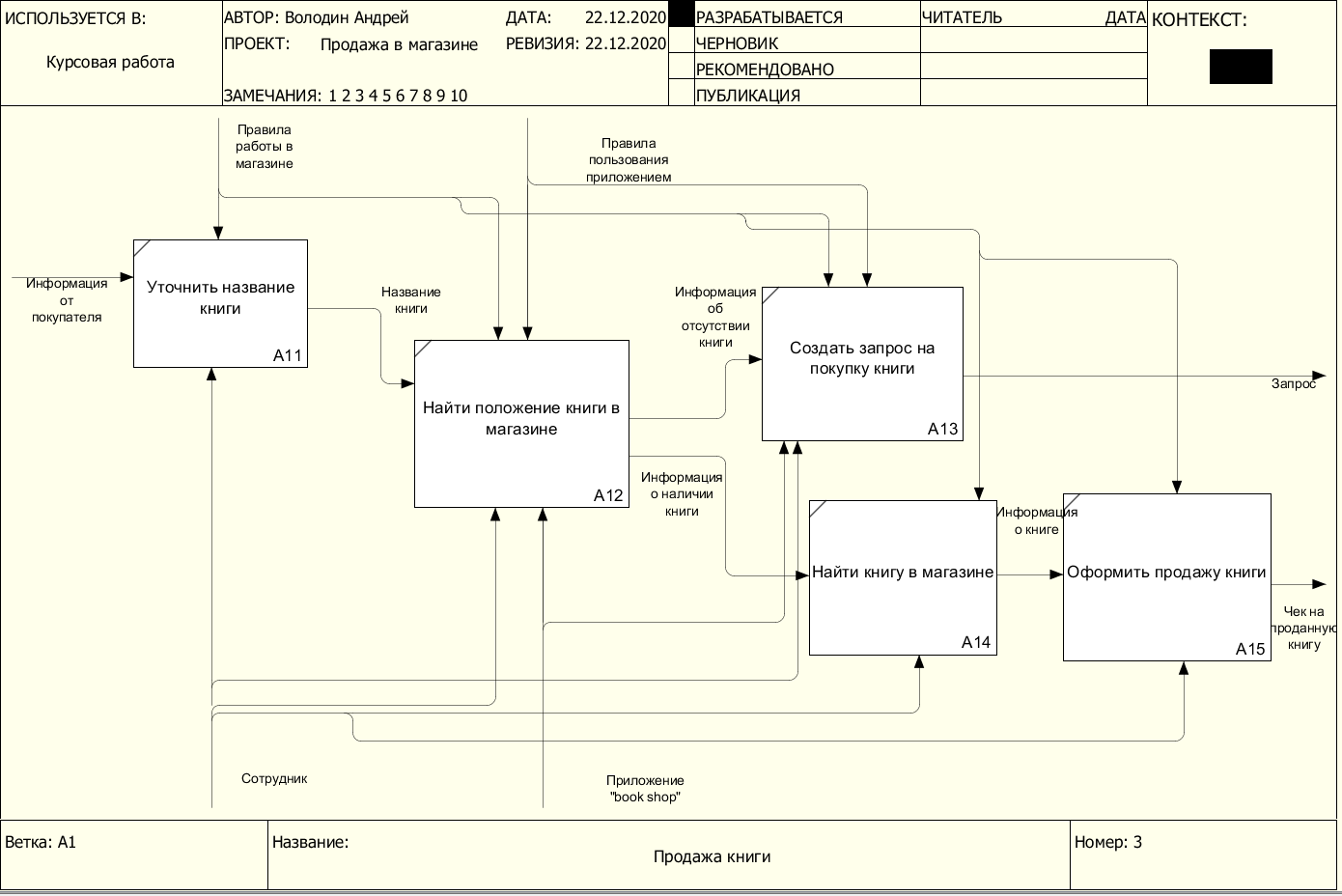


Рисунок 6-Дочерняя диаграмма процесса продажи книги

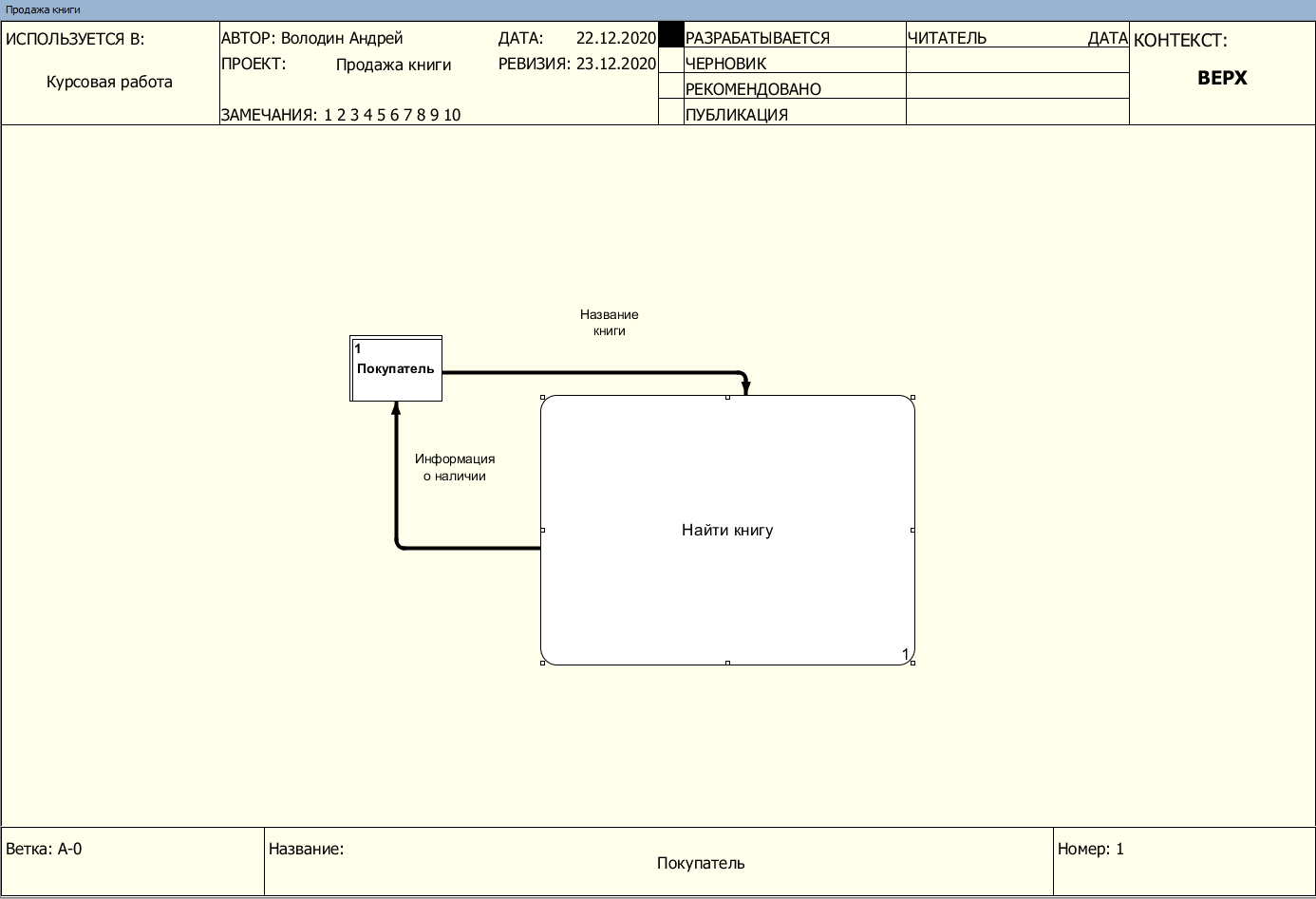


Рисунок 7-DFD диаграмма процесса нахождения книги в магазине через приложение

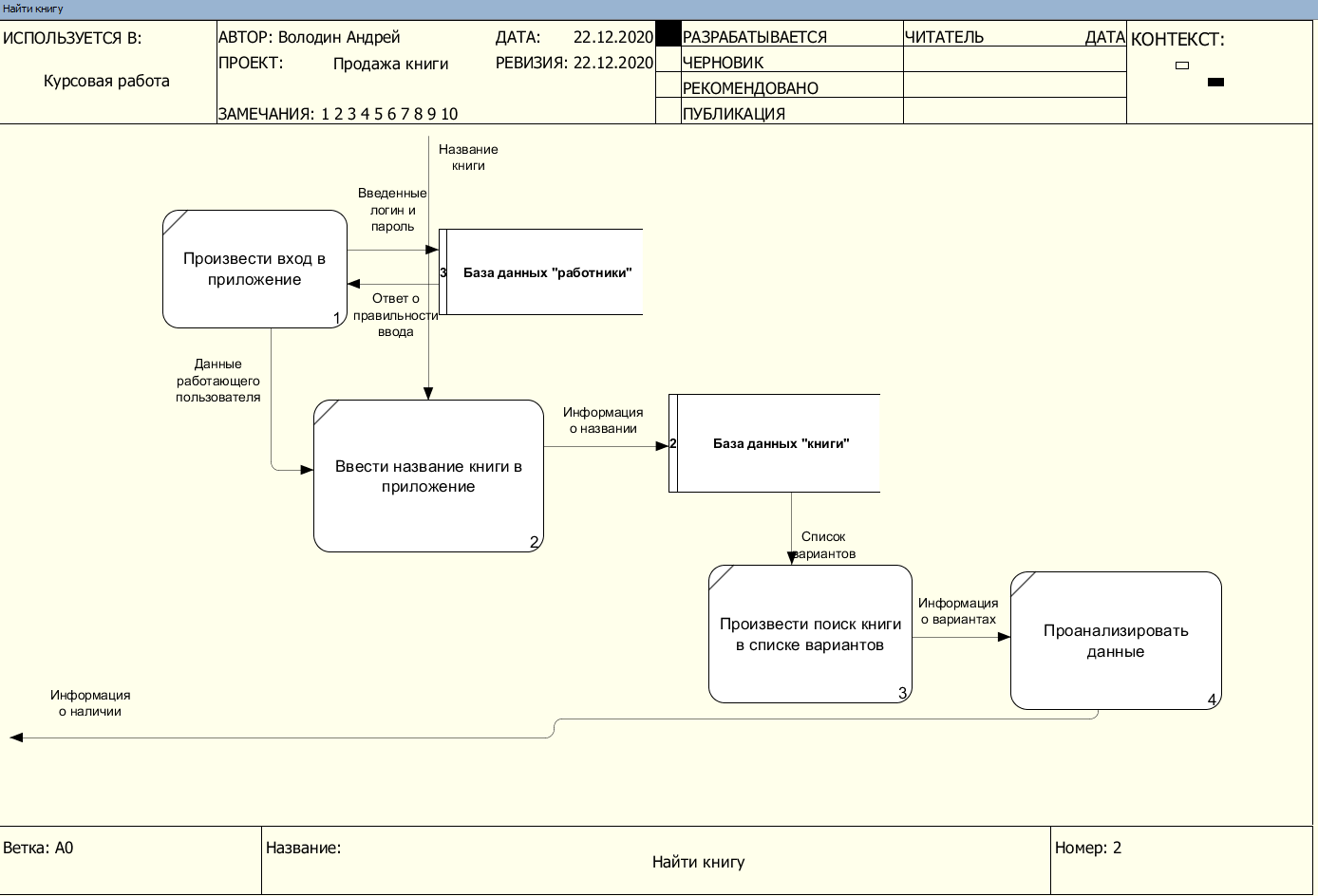


Рисунок 8-Дочерняя DFD диаграмма.

## 4.7 Портрет пользователя

Данное приложение рассчитано на довольно узкую специализацию, поэтому разнообразия портретов пользователей у данной работы нет. Она рассчитана на средние по размеру книжные магазины, в магазинах малого размера потребности в таком приложении нет, в магазинах большого размера слишком много может быть единиц товара и приложение может не выдерживать таких объемов информации или же работать медленно.

Однако, заказчик приложения и его конечный пользователь скорее всего не будут совпадать. Заказчиком в данном случае является владелец магазина. Его не удовлетворяют продажи магазина или же эффективность персонала. В данном случае владелец будет искать решения данных проблем, одним из которых и является программный продукт, разработанный в данной курсовой работе. Владелец получит значительный прирост в производительности своих сотрудников, а сотрудники получат довольно удобное средство для ориентирования по продукции.

Главными потребностями пользователя являлись именно получение данных о конкретной книге или о книгах в магазине в целом. Второстепенными потребностями было получение данных о его заработной плате на данный момент и о создании запросов на новые книги. Программа, созданная в ходе данной курсовой работы должна выполнять потребности как заказчика, так и рядового пользователя.

Так же в данное приложение может быть добавлен вид аутентификации «Гость», который будет обладать ограниченным функционалом и будет предоставлять доступ только к поиску книги. Такое приложение может быть установлено на стационарную технику и использоваться любым посетителем в магазине, а не только работниками. Данный подход убирает проблему поиска свободного сотрудника в книжном магазине.

## 4.8 Схема сценариев использования

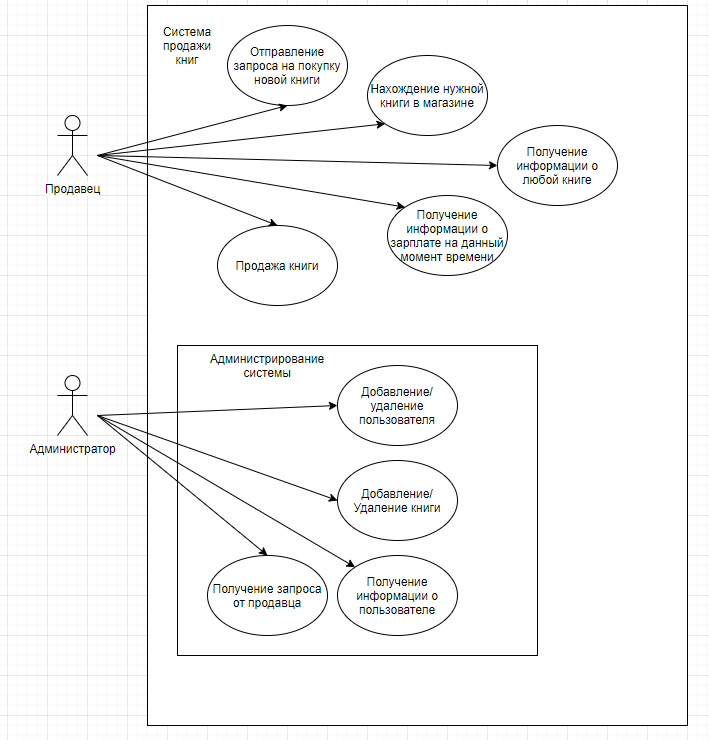


Рисунок 9-Сценарии использования программы.

## 4.9 Интерактивная модель

Для создания интерактивной модели будущего приложения использовался онлайн конструктор «wondershare mockitt», который обладает полным функционалом, необходимым для создания дизайна и интерактивной модели будущего приложения.

Первый экран содержит поля для ввода логина и пароля, которые уникальны у каждого продавца.

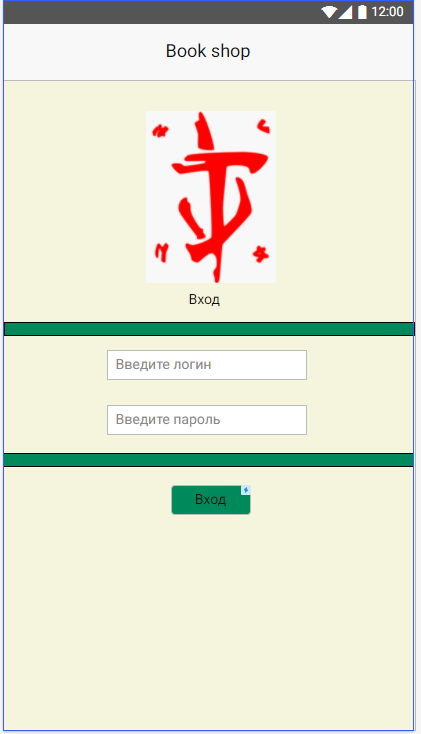


Рисунок 10-Экран входа.

После экрана входа пользователь попадает на экран главного меню, который содержит минимальную информацию о пользователе и кнопки для перехода на экраны поиска книги, показа всех книг, информации о зарплате и создания запроса.

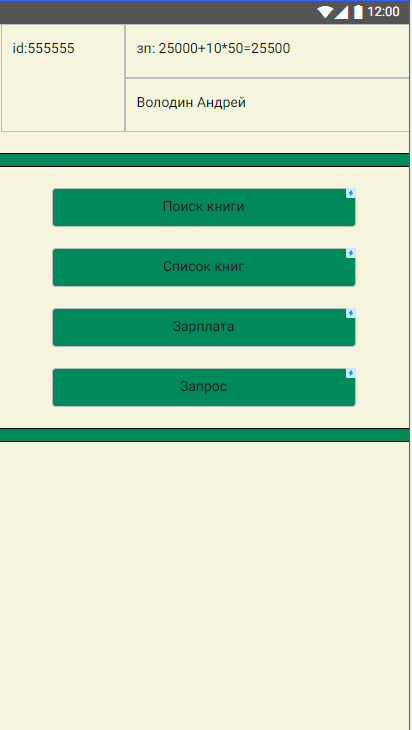


Рисунок 11-Главное меню программы.

Первая кнопка переносит нас на экран поиска книги, который состоит из поля для ввода текста, кнопки и элемента listview, который отображает результат.

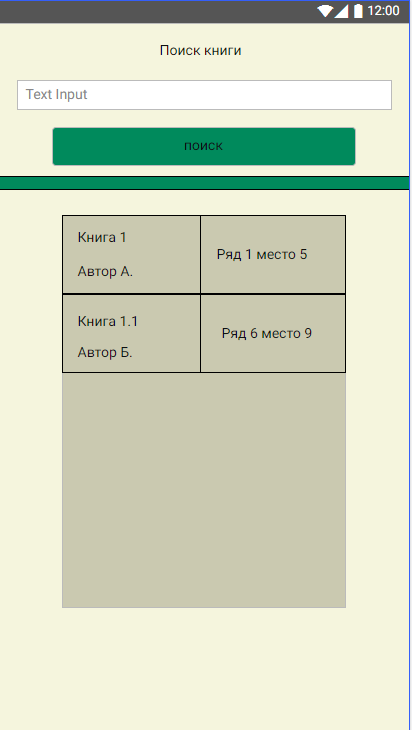


Рисунок 12-Экран поиска книги.

Вторая кнопка переносит нас на экран, в который выводятся все книги, он содержит кнопку возврата на предыдущий экран и listview, который будет содержать элементы (книги).

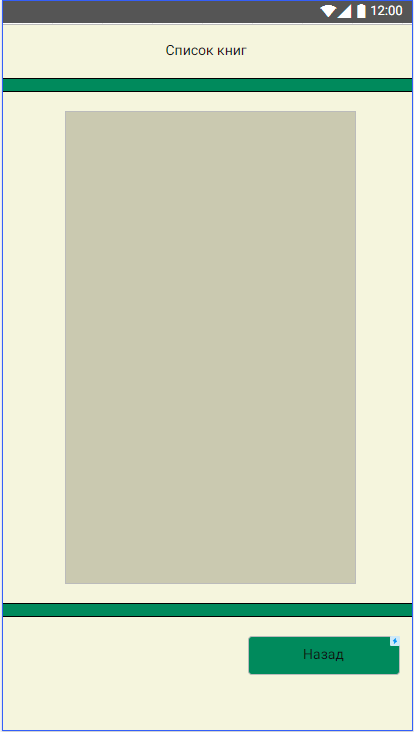


Рисунок 13-Экран вывода книг.

Третья кнопка переносит нас на экран, содержащий всю информацию о работнике, его id, зарплату и список его продаж в отдельном listview.

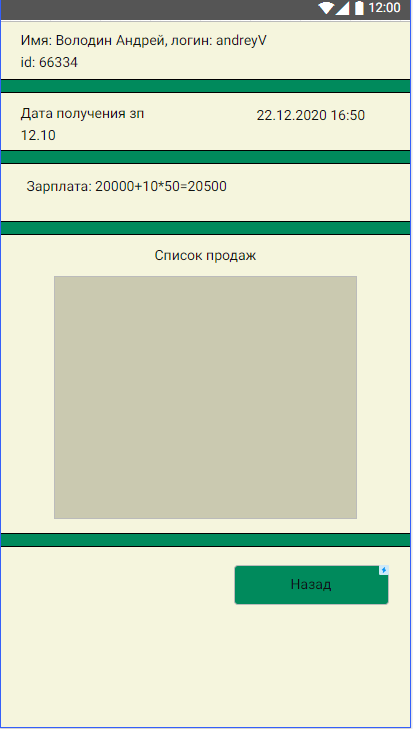


Рисунок 14-Экран зарплаты.

Четвертая кнопка переносит нас на экран создания запроса. Он содержит 4 поля для ввода текста, кнопку, при нажатии на которую из введенных полей берется информация, создается запрос и отправляется в базу данных, после чего этот запрос сможет увидеть менеджер и ответить на него.

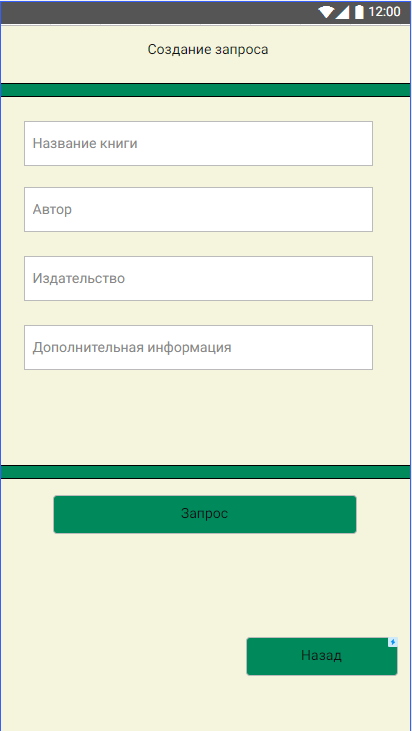


Рисунок 15-Экран запроса.

Режим менеджера содержит 4 экрана, первый из них это главное меню, которое содержит кнопки для доступа к добавлению сотрудника, добавлению книги и списку запросов. Так же этот экран содержит listview с информацией о каждом сотруднике.

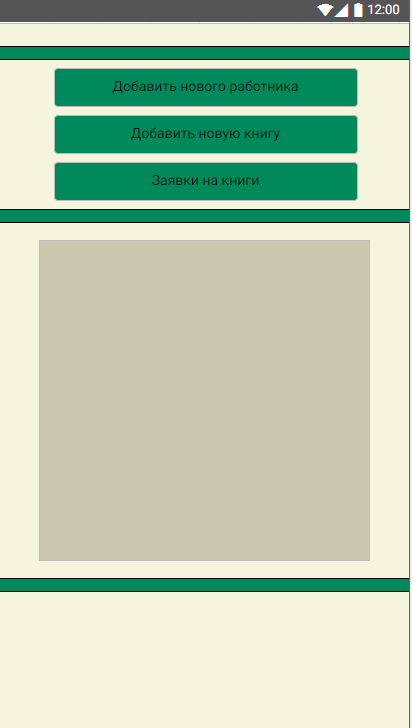


Рисунок 16-Главное меню режима менеджера.

При нажатии на первую кнопку происходит переход на экран добавления нового работника. Он содержит 7 полей для ввода информации, а именно:

* имя
* фамилия
* почта
* логин
* пароль
* дата получения зарплаты
* размер зарплаты

При вводе всех полей и нажатии на кнопку происходит добавление нового сотрудника в базу данных.

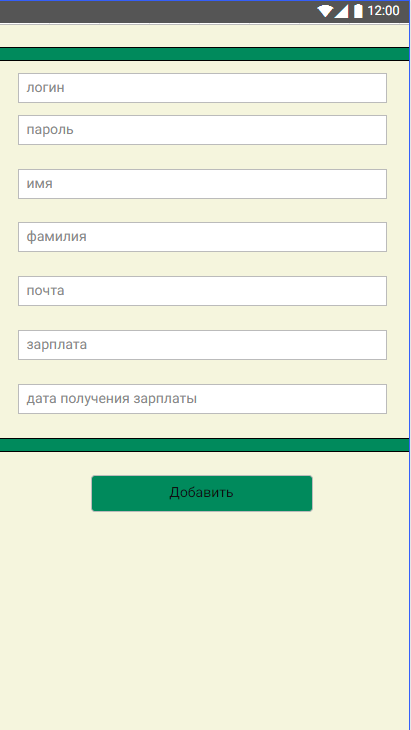


Рисунок 17-Экран добавления нового сотрудника.

При нажатии на вторую кнопку происходит переход на экран добавления новой книги. Он содержит 7 полей для ввода текста и одну кнопку. При вводе всех данных и нажатии на кнопку происходит добавление в базу данных новой записи.

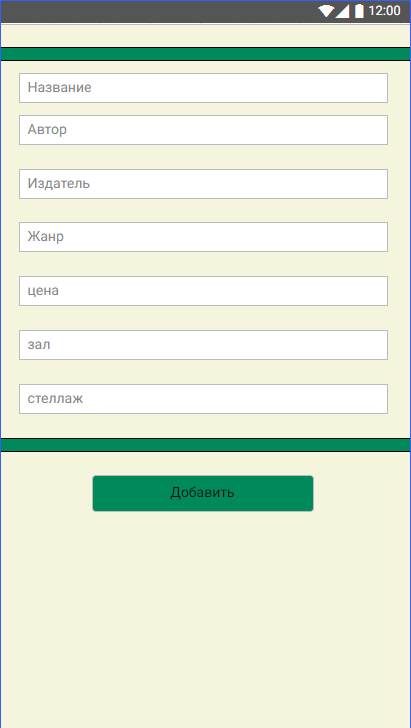


Рисунок 18-Экран добавления новой книги.

При нажатии на третью кнопку происходит переход на экран, содержащий listview, в котором будут отражаться все запросы, созданные работниками и на которые еще не было ответа.

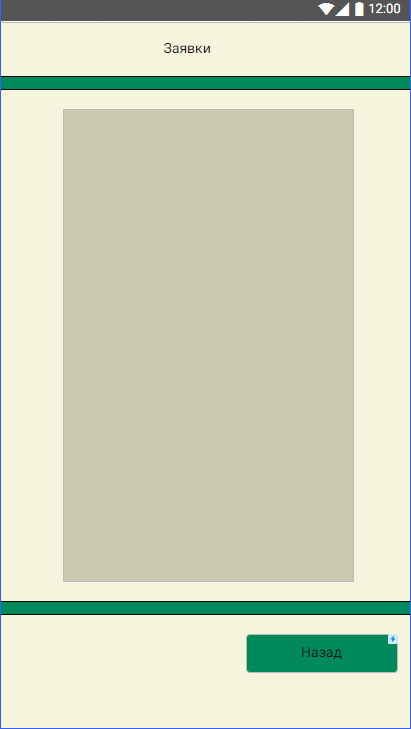


Рисунок 19-Экран вывода запросов.

# 5. Экспериментальная часть

## 5.1 Проектирование и разработка мобильного приложения

Далее будут представлены скриншоты различных окон приложения, которое тестируется на реальном физическом устройстве HONOR 10. Приложение реализовано на языке Java с использованием Firebase. Работа выполнена в приложении Android Studio. Версия SDK: 30.

Первым был разработан экран входа в приложение. На нем находятся два поля EditText для ввода логина и пароля пользователя. После их ввода происходит обращение к базе данных firebase, а именно к узлу «users», из которого в программу отправляются данные о пользователях. Далее происходит сравнение введенного логина с логинами всех работников и при нахождении совпадения проверяется пароль, если и логин, и пароль совпадают то пользователь переходит на следующий экран. Если же один из пунктов не совпадает, то выдается ошибка.

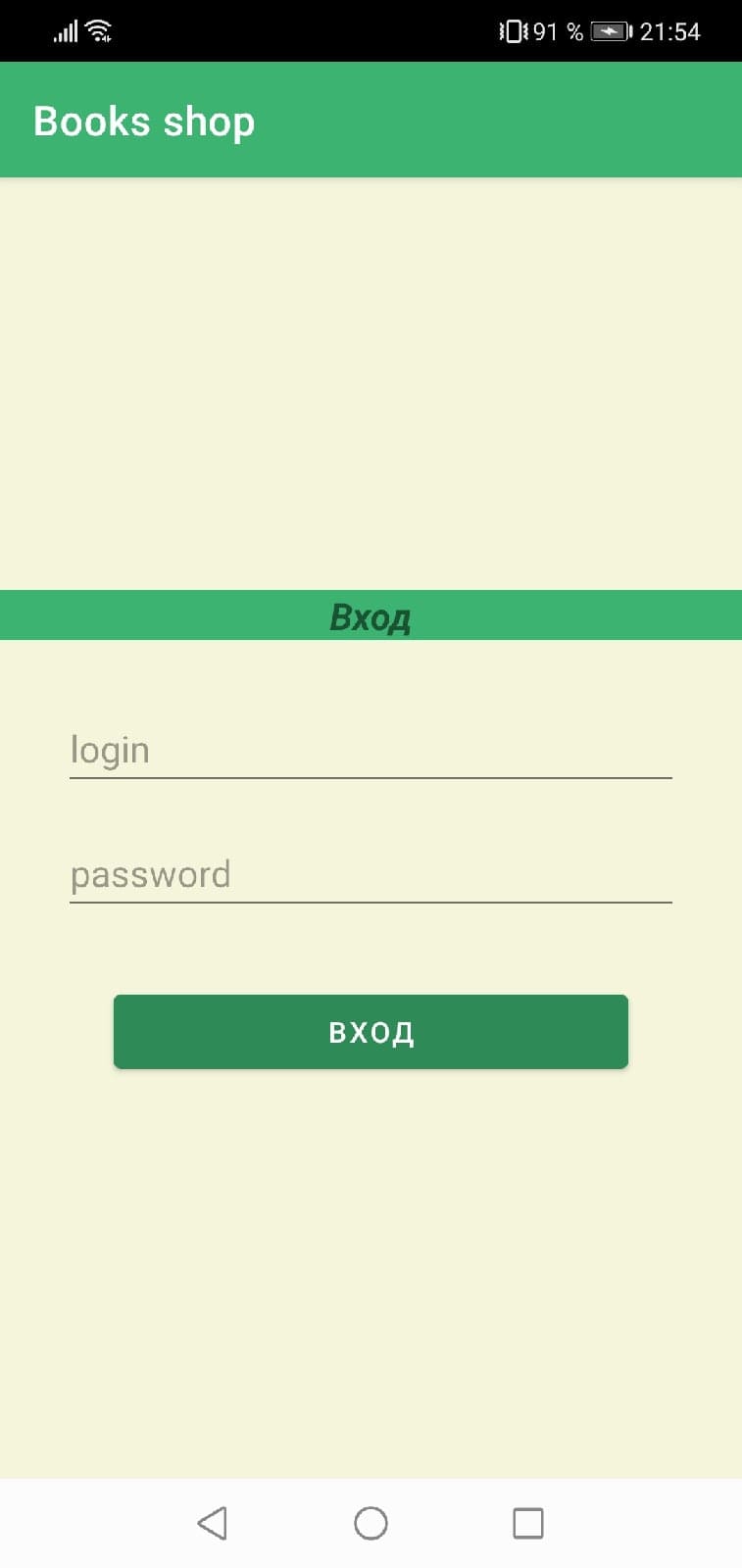


Рисунок 20-Экран входа в приложение.

Далее пользователь переходит на экран главного меню приложения. На данном экране находится минимальная информация о пользователе, его id, зарплата на данный момент и имя с фамилией. Так же тут находятся 4 кнопки для перехода на другие экраны: поиск книги, список книг, зарплата и запрос. При переходе на данную страницу приложение устанавливает в 3 textview информацию из класса пользователя, который сейчас использует приложение, так же запускается поток, который один раз в 30 секунд обновляет данную информацию. Приложение связывается с базой данных, получает от нее данные пользователя по его логину и id, обновляет работающего пользователя и выводит их на экран. Далее представлены главные меню двух пользователей.

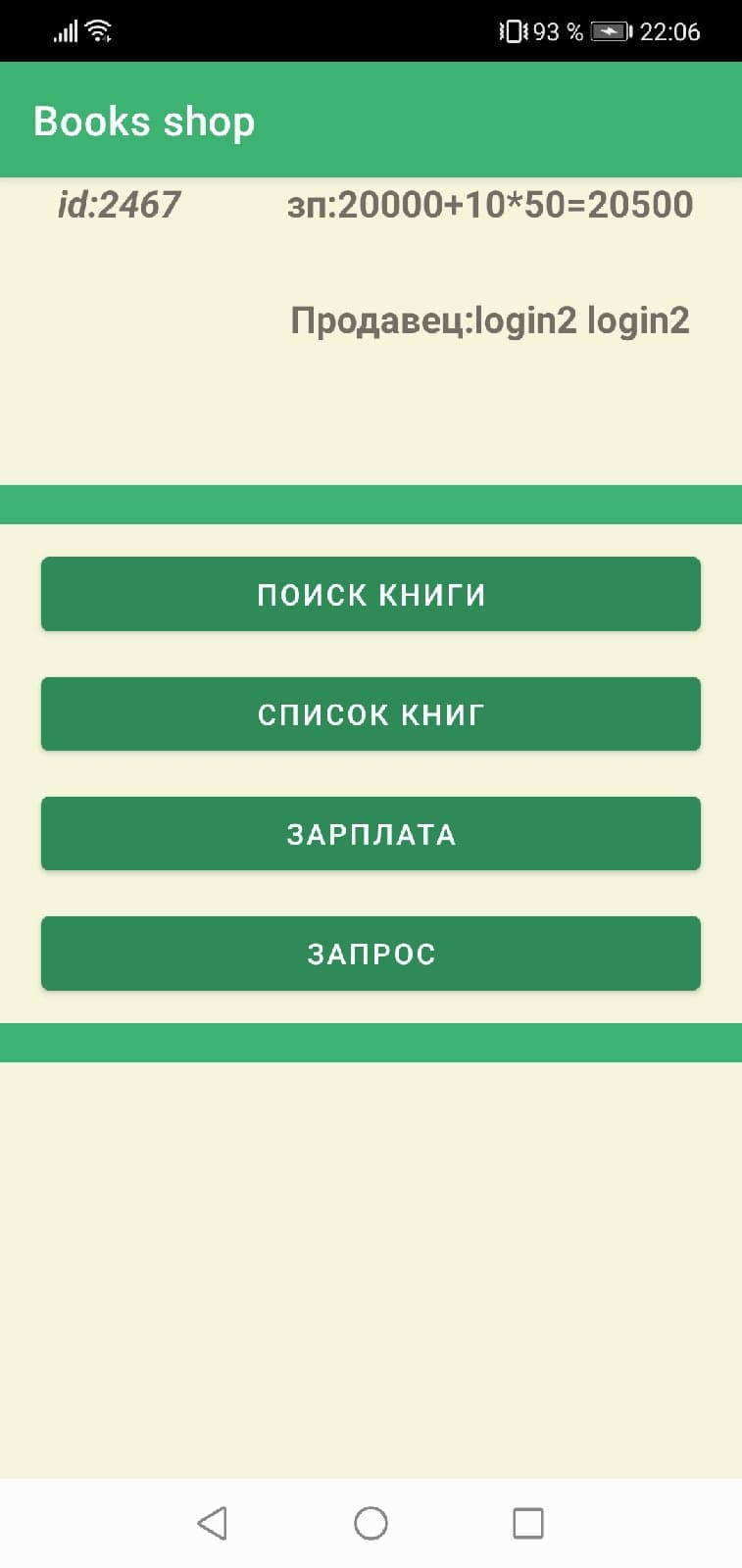


Рисунок 21-Главное меню пользователя login2.

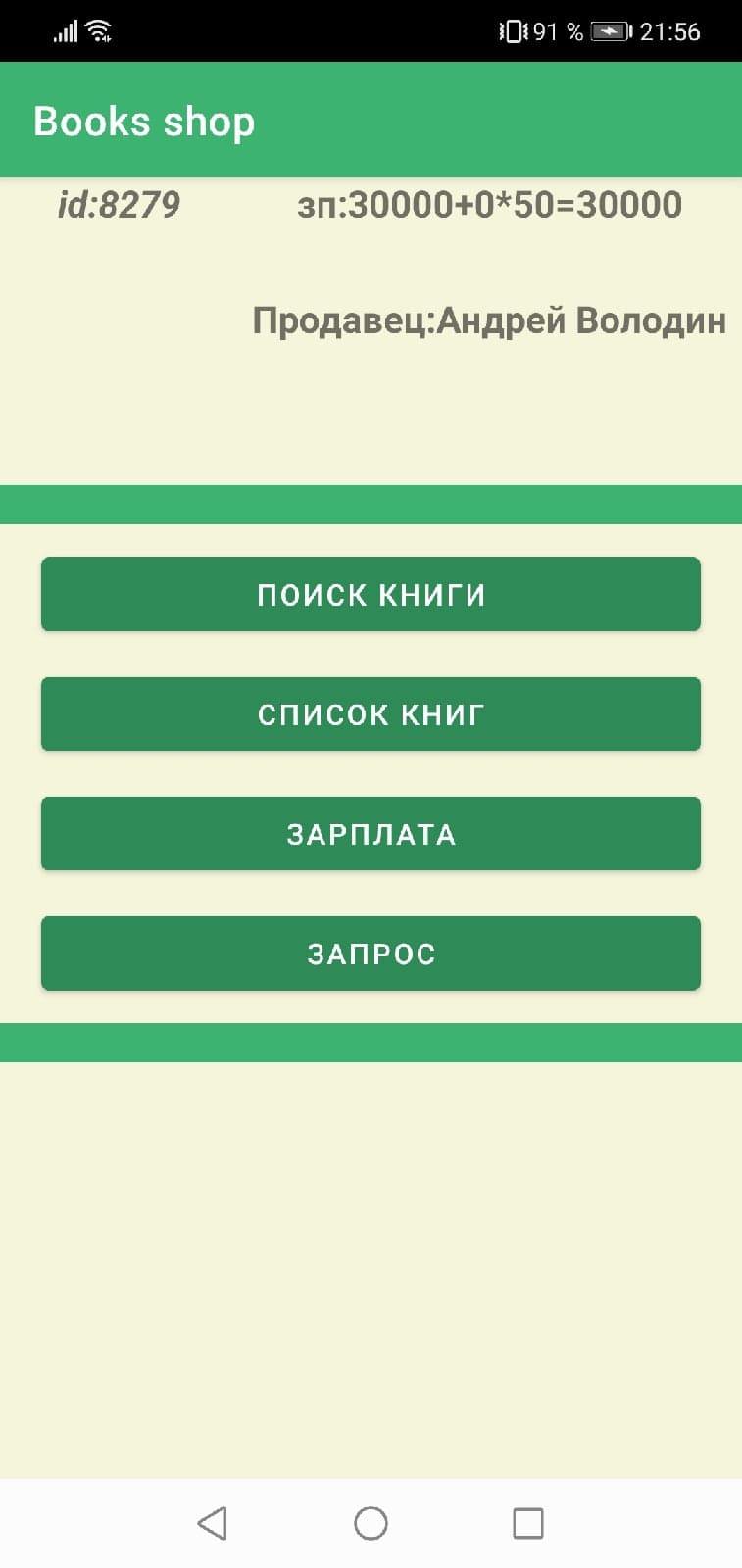


Рисунок 22-Главное меню пользователя Андрей Володин.

При нажатии на кнопку «список книг» пользователь переходит на экран, содержащий textview с «список книг», listview в который через адаптер загружаются данные о всех книгах в магазине и кнопка, возвращающая на предыдущий экран. При запуске этого окна приложение связывается с базой данных и по ключу «books» получает значения из базы данных. После чего данные помещаются в список и через адаптер выводятся на экран в listview. Выводится краткая информация о книге: ее название, автор и цена.

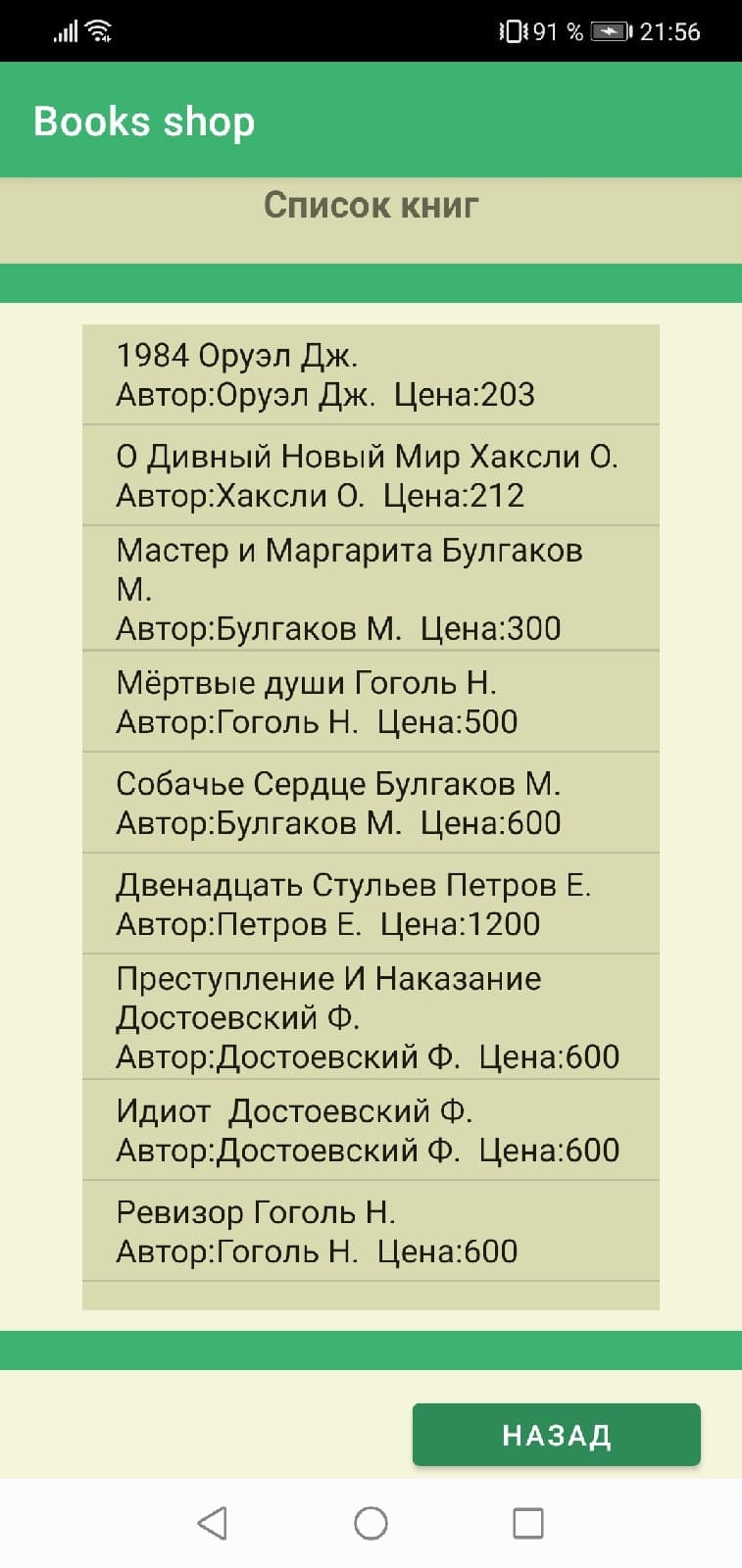


Рисунок 23-Экран вывода всех книг в магазине.

При нажатии на какую-либо книгу появляется всплывающая активность alertDialog, содержащий полную информацию о книге: ее id, название, автора, издателя, год выпуска, место в зале и цену. Единственная доступная кнопка в активности закрывает диалог, так же из него можно выйти, нажав на любое место вне диалога.

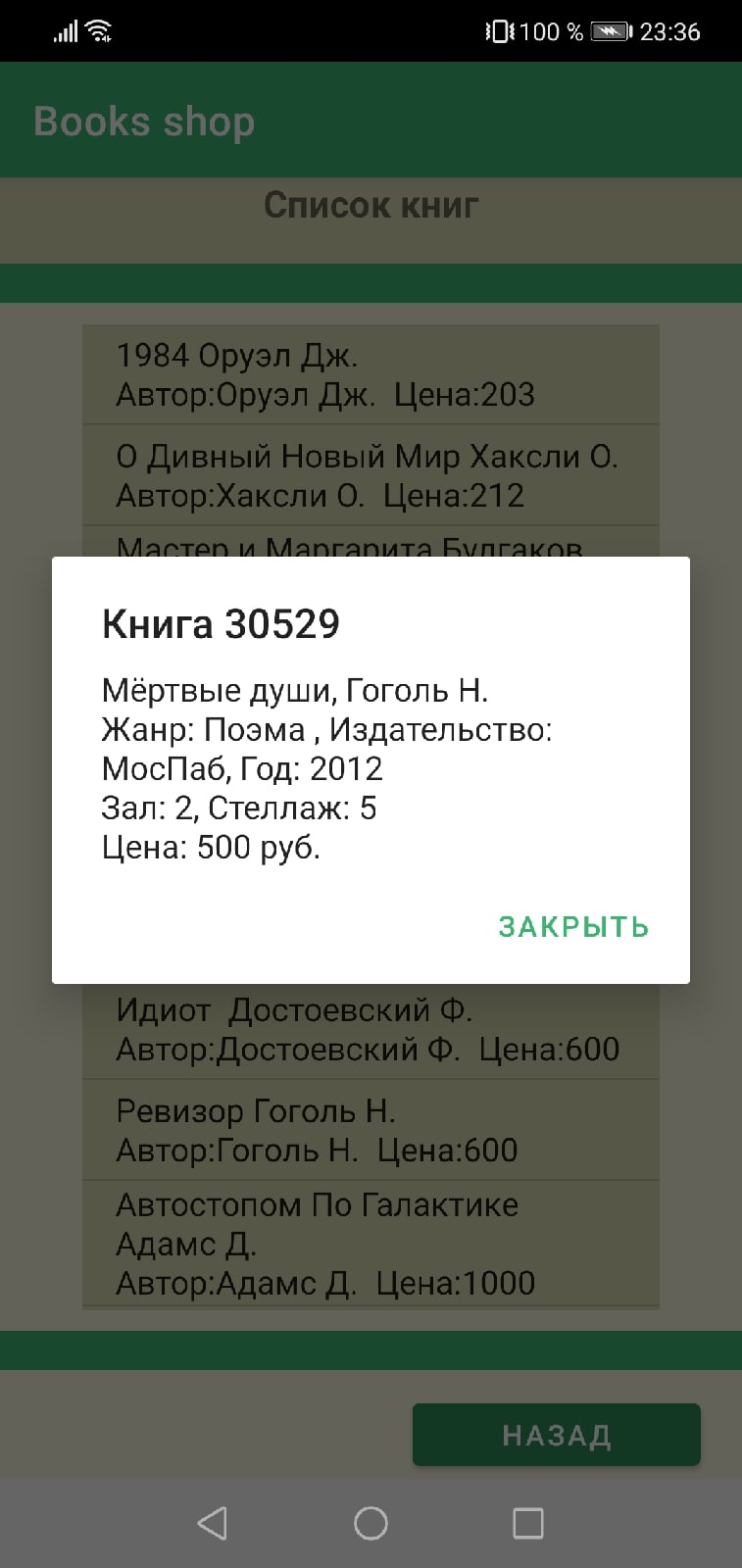


Рисунок 24-Диалог при нажатии на элемент списка.

Далее переходим в главное меню и нажимаем на кнопку «найти книгу». Она перемещает нас на экран поиска определенной книги в магазине. Данный экран содержит одно поле для ввода названия, кнопку, при нажатии на которую происходит поиск и listview, в который помещаются данные поиска. В listview данные помещаются через адаптер. При нажатии на кнопку происходит обращение к базе данных по ключу «books» и данные, полученные из базы данных записываются в список. Далее элементы списка проверяются на содержание введенной строки в названии, если название книги содержит введенную строку, то книга добавляется в адаптер для вывода в listview. Пользователь до нажатия на элемент видит название книги, ее автора, зал и стеллаж, где находится книга.

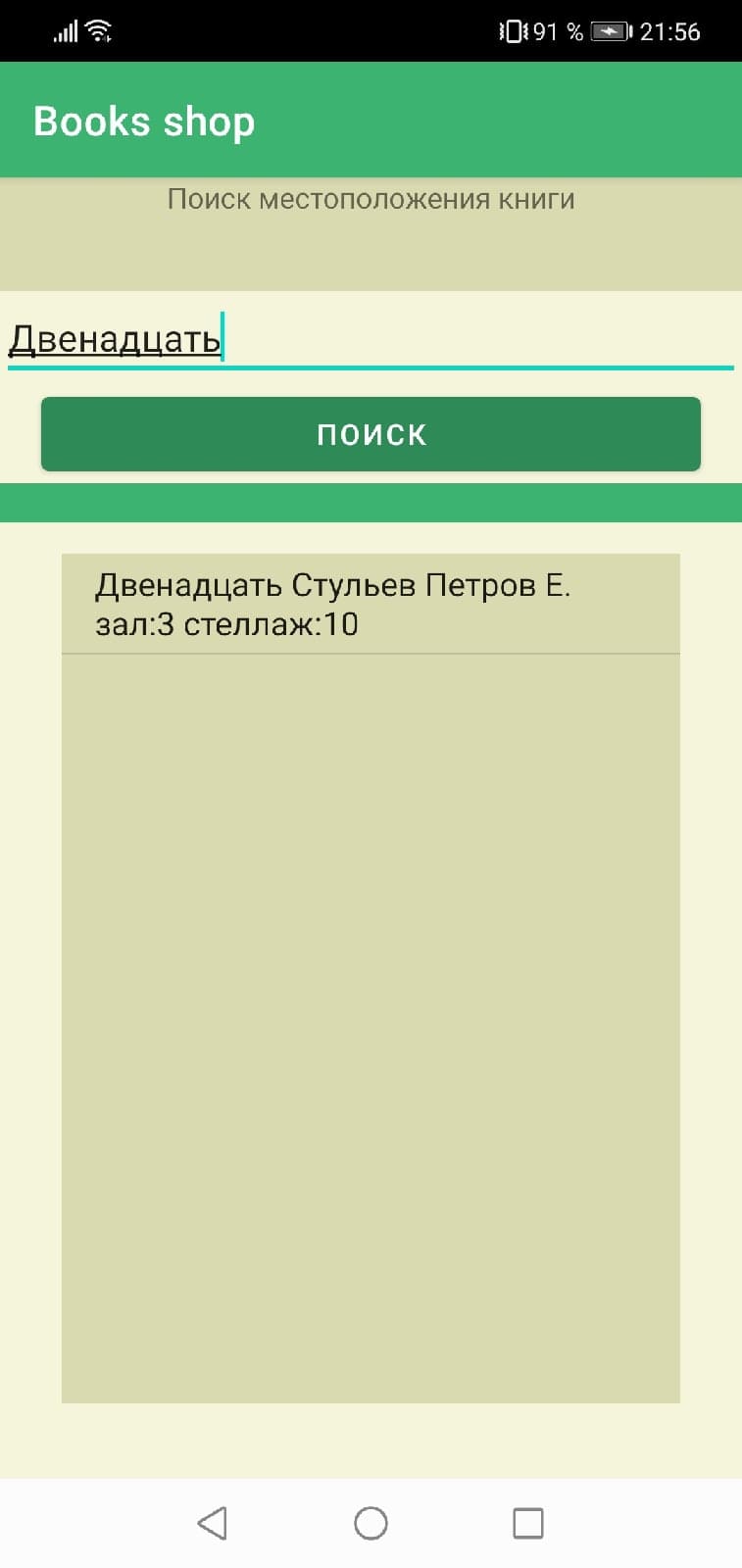


Рисунок 25-Экран поиска книги.

При нажатии на какую-либо книгу появляется всплывающая активность alertDialog, содержащий полную информацию о книге: ее id, название, автора, издателя, год выпуска, место в зале и цену. Так же в активности доступны две кнопки, «продать книгу» и «переместить книгу».

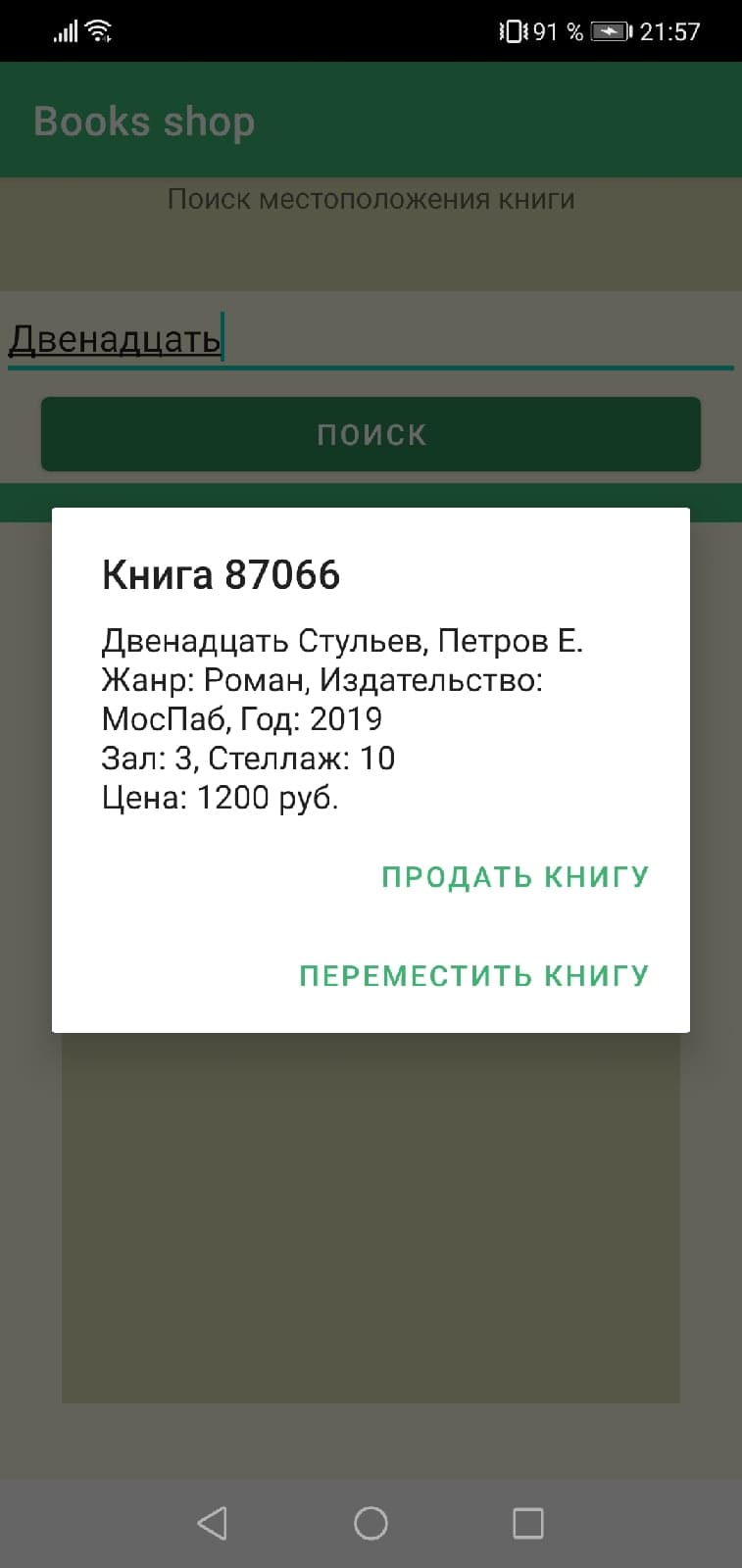


Рисунок 26-Диалог при нажатии на книгу.

При нажатии на кнопку переместить книгу открывается новый диалог, содержащий поле для ввода данных, в которое вводится через пробел номер зала и номер стеллажа, куда перемещается книга. После этого происходит обращение к базе данных и находится элемент, имя и id которого совпадает с изменяемой книгой. Далее ее поля hall и rack изменяются в соответствии с новыми введенными данными.

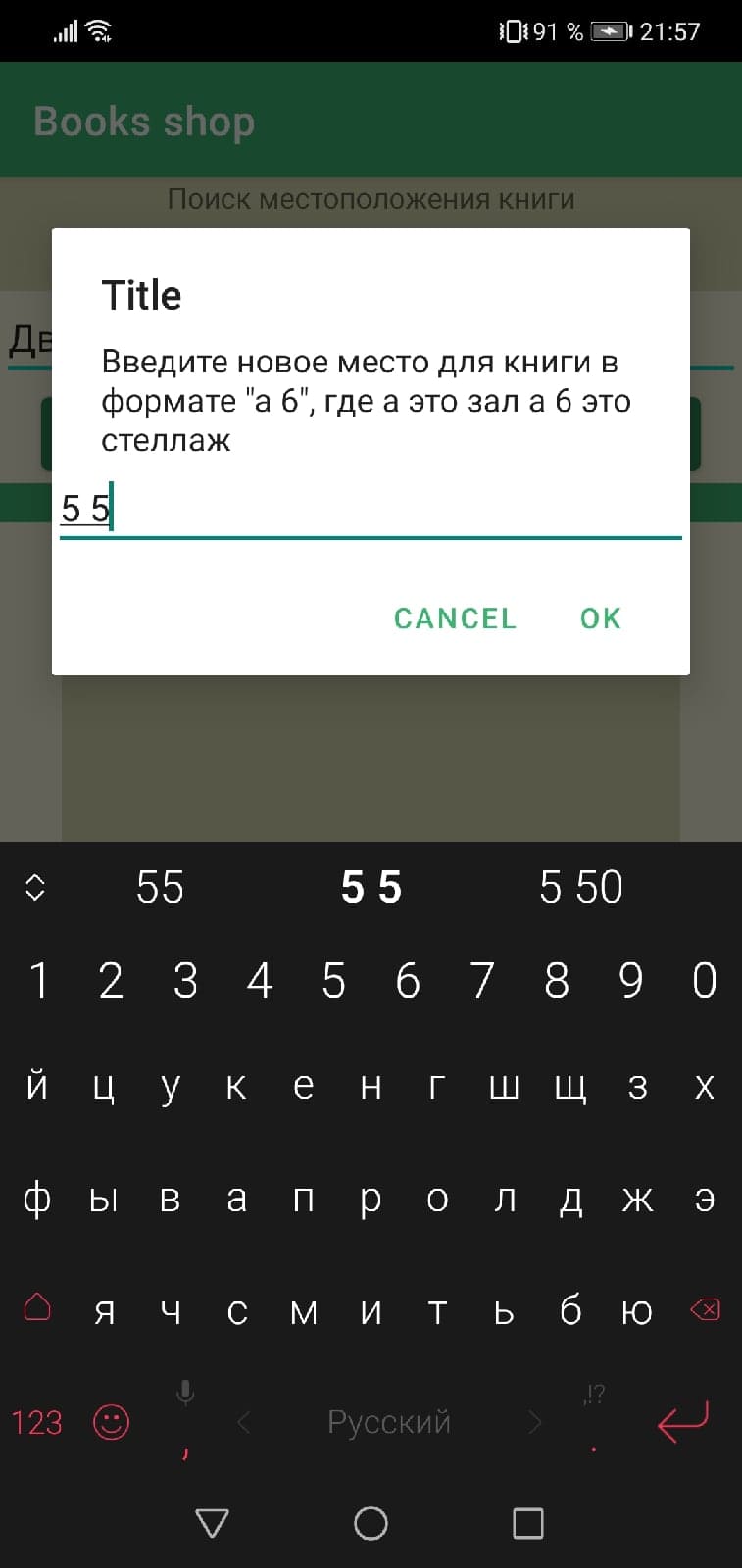


Рисунок 27-Диалог при изменении местоположения книги.

Благодаря адаптеру, данные измененные в базе данных и списке в приложении автоматически изменяются и на listview.

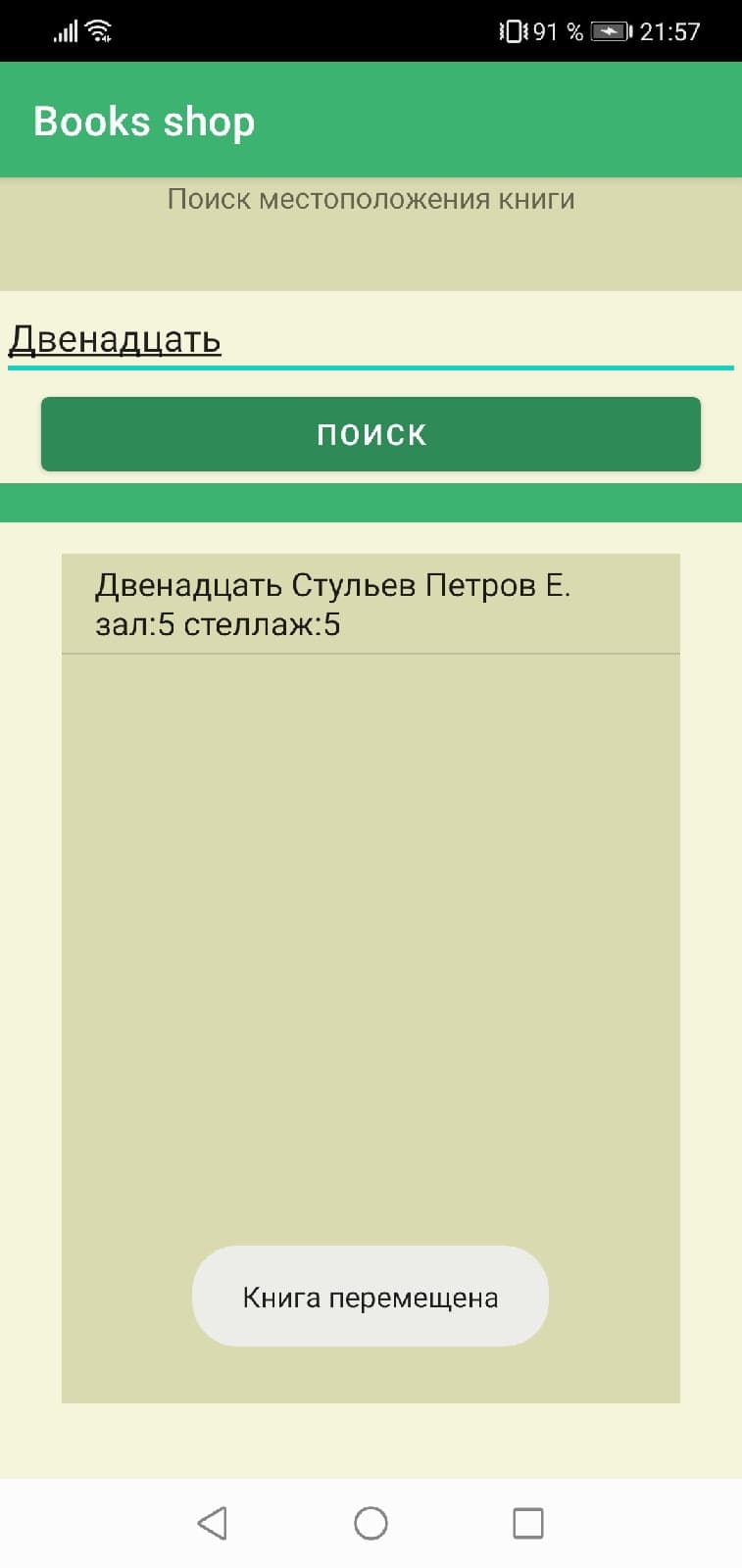


Рисунок 28-Экран с измененным местоположением книги.

При нажатии на книгу и в последующем диалоге нажатии на кнопку «продать книгу» появляется дополнительный диалог с информацией о продаже: дата продажи, соответствующая реальному времени, цена и id продаваемой книги. Данный диалог имеет две кнопки: «отмена» и «продать», при нажатии на первую, продажи не происходит, и книга остается в списке доступных. При нажатии на вторую, происходит обращение к базе данных и удалении ее из доступных книг. Так же создается элемент с ключом «book\_sale» и создается запись о продаже книги, в которой содержится дата, id работника, его фамилия, название проданной книги и ее цена. Так же работнику зачисляется плюс одна продажа, которая увеличивает его зарплату на какое-то конкретно заданное число.

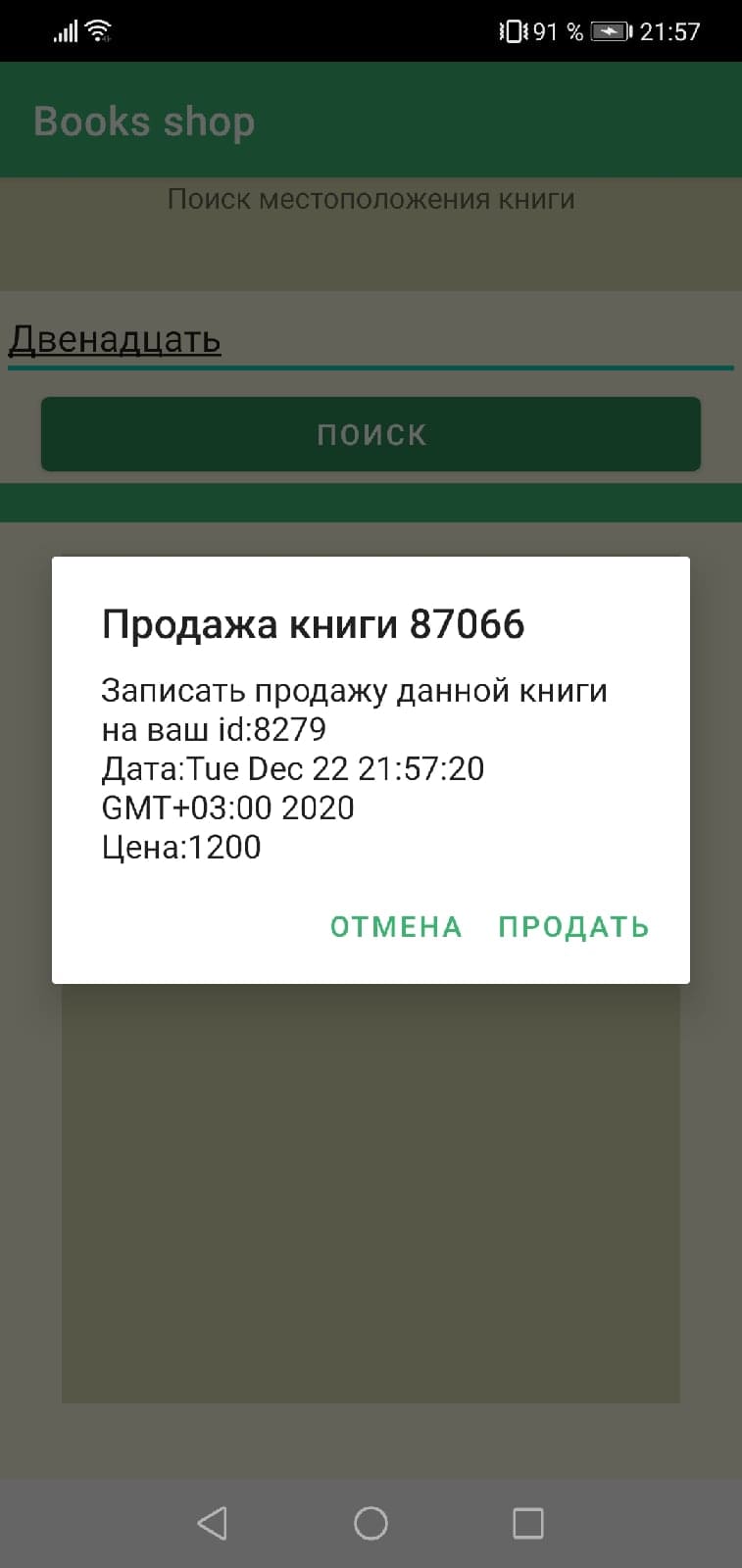


Рисунок 29-Экран с измененным местоположением книги.

После продажи книги, она пропадает из listview и из базы данных, а запись о ее продаже сохраняется под ключом «book\_sale».

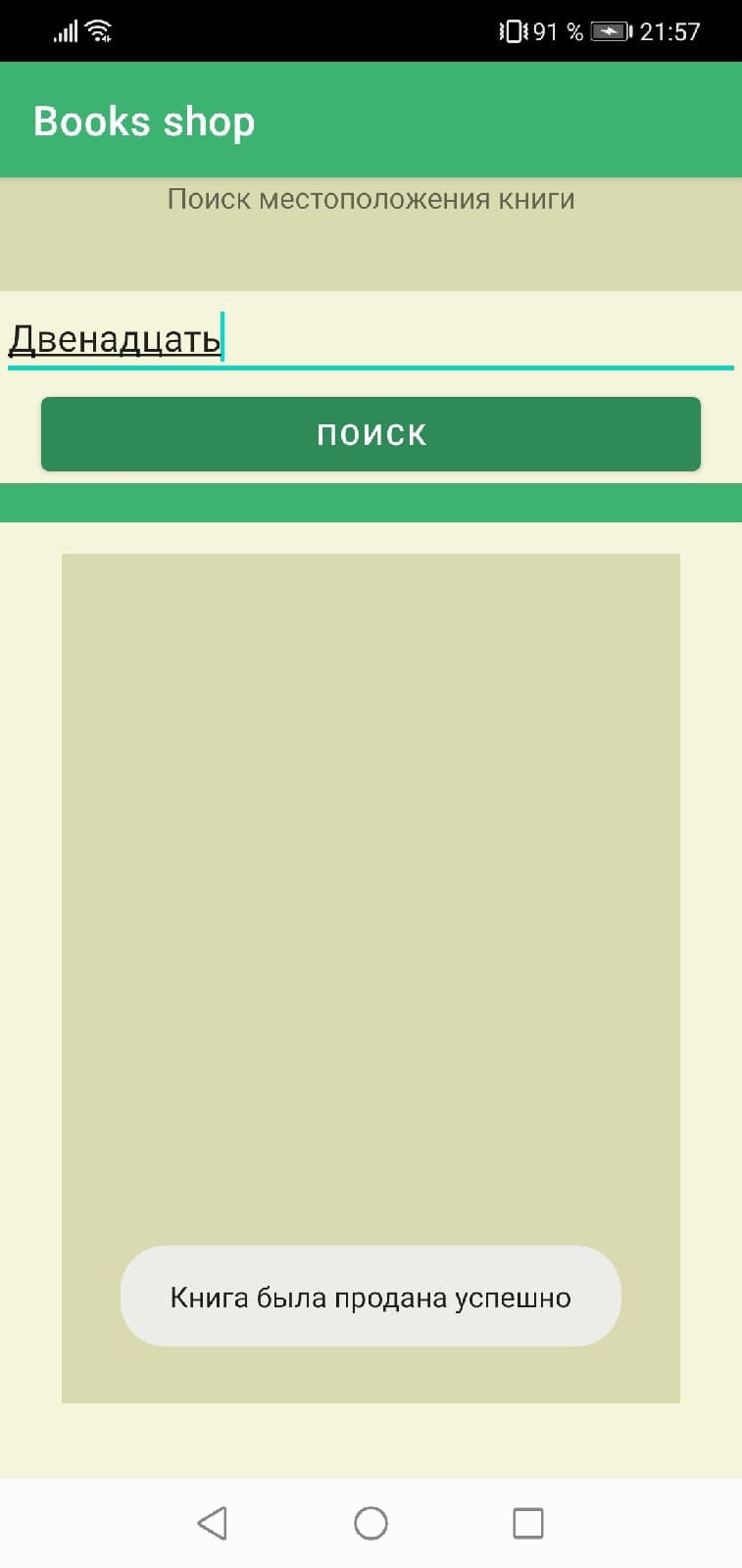


Рисунок 30-Экран поиска книги после покупки.

Далее, чтобы получить информацию о пользователе переходим в главное меню и нажимаем на кнопку «Зарплата». Мы попадаем на экран, содержащий полную информацию о пользователе: его id, имя и фамилию, дату получения зарплаты и сегодняшнюю дату, конечную зарплату и список продаж, который находится в listview и выводится в него через адаптер. Получается этот список путем обращения к базе данных и получения элементов по ключу «book\_sale». Выводятся только те продажи, которые совершил данный пользователь. Если день зарплаты еще не наступил, то выводится дата получения, если сегодня день зарплаты, то выводится уведомление, если день зарплаты уже наступил, то выводится дополнение про получение в следующем месяце.

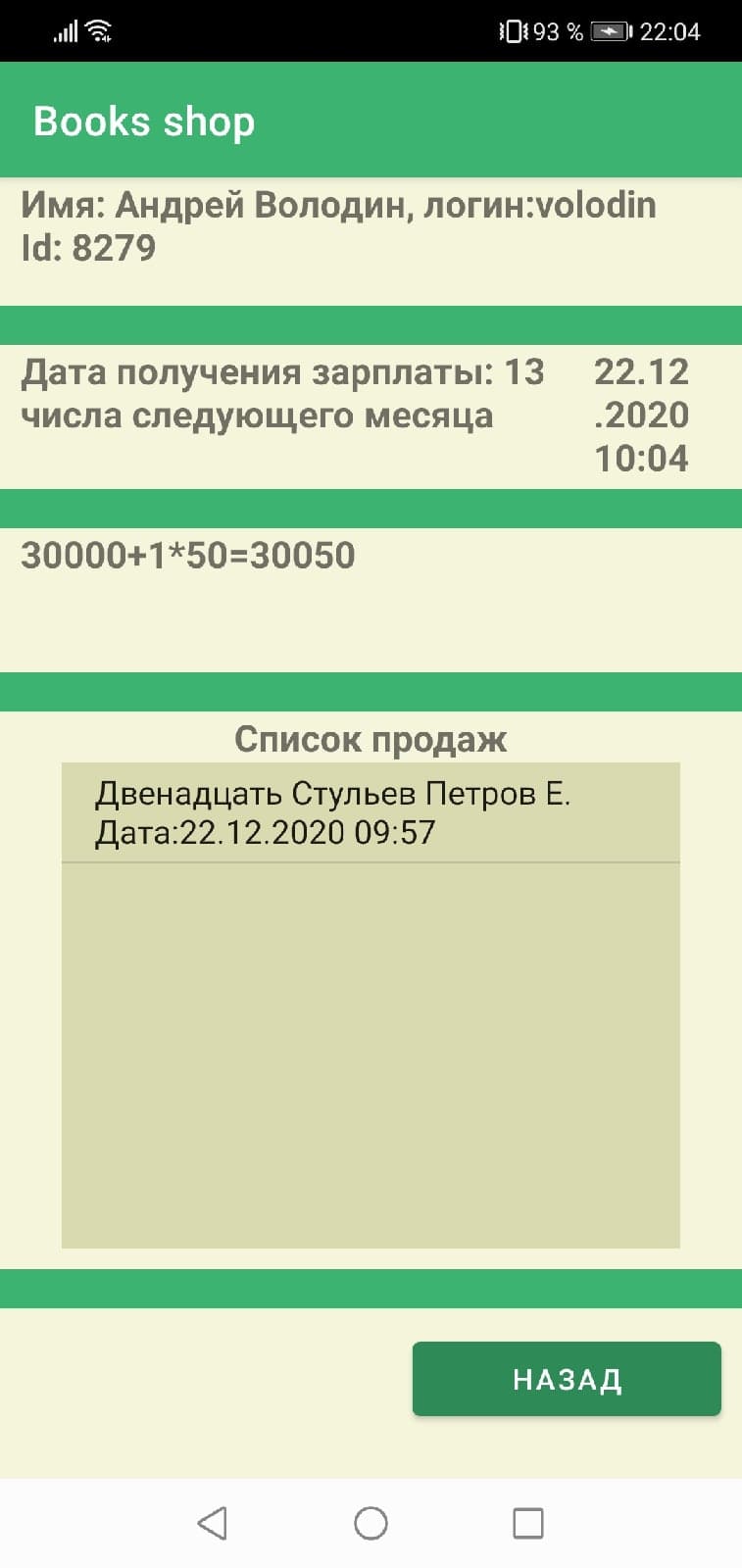


Рисунок 31-Экран информации о работнике.

Для создания запроса на новую книгу необходимо перейти в главное меню и нажать на кнопку «создать запрос». После этого совершается переход на новый экран, содержащий 4 поля для ввода данных: название книги, ее автор и издатель и доп. информация. Так же есть две кнопки, «назад» и «отправить запрос». При нажатии на первую пользователь перемещается в главное меню, при нажатии на вторую пользователь отправляет в базу данных новый элемент с ключом «request». Доступ к данным с этим ключом имеет менеджер.

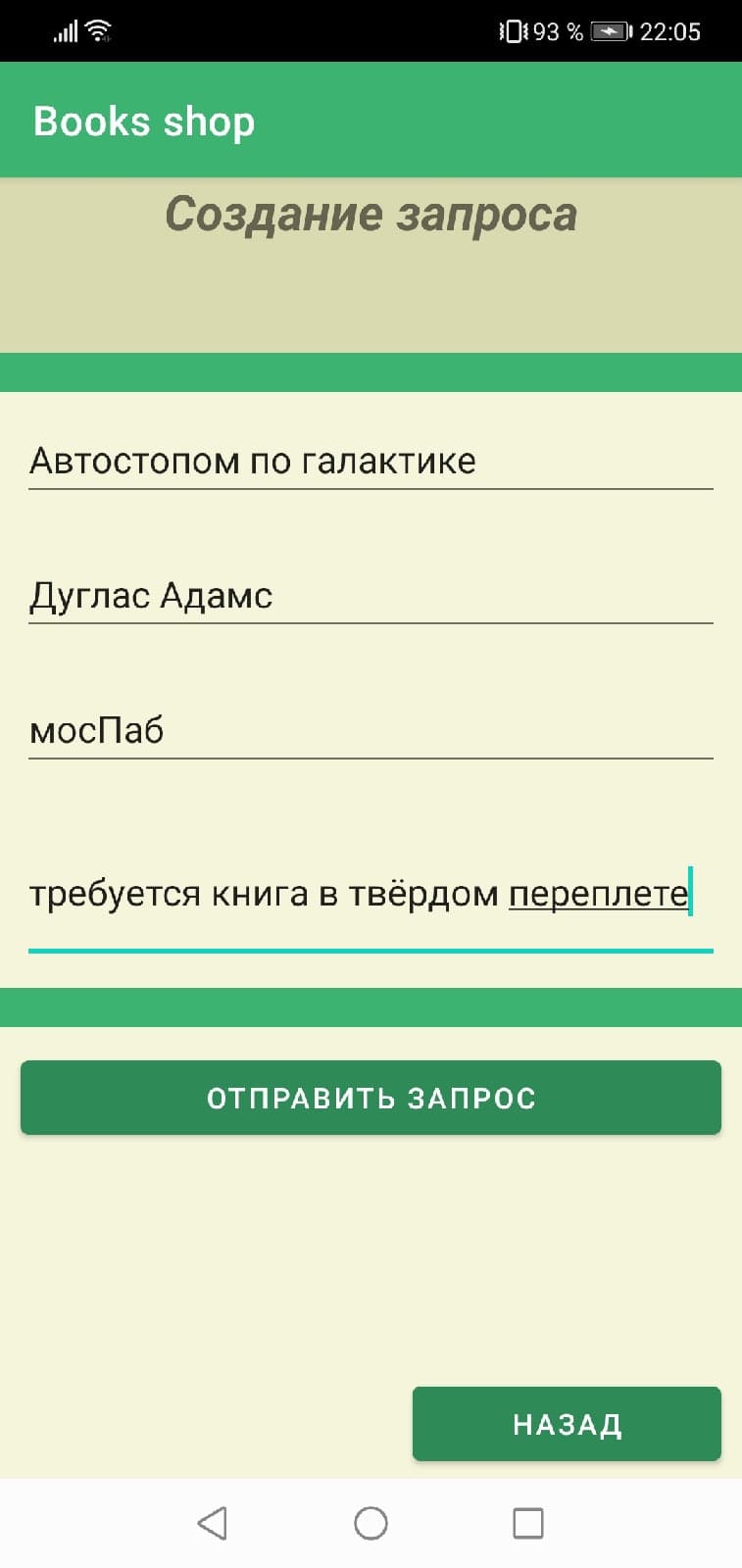


Рисунок 32-Экран отправления запроса.

После отправления поля очищаются и запись отправляется в базу данных.

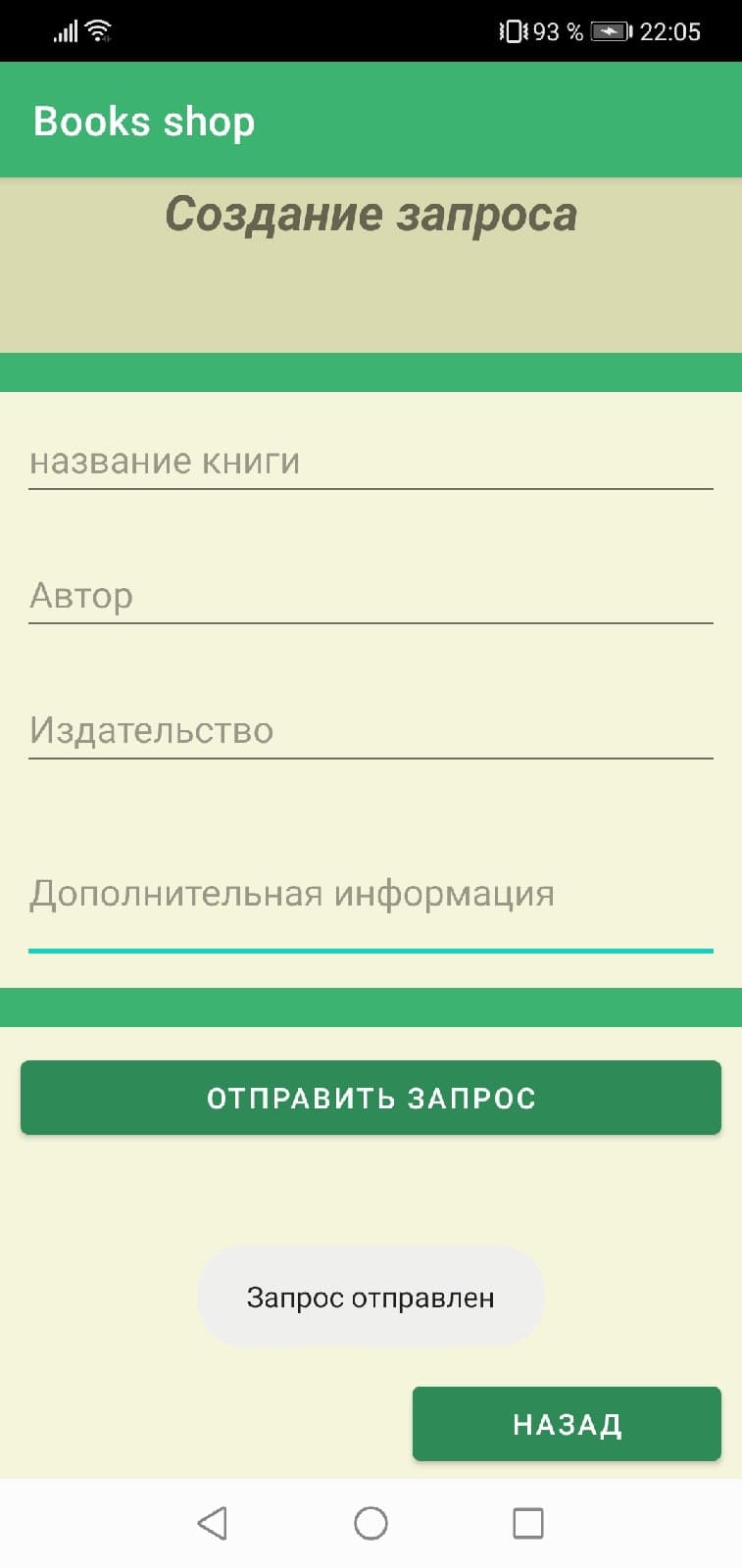


Рисунок 33-Экран отправления запроса после отправления запроса.

На данный момент, это все функции, доступные пользователю в режиме продавца. Для попадания в режим менеджера необходимо во время входа в приложение в полях логина и пароля указать слово «admin». После чего происходит переход на экран главного меню менеджера. Данный режим позволяет управлять данными путем их добавления и удаления. На экране главного меню располагаются три кнопки: «добавить нового работника», «добавить новую книгу» и «заявки на книги». Так же есть textview, в котором содержатся данные о зарегистрированных работниках. При нажатии на работника появляется диалоговое окно с его полной информацией.

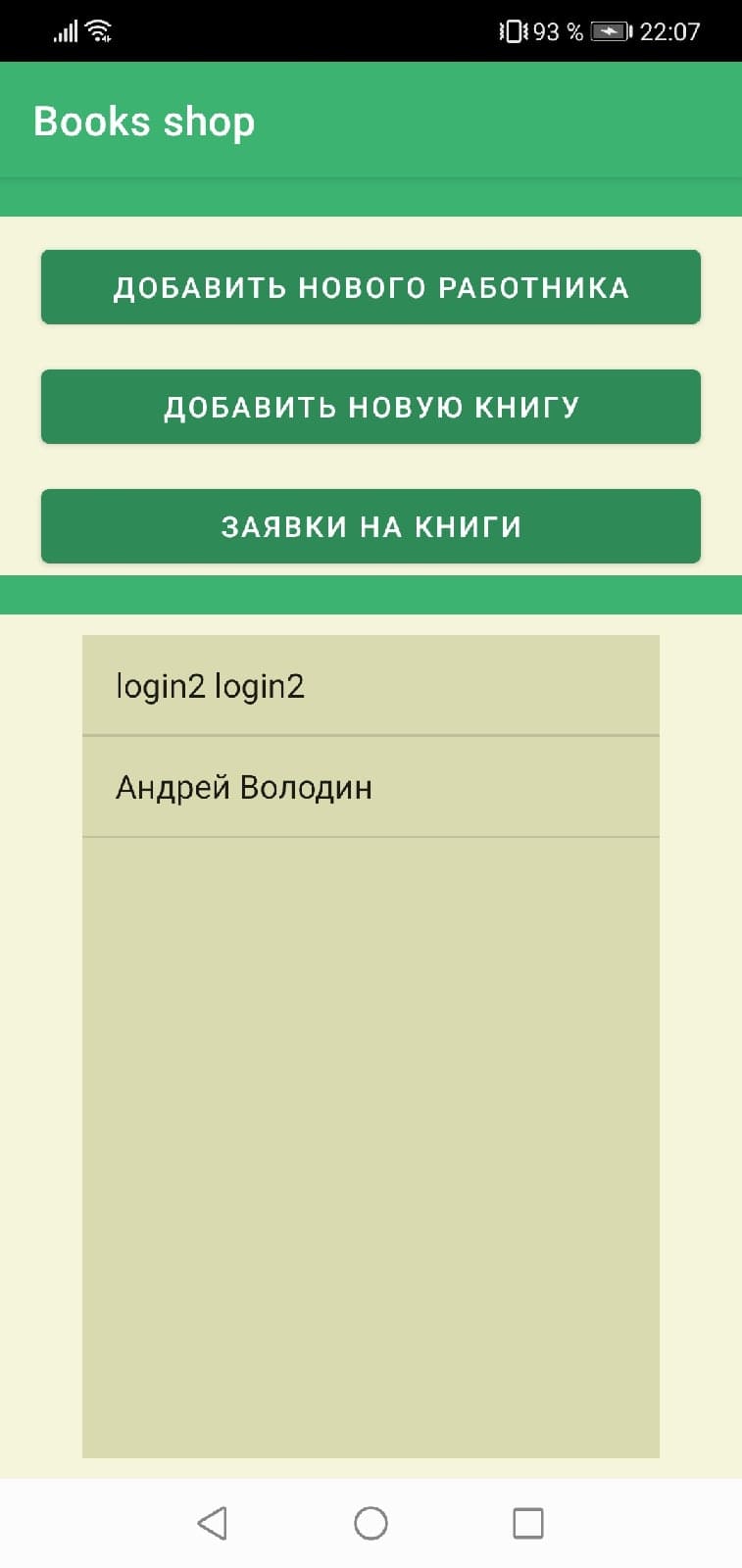


Рисунок 34-Главное меню режима менеджера.

При нажатии на кнопку «добавить сотрудника» происходит переход на экран добавления нового сотрудника. На нем располагаются 7 полей для ввода информации о сотруднике:

* имя
* фамилия
* логин
* пароль
* почта
* зарплата
* дата получения зарплаты

Под полями находится кнопка, при нажатии на которую происходит добавление работника в базу данных, и он может зайти в приложение используя установленный логин и пароль.

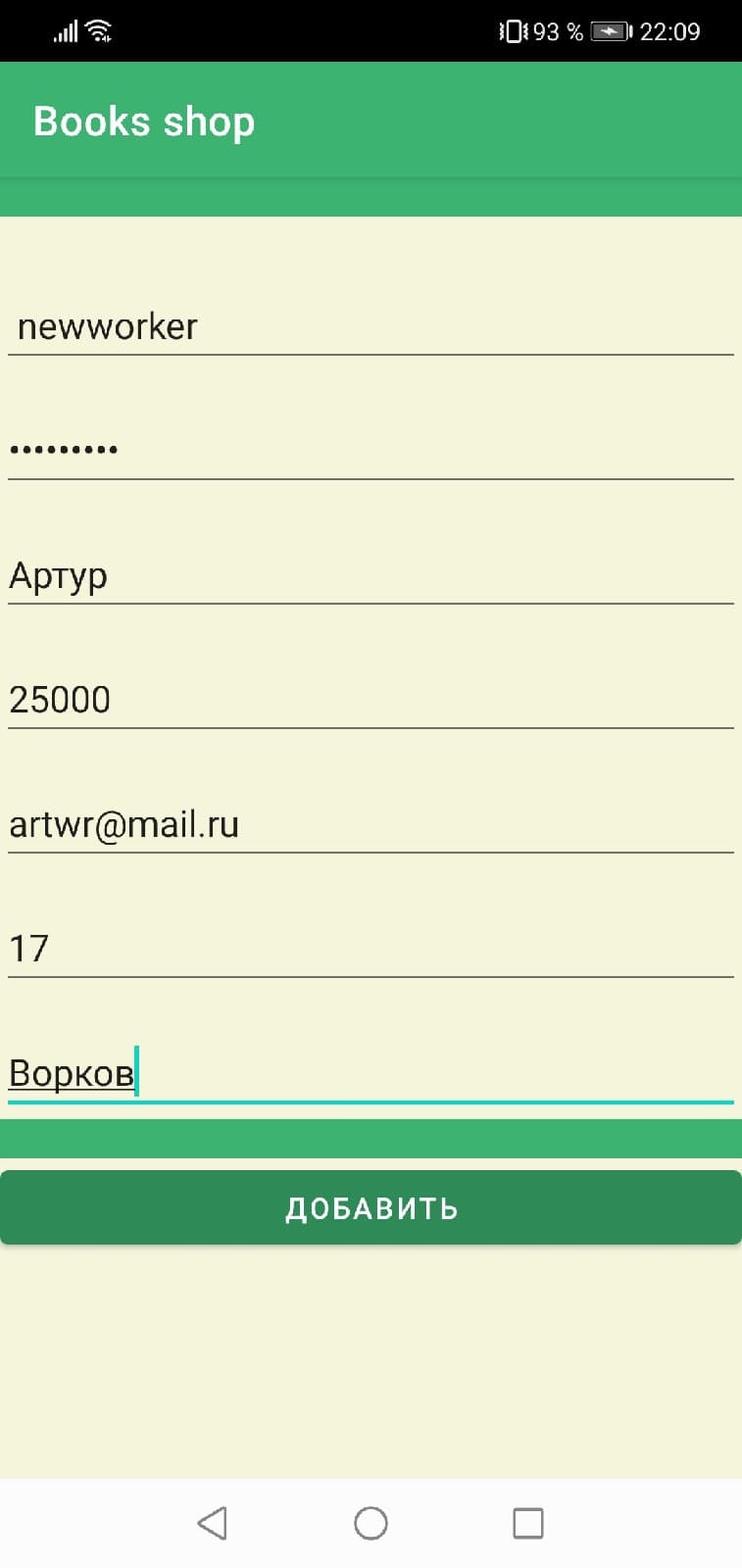


Рисунок 35-экран добавления нового сотрудника.

Новый сотрудник сразу появляется в listview и его можно сразу же проверить.

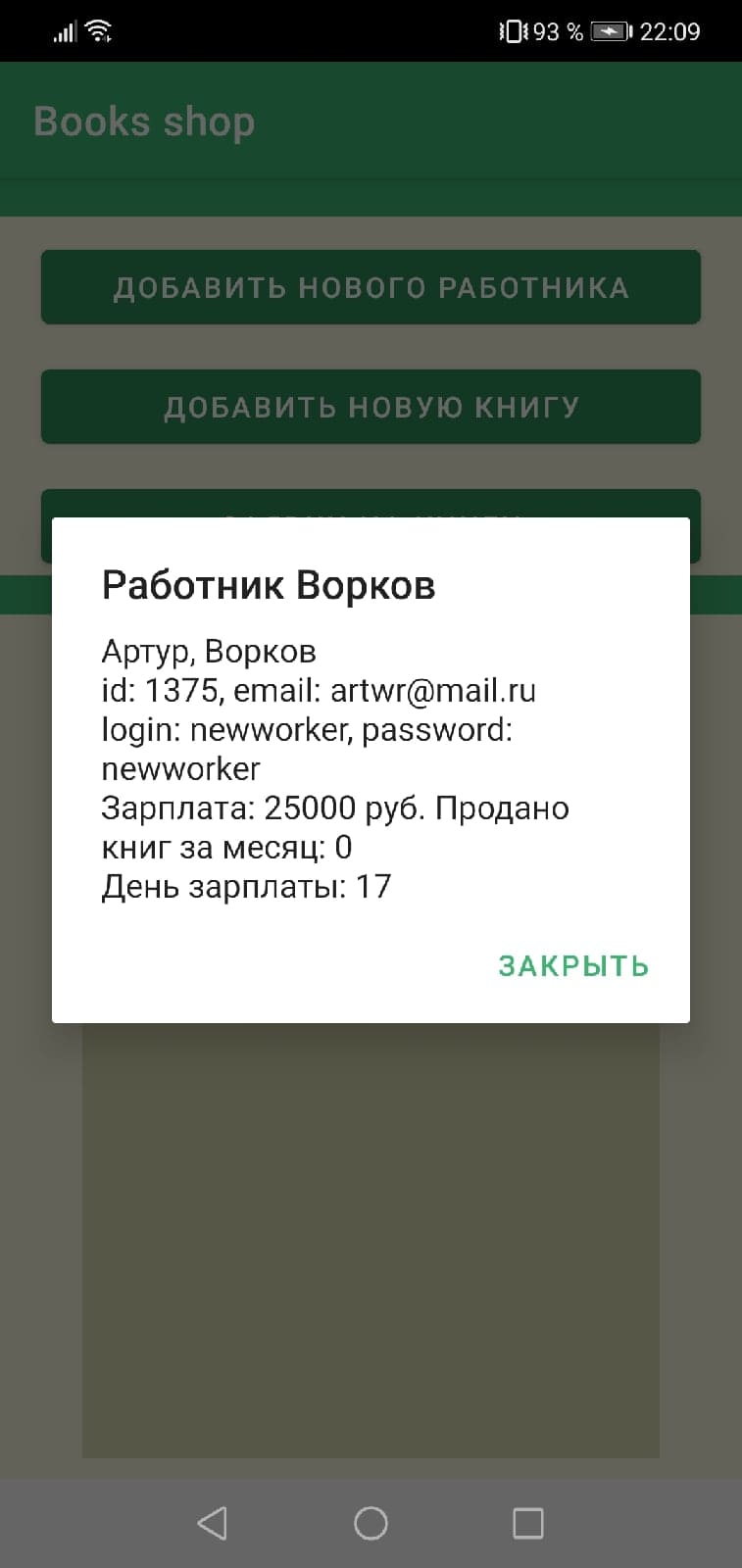


Рисунок 36-диалоговое окно с новым сотрудником.

При нажатии на кнопку «запросы» в главном меню пользователь переходит на экран с запросами, которые ему поступают от продавцов. На экране расположен listview, с кратким описанием запросов и кнопка для возвращения в главное меню. Запросы берутся из базы данных и через адаптер помещаются в listview, и при изменении данных в базе данных, так же изменяются и элементы в listview.

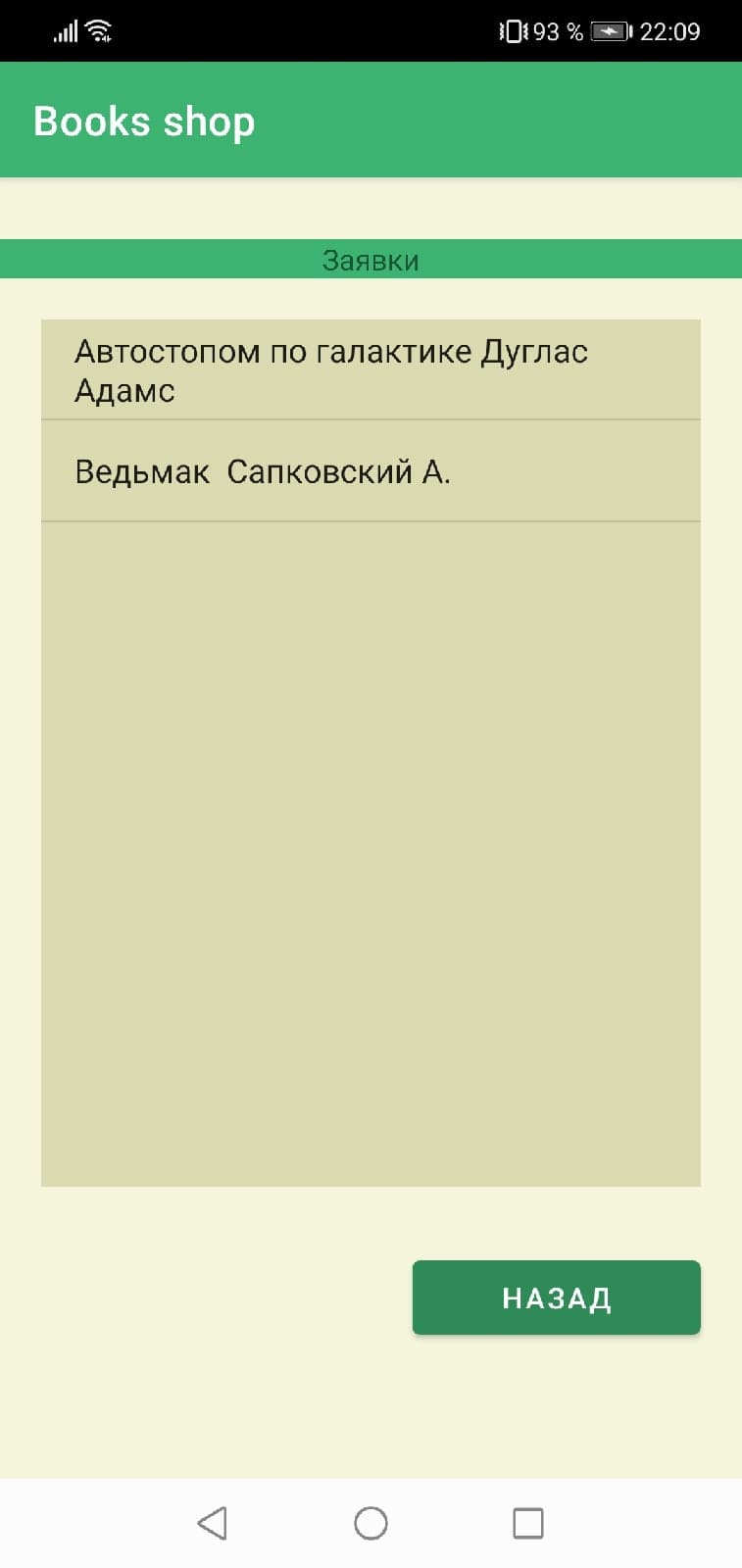


Рисунок 37-Экран с запросами.

При нажатии на запрос выводится диалоговое окно с полным описанием данного запроса: от кого он пришел, на какую книгу, от какого автора и издателя и когда был выполнен запрос. В диалоговом окне есть кнопка принять, которая удаляет запрос из базы данных. Если нажать вне диалогового окна, то оно скроется. Ответить на запрос определенному сотруднику через приложение нельзя.

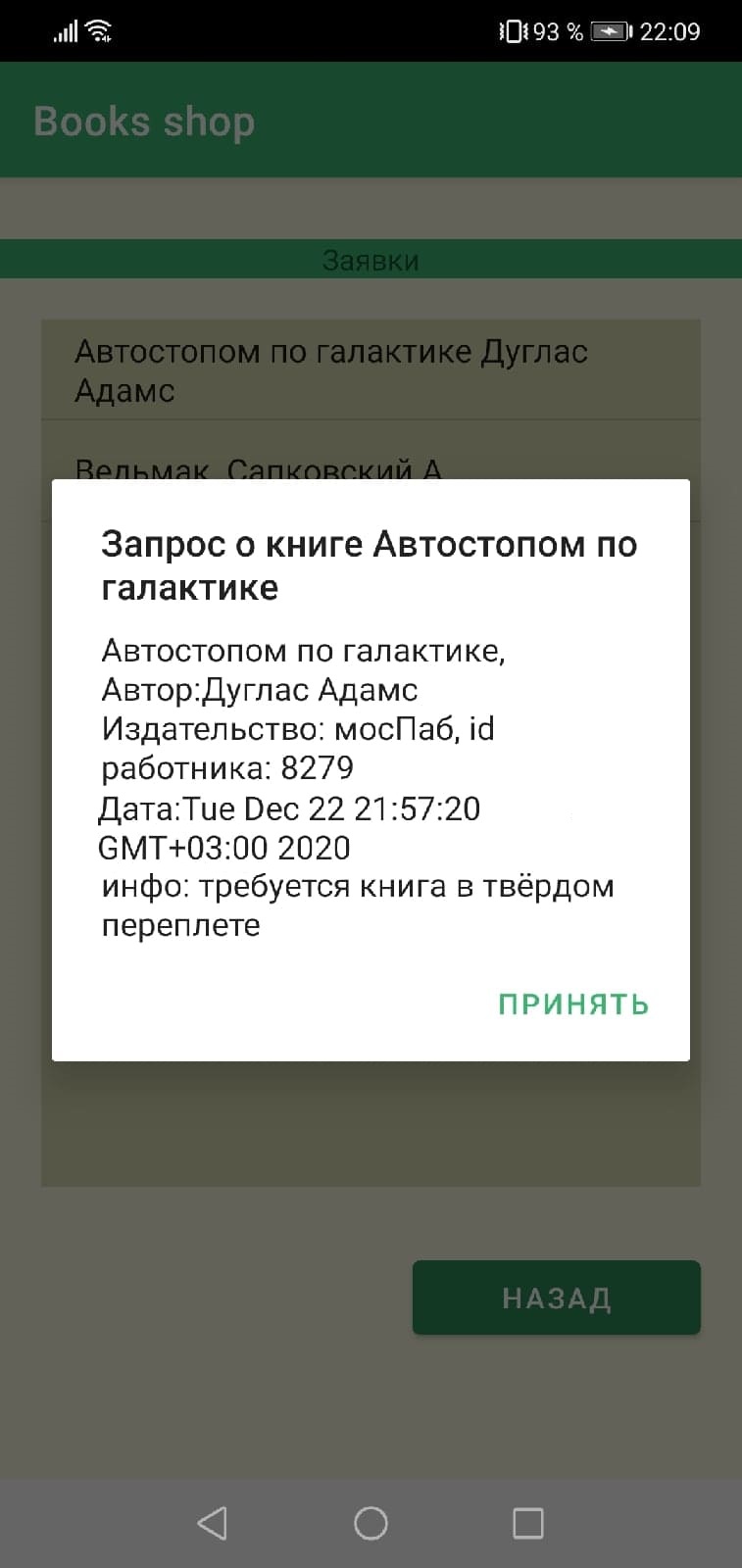


Рисунок 38-Диалоговое окно с запросом.

После нажатия на кнопку принять происходит удаление запроса без возможности его повторного получения. Если же выйти из диалога, не нажимая на кнопку принятия то запрос и информация о нем останутся до момента их удаления. Так как данные добавляются через адаптер то изменения в listview происходят, как только происходят изменения в базе данных.



Рисунок 39-Удаление запроса.

Последняя доступная функция у менеджера — это добавление книг. Для ее осуществления необходимо из главного меню нажать на кнопку «добавление книг» и перейти на экран с добавлением. На нем расположены 7 полей для ввода данных и кнопка, при нажатии на которую из введенных данных формируется объект класса Book и заносится в базу данных через соединение с ней. В поля для ввода необходимо ввести: название, автора, жанр, год издания, издательство, доп. информацию, зал, стеллаж и стоимость.

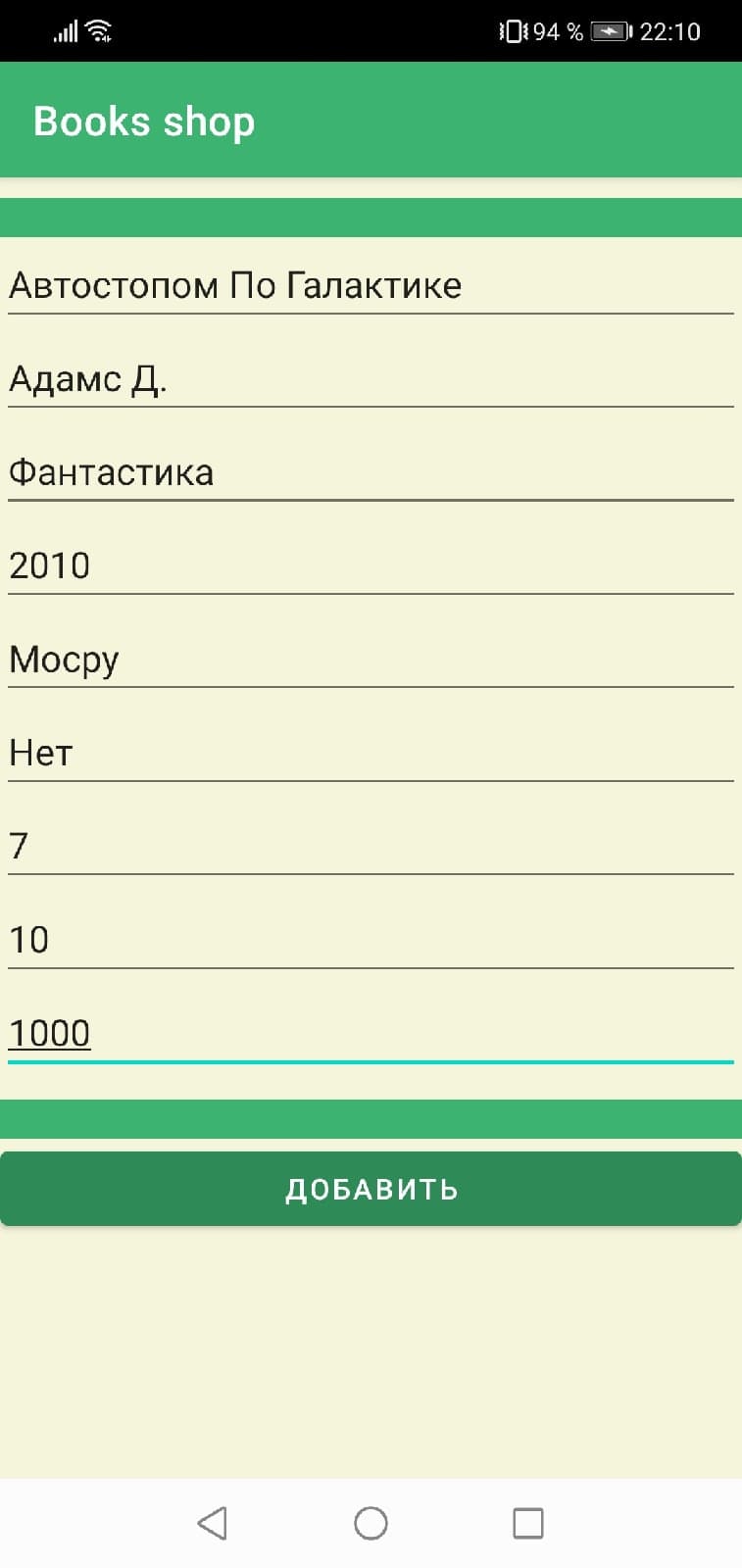


Рисунок 40-Добавление книги.

После создания книги, она появляется у всех остальных работников и доступна для продажи или ее перемещения. Для проверки ее наличия вернемся в режим продавца и проверим появление новой книги.

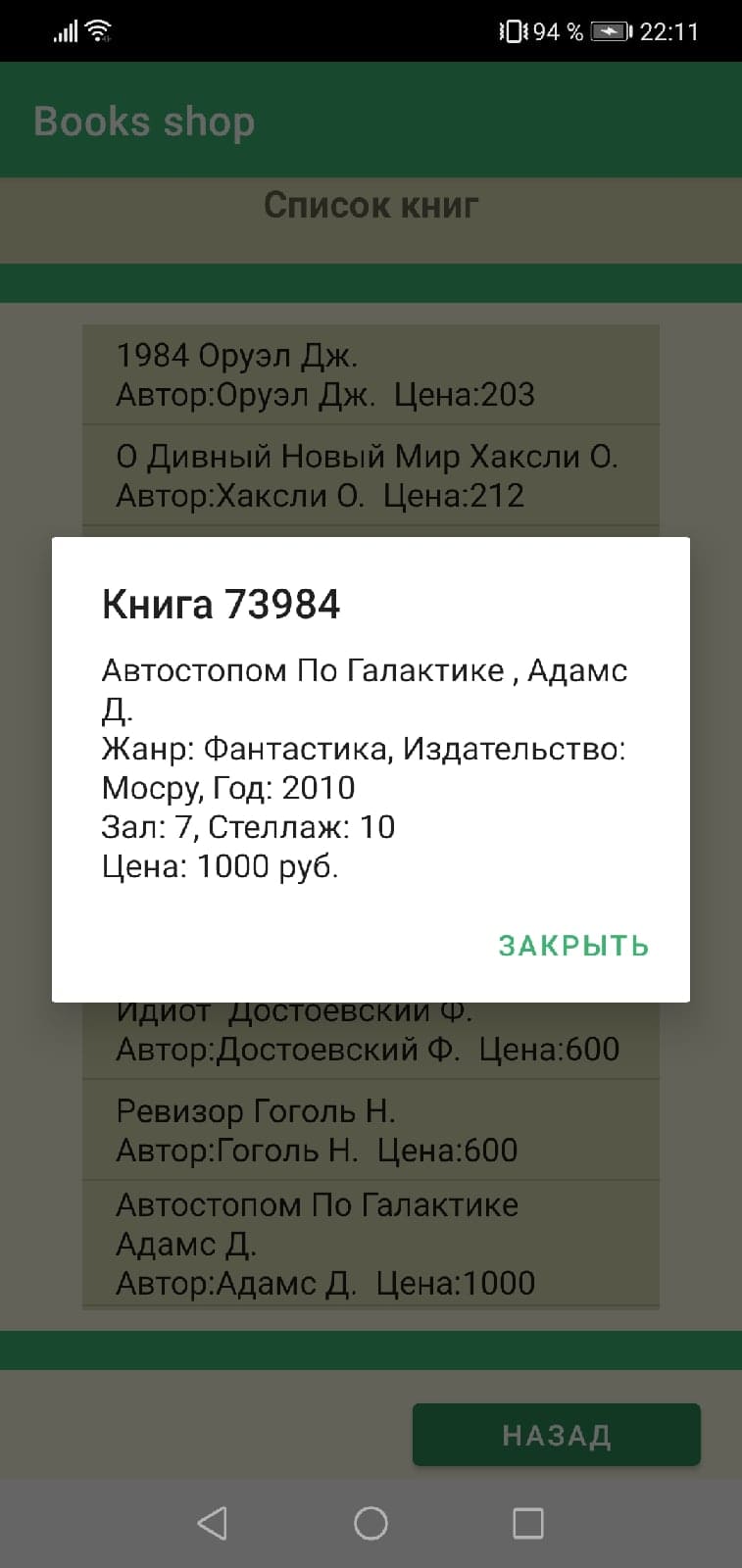


Рисунок 41-Новая книга.

## 5.2 Файловая структура приложения

Данное приложение содержит 16 классов, 12 из которых привязаны к различным activity. Классы, которые не привязаны к activity выполняют роль данных.

Класс Worker-класс пользователя приложения, он содержит данные для правильного отображения информации и правильного учета зарплаты.

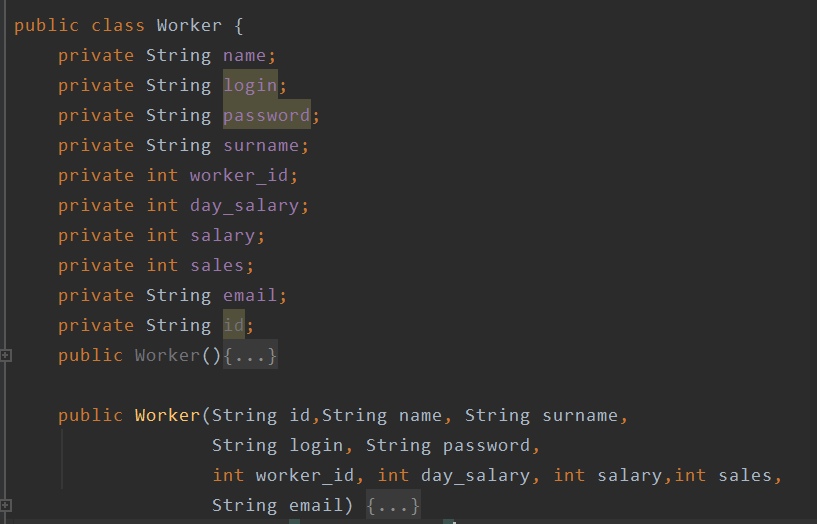


Рисунок 42-Класс Worker.

Класс Book-класс для хранения информации о каждой книге, отвечает за все данные и используется почти в каждом активити программы.

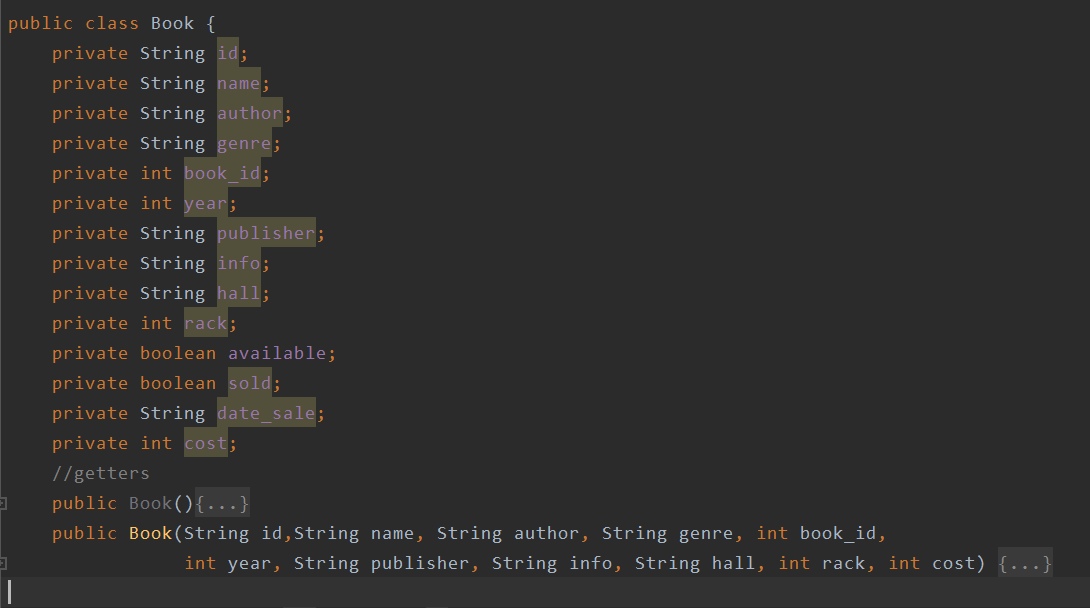


Рисунок 43-Класс Book.

Класс Sailinfo-класс, хранящий информацию про покупки и служит для вывода ее в listview.

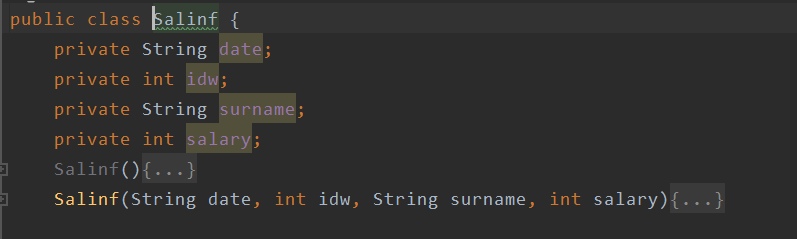


Рисунок 44-Класс Sailinfo.

Класс Request-класс для создания и хранения данных о запросах продавцов к менеджеру.

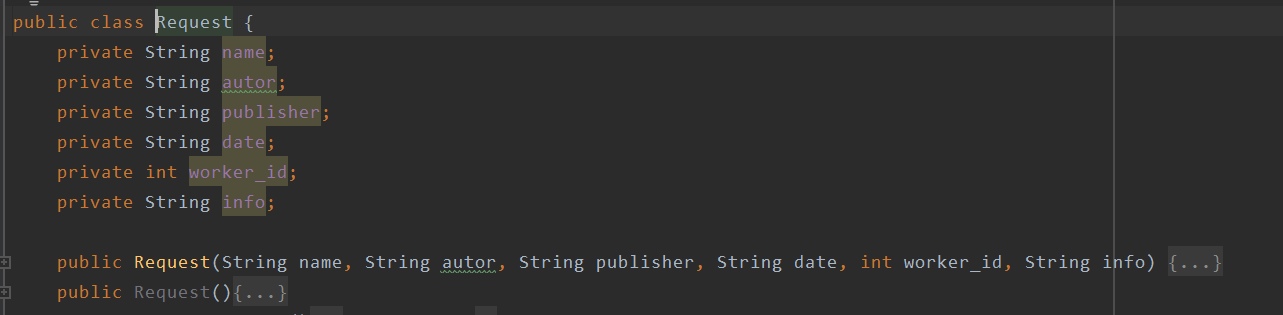


Рисунок 45-Класс Sailinfo.

Для всех классов необходимо было создать как конструкторы с параметрами, так и конструкторы без параметров, чтобы корректно обрабатывалась передача данных в базу данных. Кроме конструкторов в каждом классе присутствуют get и set методы, которые не попали на изображения.

Остальные классы привязаны к activity и обрабатывают различный функционал, связанный с ними.

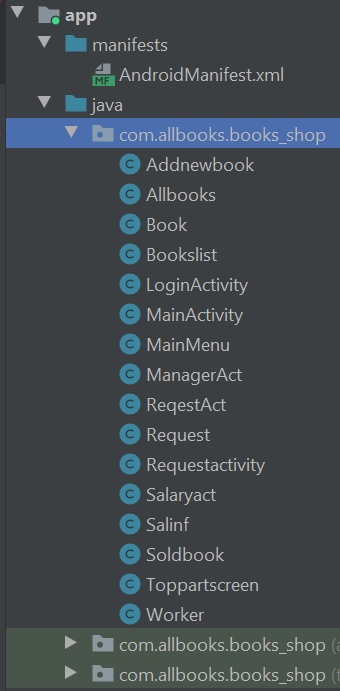


Рисунок 46-Файлы классов.

Addnewbook- класс для activity добавления новой книги в режиме менеджера. Отвечает за обработку полей ввода и добавления информации о новой книге в базу данных.

Allbooks- класс для activity вывода всех книг на экран. Отвечает за обработку нажатия кнопки выхода и вывода элементов базы данных на экран с помощью listview

Bookslist- класс для activity поиска и продажи определенной книги в режиме продавца. Отвечает за поиск книги в базе данных и обновление данных в случае продажи или перемещения книги.

LoginActivity- класс для activity для регистрации новых пользователей в режиме менеджера. Отвечает за обработку вводимых данных и добавления их в базу данных.

MainActivity- класс для activity входа в приложение. Отвечает за проверку правильности ввода логина и пароля и допускает сотрудника до работы.

MainMenu- класс для activity для главного меню режима продавца. Отвечает за обработку кнопок перехода на другие activity.

ManagerAct- класс для activity для главного меню режима менеджера. Отвечает за обработку кнопок перехода на другие activity.

ReqestAct- класс для activity получения и проверки запросов от сотрудников в режиме менеджера. Отвечает за вывод данных на экран и их изменение в базе данных в случае удаления.

RequestActivity- класс для activity для создания запроса менеджеру в режиме продавца. Отвечает за ввод данных и их отправку в базу данных.

Salaryact- класс для activity вывода полной информации о сотруднике.

Toppartscreen-Класс для фрагмента, который находится в главном меню режима продавца, отображает его id, зарплату и имя с фамилией.

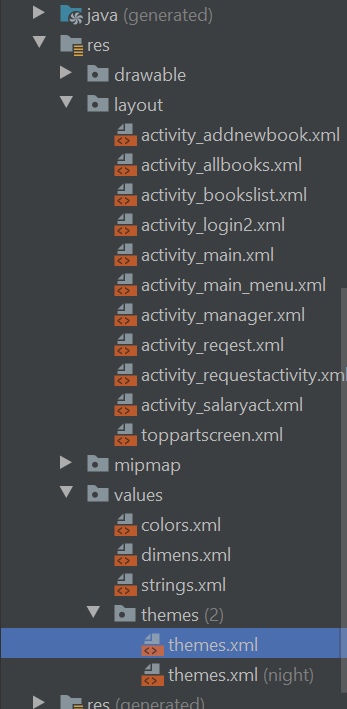


Рисунок 47-XML файлы разметки activity.

## 5.3 Подключение Firebase к проекту.

Одна из самых мощных функций платформы Firebase - это база данных в режиме реального времени: все выполняемые на нем операции записи мгновенно доступны для всех клиентов, которые ее наблюдают. С помощью такой базы данных вы можете создавать приложения, которые позволяют пользователям беспрепятственно переключаться с одного устройства на другое, а также мгновенно взаимодействовать с другими пользователями.

Для подключения Firebase к проекту нужно создать проект на самом сервере Firebase, затем ввести необходимые данные и скачать файл json, который нужно подключить к проекту.

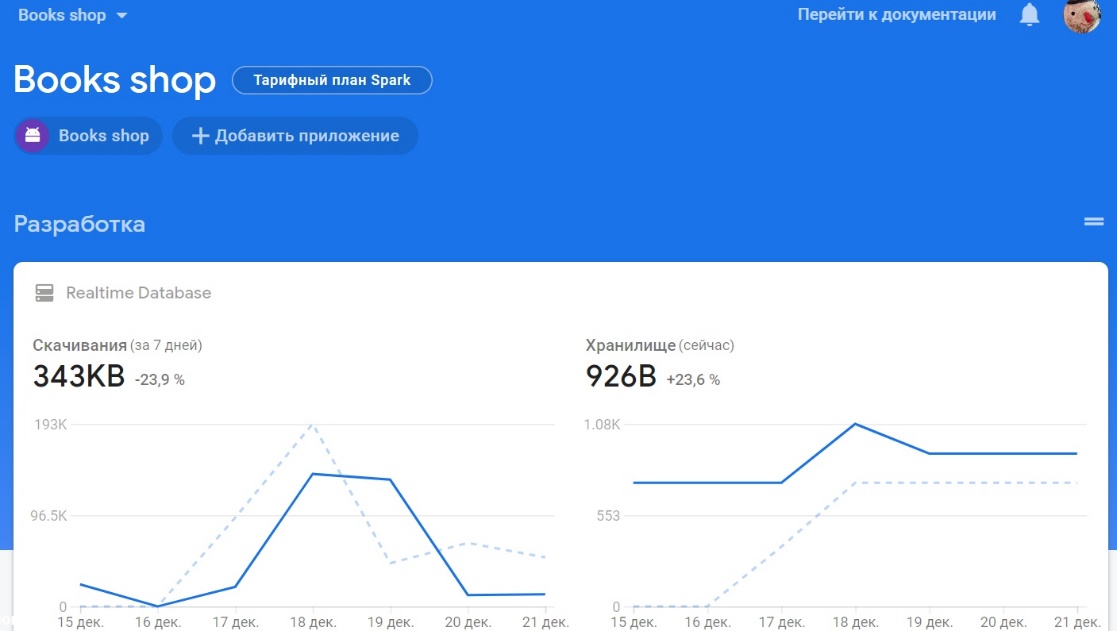


Рисунок 48-Проект на Firebase.

После чего нужно добавить SDK файл в приложение.



Рисунок 49-Добавление SDK.

# Заключение

Мобильные устройства в наше время — это вещь, без которой тяжело представить жизнь современного человека. Каждый человек сейчас имеет при себе сотовый телефон, который мощнее стационарных домашних компьютеров десятилетней давности. Использование мобильного телефона в работе — это следующий шаг в эволюции способов работы и рано или поздно, все будет завязано именно на мобильных устройствах. Книжная отрасль сейчас находится на подъеме, продажи растут из года в год, и люди все больше интересуются книгами. Хоть вся торговля и уходит в интернет, но обычные книжные магазины не уйдут никогда и всегда будут существовать в любом городе. Главной целью данной работы было определение проблем, связанных с работой книжных магазинов и создание решения этой проблемы с использованием мобильного приложения.

В ходе изучения отрасли была выявлена напрасная трата времени сотрудников и клиентов на поиски книг и хождение по магазину до компьютера, дабы получить информацию. Созданное в ходе работы приложение способно исправить эту проблему и повысить эффективность работы сотрудников и может привести к росту продаж. Само приложение не является полностью автономным продуктом и внедряться в реальное производство еще не может, так как требуется разработка приложения для компьютера, которое будет учитывать все продажи и удалять из базы данных купленные людьми книги. В дальнейшем данное приложение может быть улучшено и интегрировано в реальное производство.

В ходе данной работы были получены навыки работы с базой данных в реальном времени Firebase, с созданием приложений под операционную систему Android с помощью программы Android Studio, были получены навыки по анализу рынка и работе на языке программирования Java.

# Список используемой литературы

1. Android Studio Features. [Электронный ресурс] URL: https://developer.android.com/studio/features.html (дата обращения: 19.11.2020).

2. Appery.io: Enterprise Mobile App Builder & MBaaS. [Электронный

ресурс] URL: https://appery.io/ (дата обращения: 19.11.2020).

3. Download Android Studio and SDK Tools | Android Studio. [Электронный ресурс] URL: https://developer.android.com/studio/index.html (дата обращения: 25.11.2020).

4. Firebase a realtime database [Электронный ресурс] URL: https://firebase.google.com/?hl=RU (дата обращения:

20.11.2020).

5. Medianation: «Исследование: книжный рынок оцифровывается с трудом» [Электронный ресурс] URL:

https://medianation.ru/blog/issledovaniya/issledovanie-knizhnyy-rynok-otsifrovyvaetsya-s-trudom/ (дата обращения: 20.12.2020).

6. IntelliJ IDEA the Java IDE – JetBrains. [Электронный ресурс] URL:

https://www.jetbrains.com/idea/ (дата обращения: 15.12.2020).

7. IT Samsung academy. [Электронный ресурс] URL:

https://myitacademy.ru/edu/ (дата обращения: 15.12.2020).

8. MVP and MVC Architectures in Android. [Электронный ресурс]

URL: https://www.techyourchance.com/mvp-mvc-android-1/ (дата обращения:

21.11.2020).

9. Архитектура Android-приложений. Часть II – архитектурные

стили и шаблоны. [Электронный ресурс] URL: https://habrahabr.ru/post/140655/ (дата обращения: 29.11.2020).

10. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. 3-е издание. – СПб.: Издательство «Питер», 2016. – 512 с.

11. Введение в Firebase. [Электронный ресурс] URL: https://habr.com/ru/post/277941/ (дата обращения: 25.11.2020).

12. Дело модульбанка Устройство книжного рынка. [Электронный ресурс] URL:

https://delo.modulbank.ru/sales/bookmarket (дата обращения: 19.11.2020).

13. Дистанционное обучение РТУ МИРЭА. [Электронный ресурс] URL: https://online-edu.mirea.ru/my/ (дата обращения: 11.11.2020)

14. Книжный рынок: Состояние, тенденции и перспективы развития.

[Электронный ресурс] URL: https://www.bookind.ru/images/docs/Bookmarket-2018.pdf (дата обращения: 11.12.2020).

15. Мобильная разработка Макаревич А. [Электронный ресурс] URL: https://online-edu.mirea.ru/pluginfile.php?file=%2F574724%2Fmod\_resource%2Fcontent%2F0%2FЛК%20Функциональное%20и%20логическое%20программирование%20%28Миронов%20А.Н.%2C%20Макаревич%20А.Д.%2C%20Шешуков%20Л.С.%29.pdf (дата обращения: 11.12.2020).

16. Операции. [Электронный ресурс] URL: https://developer.android.com/guide/components/activities.html (дата обращения: 10.12.2020).

17. Основы создания приложений. [Электронный ресурс] URL:

https://developer.android.com/guide/components/fundamentals.html (дата обращения: 10.12.2020).

18. Колесниченко Д. Программирование для Android. 3-е

издание. – СПб.: Издательство «БВХ-Петербург», 2021. – 288 с.

19. Шилдт Г. Java 8. Полное руководство. – М.: Издательский дом

«Вильямс», 2017. – 1376 с.

20. Мартин Р. Чистый код: создание, анализ и рефакторинг. – СПб.: Издательство «Питер», 2021. – 464 с.

# Приложения

## Ссылки на проект

Ссылка на проект на github: https://github.com/kefalar/book\_shop.git

Ссылка на интерактивный макет: https://mockittapp.wondershare.com/app/c050499aa3e628b632eff4ee148fe4a8832f2cc7?simulator\_type=device&sticky