

АННОТАЦИЯ

Дипломный проект на тему мобильное приложение «Shabashka» состоит из программного средства, пояснительной записки и графической части.

Пояснительная записка к программному средству описывает процесс разработки, функционирования и тестирования приложения, предназначенного для глубокого анализа человеческого лица.

Shabashka это приложение, реализованное с использованием Java, Firebase и Android Studio. Оно позволяет искать подработку соискателю и публиковать работу работодателю.

Пояснительная записка выполнена на 40 листах, содержит четыре раздела и два приложения.

Графическая часть выполнена на двух листах, включает диаграмму вариантов использования и диаграмму сущность-связь.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	4
2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	5
2.1. Постановка задачи.....	5
2.2. Описание организации данных	5
2.3. Описание программных средств	7
2.4. Проектирование интерфейса приложения	8
2.5. Описание логической структуры программы	10
3. ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ	11
4. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА	21
5. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ... Ошибка! Закладка не определена.	
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	40
ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ.....	41
ПРИЛОЖЕНИЕ 1 (ОБЯЗАТЕЛЬНОЕ) – ЛИСТИНГ ПРОГРАММЫ	42

ВВЕДЕНИЕ

Задачами дипломной практики являются:

- изучение способов создания мобильных приложений;
- закрепление изученного материала;
- подготовка материалов к дипломному проекту.

Отчет по дипломной практике содержит следующие разделы:

1) «Назначение и область применения» – описывается назначение программы, краткая характеристика области применения программы. Проводится анализ предметной области и инфологическое проектирование разрабатываемой базы данных. Приводится обзор существующих подходов к решению задачи, их достоинства и недостатки, отличие разработанной программы от аналогов. Рассматриваются существующие аналоги.

2) «Технические характеристики» – рассматриваются следующие вопросы: требования к функциям (раздел «Постановка задачи»); формирование требований к системе в целом; требования к организации данных в проекте, описание их организации, структуры, построение концептуальной и логической модели данных, способа кодирования (при необходимости) и предварительной подготовки (раздел «Описание организации данных»); обоснование выбора программного обеспечения, используемого для разработки программного средства (раздел «Описание программных средств»); проектирование программного средства. (разделы «Описание физической структуры базы данных», «Проектирование интерфейса», «Описание логической структуры программы»).

3) «Тестирование и анализ полученных результатов» – описывается проверка работы программы на различных тестах и приводится анализ ожидаемых и полученных результатов, выводы по результатам тестирования.

4) «Руководство по использованию программного средства» – описывается последовательность действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведены описание функций, формата и возможных вариантов команд, с помощью которых пользователь управляет выполнением программы, а также ответы программы на эти команды.

5) «Технико-экономическое обоснование» – проводится оценка экономической эффективности разработки программы. Включает в себя расчет затрат на создание программного средства, прогнозируемую прибыль от его использования, оценку окупаемости проекта, а также анализ конкурентоспособности программного продукта на рынке.

Заключение содержит в себе перечисление основных результатов, характеризующих полноту решения поставленных задач и подводящих итог содержания дипломного проекта.

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

В данном дипломном проекте необходимо разработать мобильное приложение для поиска подработки. Данное программное средство (ПС) даёт пользователю возможность найти подработку по своему предпочтению, просмотрев нужную информацию о интересующей вакансии.

Аналогом данного проекта является программный продукт «Rabota.by». Преимуществом разрабатываемого программного средства над аналогом является простота пользовательского интерфейса и минимальное количество входных данных, что упрощает пользование приложением, также присутствует возможность просмотра контента при потере доступа к интернету, посредством кеширования информации.

Достоинствами данного продукта являются: простота интерфейса, возможность фильтрации информации, низкие требования к ресурсам устройства. Недостатками продукта являются: кеширование данных что может занимать память устройства.

Отличительной особенностью разрабатываемого в рамках дипломного проекта мобильного приложения является функционал, который акцентирован на поиске именно подработок.

В основе программного средства лежит идея приложения для поиска подработки. В данном проекте, эта механика обеспечивает удобство и индивидуальный подход к потребностям пользователя.

Данное программное средство рассчитано на людей различной возрастной категории, преимущественно студентам, которые желают быстро заработать.

Требования к аппаратным и операционным ресурсам для разработки:

- 1) Операционная система Android 7+.
- 2) Процессор MediaTek Helio A22.
- 3) Оперативная память 2GB.

2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1. Постановка задачи

Разработать и реализовать базу данных (БД) для хранения информации о пользователе, информации о вакансиях, а также информации о избранных вакансиях пользователя, статистике посещения и логах об ошибках. Разработать мобильное приложение для поиска подработки. Основной целью ПС является предоставление пользователю возможности найти подработку.

Для пользователя в программе необходимо осуществить отображение информации о вакансиях: заголовок вакансии, описание, место положения, является ли оплата почасовой и оклад. Также следует реализовать функцию фильтрации вакансий по заданным критериям.

В ПС предусмотреть возможность добавления и удаления вакансий из избранного.

В программе необходимо осуществить функции регистрации и авторизации пользователя.

При вводе данных обеспечить контроль вводимых данных, обработку исключений.

2.2. Описание организации данных

Данные для проекта необходимо хранить в базе данных. В таблице «Authentication» хранится информация о пользователях. Структура данных таблицы «Authentication» представлена в таблице 2.1.

Таблица 2.1 – Структура данных таблицы «Authentication»

Наименование столбца	Тип данных	Описание
Identifier	Текст	Хранить уникальную почту пользователя
Providers	Текст	Хранить способ входа
Created	Дата	Хранить дату регистрации пользователя
Signed In	Дата	Хранить дату последнего входа пользователя
User UID	Текст	Идентификатор пользователя

В таблице «Jobs» хранится информация о вакансиях. Структура данных таблицы «Jobs» представлена в таблице 2.2.

Таблица 2.2 – Структура данных таблицы «Jobs»

Наименование столбца	Тип данных	Описание
Description	Текст	Хранить описание вакансии
Hourly	Логический	Хранить является ли оплата почасовой
Location	Текст	Хранить локацию работы
Salary	Int	Хранить зарплату за выполненную работу
Title	Текст	Хранить заголовок подработки
UserId	Текст	Хранить идентификатор пользователя, который разместил вакансию
JobId	Число	Хранить идентификатор вакансии
Contacts	Текст	Хранить контакты к вакансии

В таблице «Favorites» хранится вся необходимая информация об избранных вакансиях пользователя. Структура данных таблицы «Favorites» в базе данных представлена в таблице 2.3.

Таблица 2.3 – Структура данных таблицы «Favorites»

Наименование столбца	Тип данных	Описание
UserId	Число	Хранить идентификатор пользователя
JobId	Число	Хранить идентификатор вакансии

В таблице «ErrorsLogs» хранится информация о ошибках приложения. Структура данных таблицы «ErrorsLogs» представлена в таблице 2.4.

Таблица 2.4 – Структура данных таблицы «ErrorsLogs»

Наименование столбца	Тип данных	Описание
Date	Дата	Хранить дату возникновения ошибки
Description	Текст	Хранить описание ошибки

В таблице «Statistics» хранится информация о статистике. Структура данных таблицы «Statistics» представлена в таблице 2.5.

Таблица 2.5 – Структура данных таблицы «Statistics»

Наименование столбца	Тип данных	Описание
UserId	Число	Хранить идентификатор пользователя
VisitsCount	Число	Хранить количество посещений приложения

Входными данными программы являются: телефон и почта пользователя.

Выходными данными программы являются: информация по запросу пользователя.

2.3. Описание программных средств

Для разработки в рамках дипломного проектирования ПС была выбрана среда разработки Android Studio с использованием языка программирования Java. БД реализована с помощью Firebase .

Java – это высокоуровневый, строго типизированный, объектно-ориентированный язык программирования с кроссплатформенной поддержкой.

Плюсы языка Java:

- 1) Кроссплатформенность.
- 2) Высокая производительность.
- 3) Богатая экосистема.

Минусы языка Java:

- 1) Сложность для новичков.
- 2) Большой объем кода.
- 3) Потребление памяти.

Android Studio – это официальная интегрированная среда разработки (IDE) для создания приложений под Android, разработанная Google на базе IntelliJ IDEA.

Плюсы Android Studio:

- 1) Официальная поддержка Google.
- 2) Мощный редактор кода.
- 3) Гибкая система сборки (Gradle).

Минусы Android Studio:

- 1) Требовательность к ресурсам.
- 2) Долгая сборка проектов.
- 3) Сложность для новичков.

База данных для курсового проекта разрабатывалась на платформе Firebase. Данная платформа поддерживает по SQL базы данных.

Плюсы Firebase:

- 1) Реальное время (Firestore, Realtime DB).
- 2) Аутентификация.

Минусы Firebase:

- 1) Привязка к Google Cloud.
- 2) Ограничения бесплатного тарифа.

2.4. Проектирование интерфейса приложения

В данном проекте программный продукт должен быть реализован восьми страницах: главная страница, страница профиля, страница с избранным, страница добавления вакансии, страница настроек, страница подробного рассмотрения вакансии страница авторизации, страница регистрации.

На главной странице пользователь может просмотреть актуальные предложения, что изображено на рисунке 2.1.

Элементы управления главной страницы представлены в таблице 2.7.

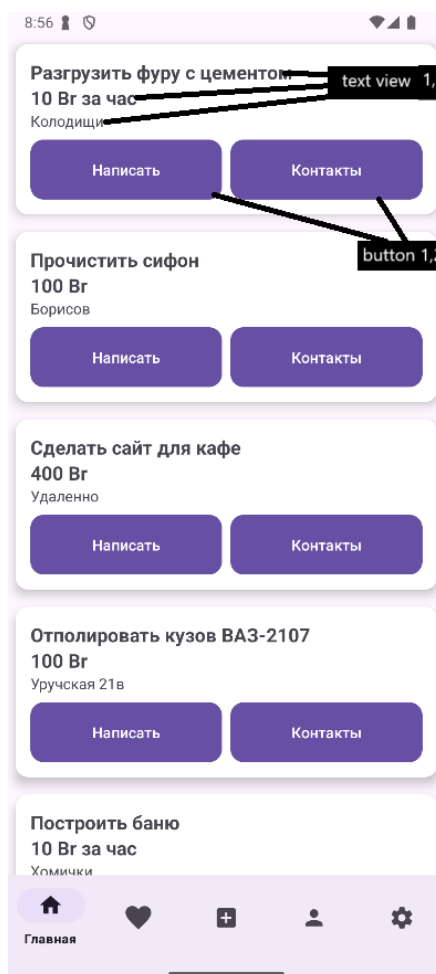


Рис. 2.1. Главная страница

Таблица 2.6 – Элементы управления главной страницы

Элементы управления	Свойства	Назначение
Text view 1	Type=bold	Заголовок
Text view 2	Type=bold	Размер оплаты
Text view 3	Type=normal	Локация

Форма регистрации изображена на рисунке 2.2.

Элементы управления формы регистрации представлены в таблице 2.7.

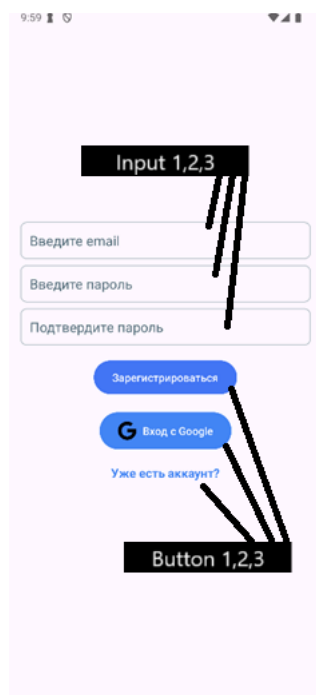


Рис. 2.2. Форма регистрации

Таблица 2.7 – Элементы управления формы регистрации

Элементы управления	Свойства	Назначение
Input 1	Hint=Введите email	Поле ввода почты
Input 2	Hint=Введите пароль	Поле для ввода пароля
Input 3	Hint=Подтвердите пароль	Поле для подтверждения пароля
Button 1	Text=Зарегистрироваться	Кнопка для регистрации
Button 2	Text=Вход с Google	Кнопка для входа с гугл
Button 3	Text=Уже есть аккаунт?	Кнопка для входа

Форма авторизации изображена на рисунке 2.3.

Элементы управления формы авторизации представлены в таблице 2.8.



Рис. 2.3. Форма авторизации

Таблица 2.8 – Элементы управления формы авторизации

Элементы управления	Свойства	Назначение
Input 1	Hint=Введите email	Ввод почты
Input 2	Hint=Введите пароль	Поле для ввода пароля
Button 1	Text=Войти	Кнопка авторизации
Button 2	Text=Вход с Google	Вход с гугл
Button 3	Text=Нет аккаунта?	Кнопка для перехода на форму ркгистрации

2.5. Описание логической структуры программы

В программе реализованы следующие функции:

- 1) login – авторизация пользователя.
- 2) toggleFragments – переключение между фрагментами.
- 3) updateReview – обновление данных на главной странице.

3. ТЕСТИРОВАНИЕ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

Тестирование программного обеспечения представляет собой важнейший этап в процессе разработки, направленный на обеспечение качества и надёжности создаваемого программного продукта. Этот процесс включает в себя всестороннее исследование и испытание работы программы с целью выявления возможных ошибок, несоответствий или сбоев. Основная задача тестирования — убедиться в том, что фактическое поведение программы совпадает с ожидаемым при различных условиях использования, что достигается путём проведения набора тестов, подобранных по определённым критериям. Таким образом, тестирование позволяет не только обнаружить ошибки на ранних этапах, но и повысить уверенность в корректной работе программного обеспечения в реальных условиях эксплуатации.

Метод «черного ящика». Основан на принципе «вход-выход». Программе подаются некоторые данные на вход и проверяются результаты, в надежде найти несоответствия. При этом как именно работает программа, считается несущественным.

Этапы процесса тестирования:

1) Проверка приложения в нормальных условиях. Представляет собой тестирование с исходными данными, требуемыми для нормального функционирования программы.

2) Проверка в экстремальных условиях. Представляет собой тестирование программы с граничными и неверными исходными данными. Программа должна работать с граничными исходными данными и отвергать неверные исходные данные.

В проекте использовался метод тестирования «черного ящика». Программа была протестирована на следующих тестах:

- 1) Тест №1. Проверка на запуск мобильного приложения.
- 2) Тест №2. Регистрация пользователя.
- 3) Тест №3. Авторизация пользователя.
- 4) Тест №4. Добавление в избранное.
- 5) Тест №5. Рейтинговая система.
- 6) Тест №6. Поиск вакансии по названию.
- 7) Тест №7. Поиск вакансии по местоположению.
- 8) Тест №8. Создание предложения о работе.

Тест №1. Показал, что при входе в приложение оно должно запускаться и отображать страницу авторизации, что представлено на рисунке 3.1.

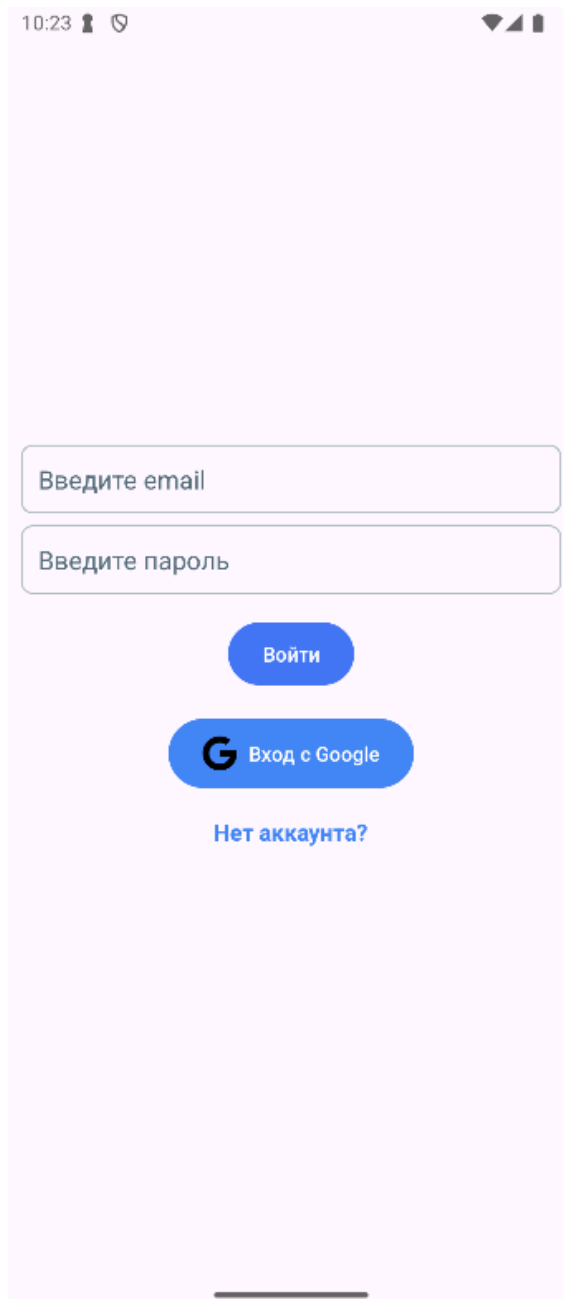


Рис. 3.1. Открытие приложения

Тест №2. Показал, что при нажатии на кнопку «Регистрация», программа регистрирует пользователя в системе и войдет в его аккаунт, что показано на рисунке 3.2.

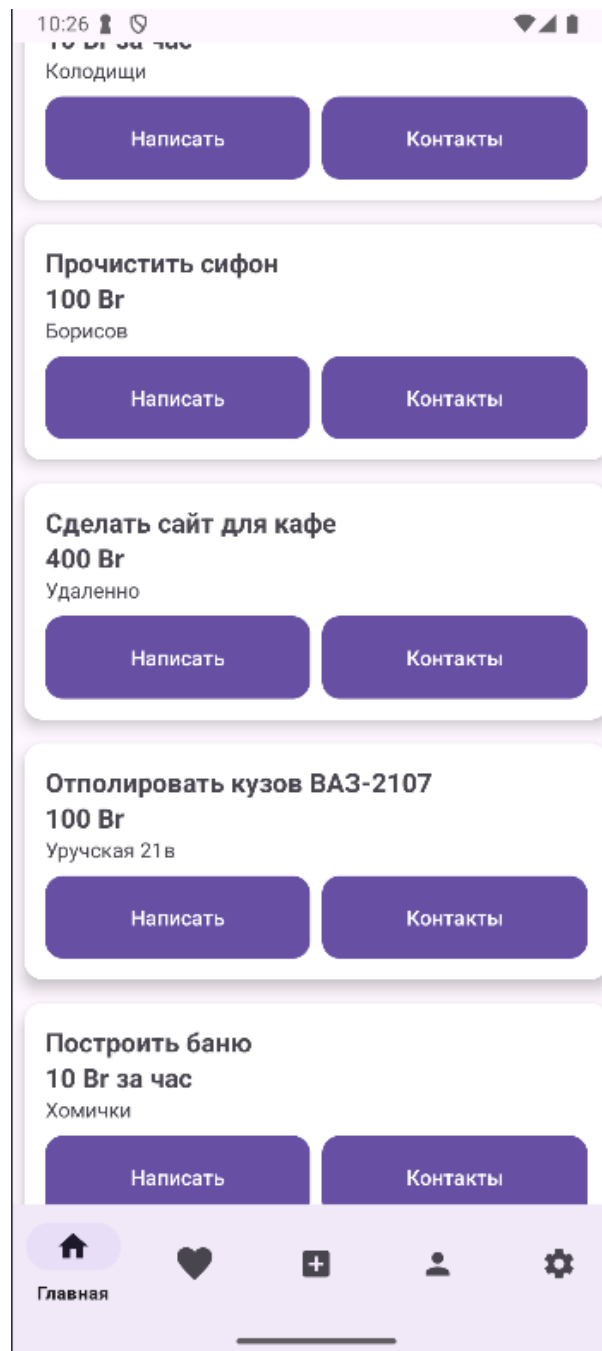


Рис. 3.2. Регистрация пользователя

Тест №3. Показал, что при неверно заполненных полях формы авторизации и нажатии на кнопку «Войти» программа выдаст ошибку и не войдет в аккаунт пользователя, что показано на рисунке 3.3.

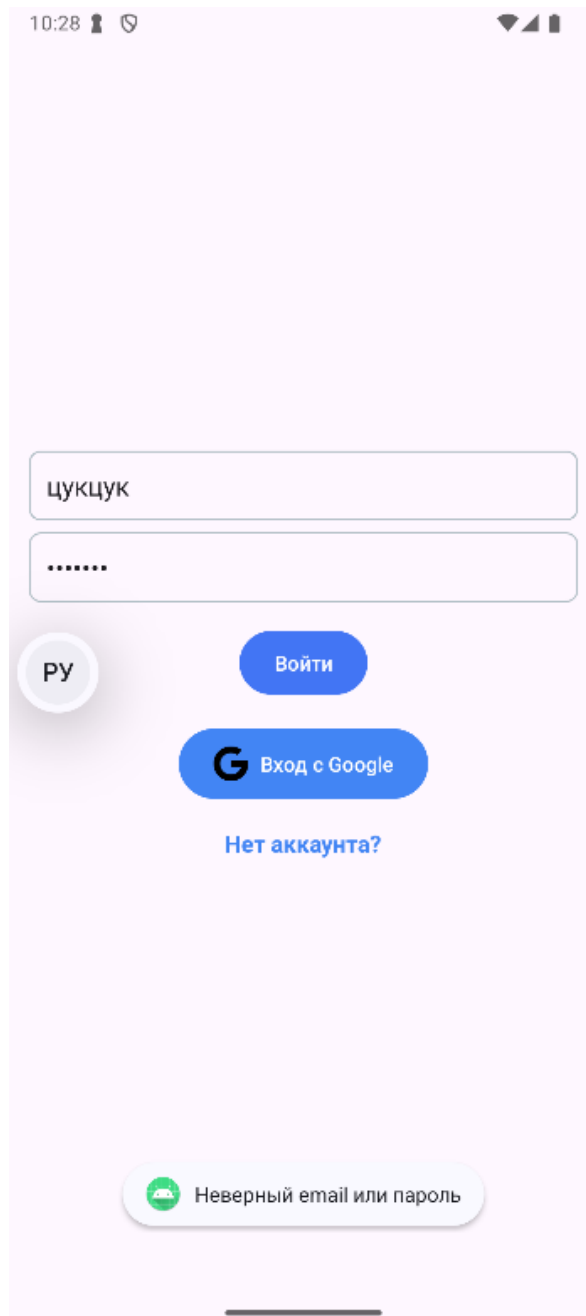


Рис. 3.3. Ошибка при авторизации

Тест №4. Показал, что при нажатии на кнопку добавления в избранное работа добавляется в избранное, что показано на рисунках 3.4, 3.5

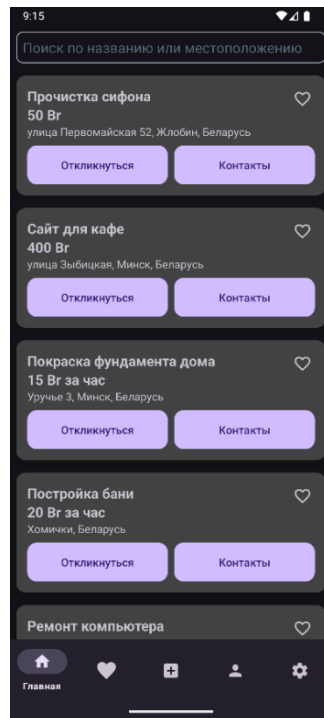


Рис. 3.4. До добавления в избранное

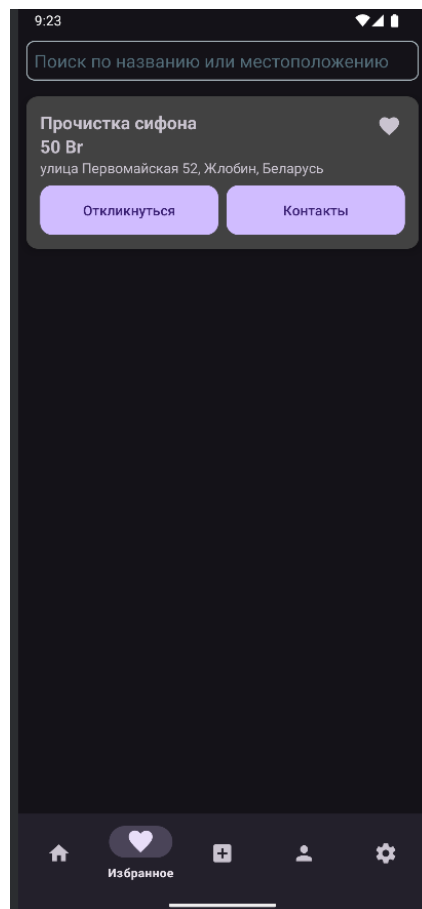


Рис. 3.5. После добавления в избранное

Тест №5. Показал, что при оценке выполненной работы рейтинг пользователя меняется что показано на рисунках 3.6, 3.7, 3.8.

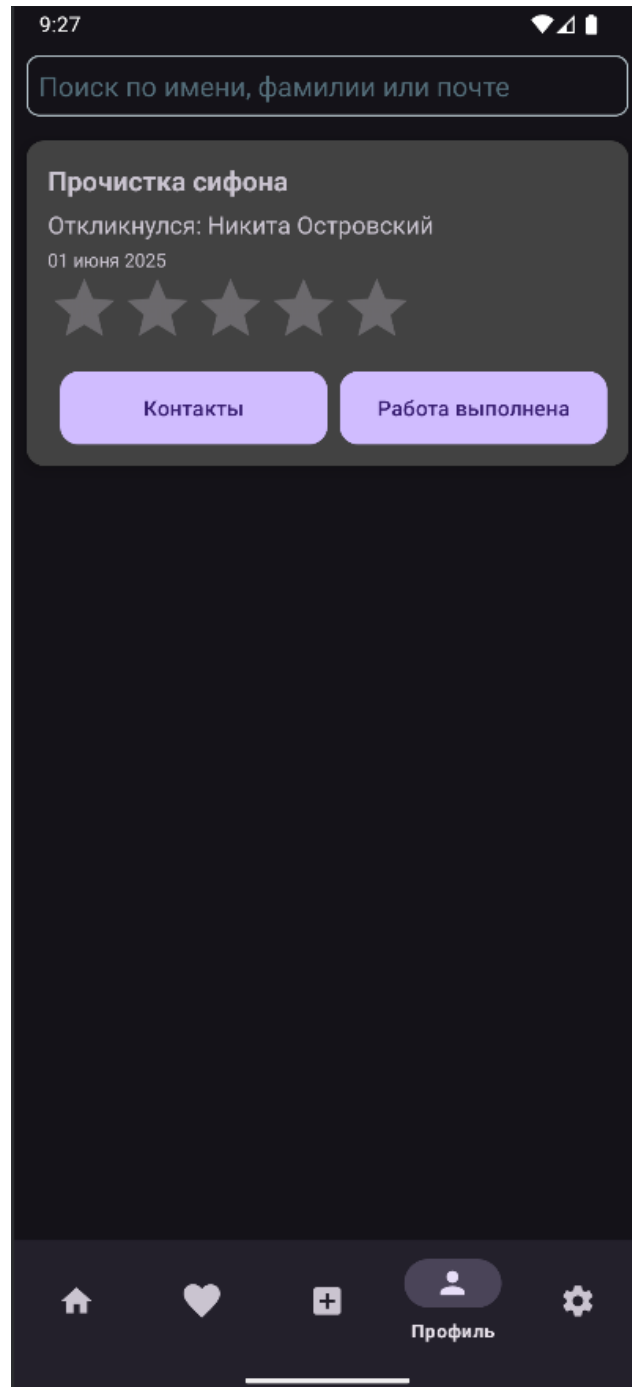


Рис. 3.6. У соискателя нет оценок

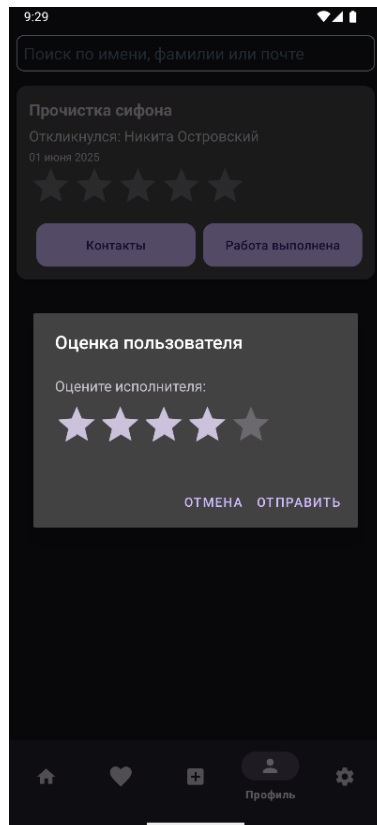


Рис. 3.7. Работодатель оставляет оценку

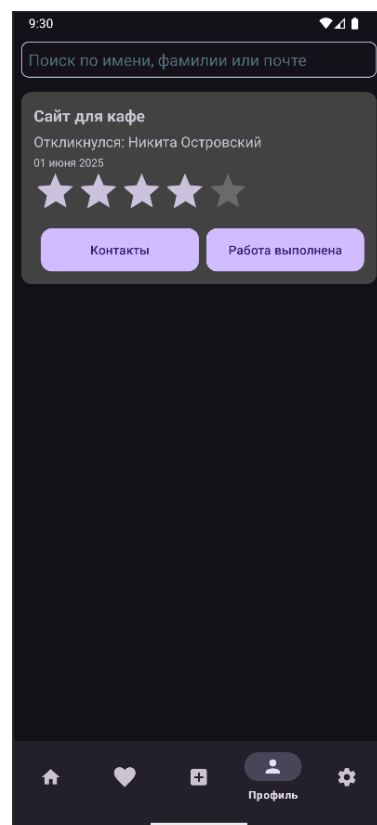


Рис. 3.8. У соискателя появился рейтинг

Тест №6. Показал поиск по названию, что показано на рисунках 3.9, 3.10.

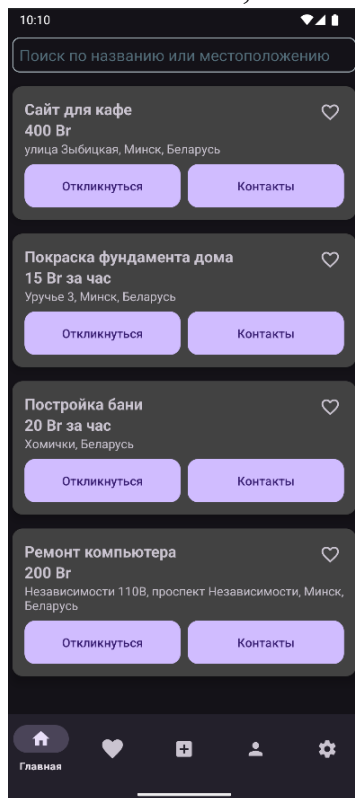


Рис. 3.9. Первоначальный список

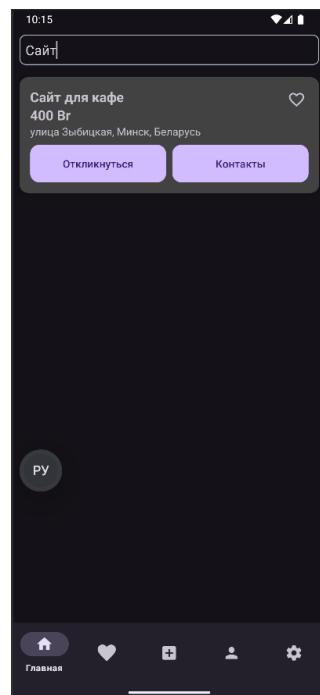


Рис. 3.10. После фильтрации

Тест №7. Показал результат поиска по местоположению, что можно увидеть на рисунках 3.11, 3.12.

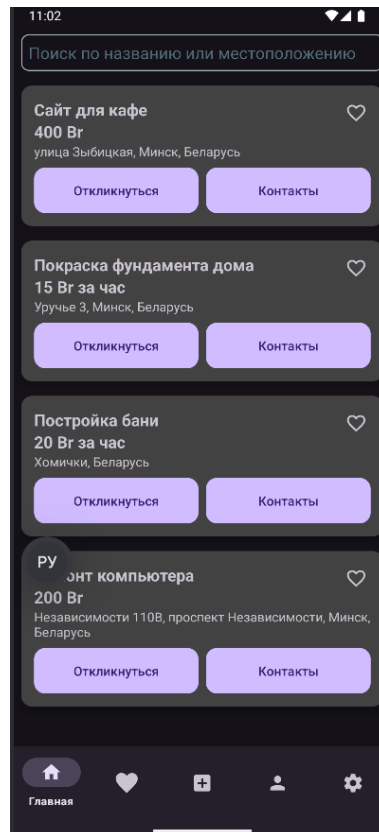


Рис. 3.11. Первоначальный список

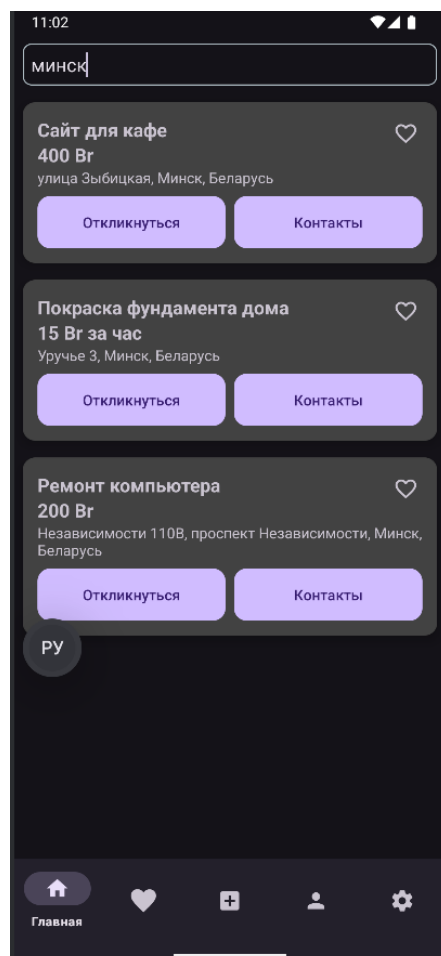
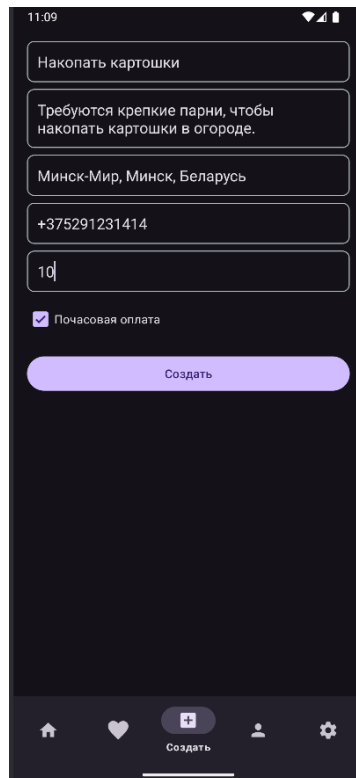


Рис. 3.12. После фильтрации

Тест №8. Результат создания вакансии представлен на рисунках 3.13, 3.14.



11:09

Накопать картошки

Требуются крепкие парни, чтобы накопать картошки в огороде.

Минск-Мир, Минск, Беларусь

+375291231414

10

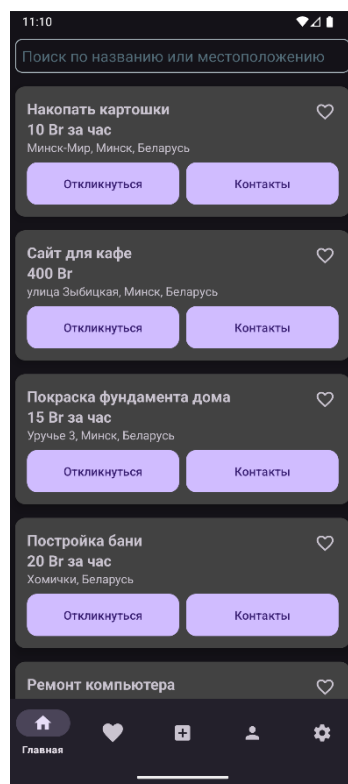
☒ Почасовая оплата

Создать

Home Heart Add Person Settings

Создать

Рис. 3.13. Добавление вакансии



11:10

Поиск по названию или местоположению

Накопать картошки
10 Вт за час
Минск-Мир, Минск, Беларусь

Откликнуться Контакты

Сайт для кафе
400 Вт
улица Зыбичкая, Минск, Беларусь

Откликнуться Контакты

Покраска фундамента дома
15 Вт за час
Уручье 3, Минск, Беларусь

Откликнуться Контакты

Постройка бани
20 Вт за час
Хомички, Беларусь

Откликнуться Контакты

Ремонт компьютера

Home Heart Add Person Settings

Главная

Рис. 3.14. Результат добавления

4. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА

Для того, чтобы зайти в приложение необходимо нажать на иконку, как указано на рисунке 4.1.



Рис. 4.1. Иконка

После чего пользователь попадает на страницу авторизации, где может войти в аккаунт, что показано на рисунке 4.2.

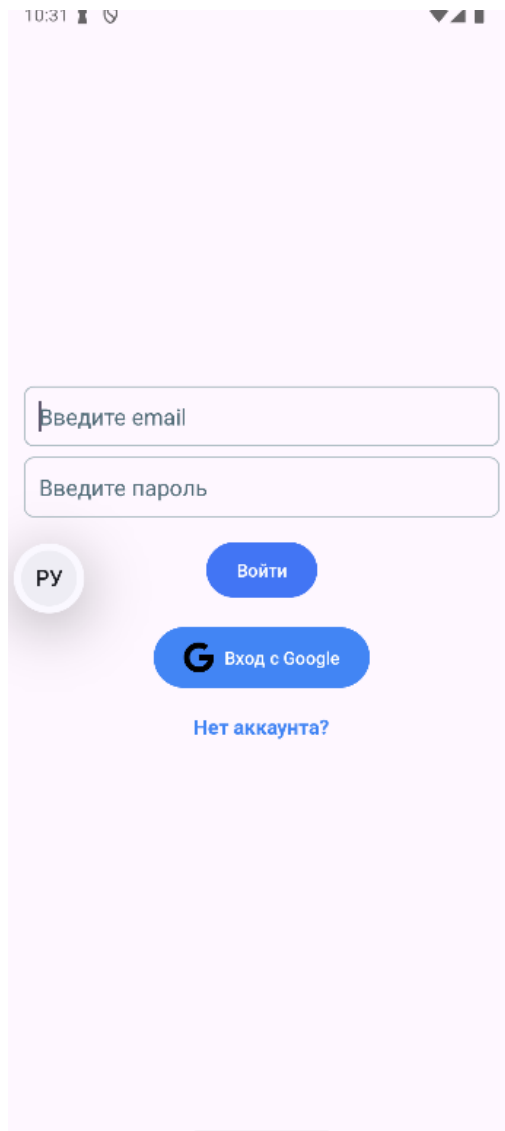


Рис. 4.2. Страница авторизации

На главной странице пользователь может просмотреть вакансии, как указано на рисунке 4.4.

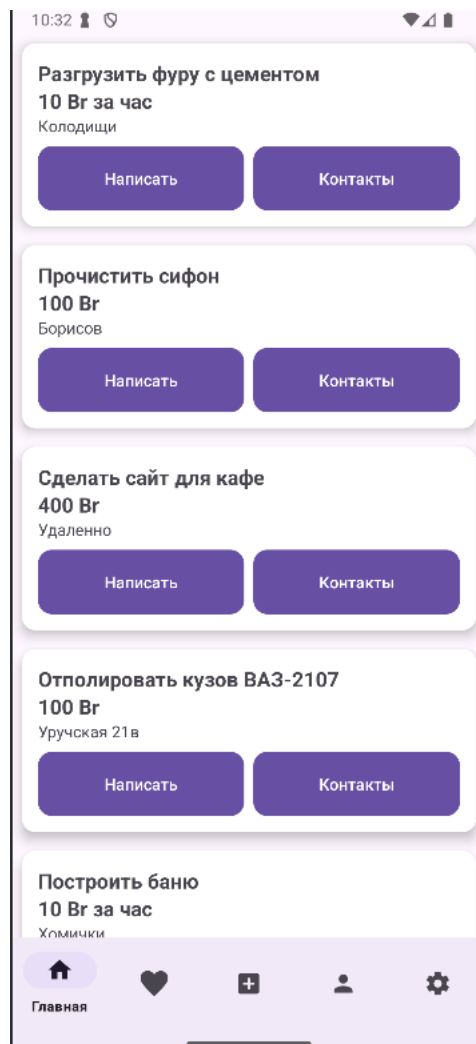


Рис. 4.4. Главная страница

При желании более подробнее просмотреть вакансию, пользователь может на нее нажать, как представлено на рисунке 4.5.



Рис. 4.5. Подробная информация

5. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО СРЕДСТВА

В рыночных условиях программное средство выступает преимущественно в виде продукции научно-технических организаций, представляющей собой функционально завершённые и имеющие товарный вид программные средства, поставляемые заказчикам и продаваемые покупателям по рыночным ценам. Все завершённые разработки ПО являются научно-технической продукцией.

Стоимостная оценка ПО и определение экономического эффекта у разработчика предполагают составление сметы затрат, которая в денежном выражении включает следующие статьи расходов:

- заработную плату исполнителей, основную (З_о) и дополнительную (З_д);
- отчисления в фонд социальной защиты населения (З_{сз});
- налоги, от фонда оплаты труда (Н_е);
- материалы и комплектующие (М);
- спецоборудование (Р_с);
- машинное время (Р_м);
- расходы на научные командировки (Р_{нк});
- прочие прямые расходы (П_з);
- накладные расходы (Р_н).

На основе общей суммы расходов по всем статьям (С_р) и результатов маркетинговых исследований на рынке ПО определяется плановая отпускная цена (Ц_о) с учётом прибыли (рентабельности) и налогов, включаемых в цену.

По уточненному объёму ПО и нормативам затрат труда в расчёте на единицу объема определяются нормативная и общая трудоемкость разработки ПО.

Нормативная трудоёмкость разработки ПО (Т_{Ен}) определяется на основании принятого к расчёту объема (V_у) и категории сложности ПО (чел./дн.), уточняется с учётом сложности и новизны проекта, степени использования стандартных модулей при разработке. Нормативная трудоёмкость разработки ПО служит основой для определения общей трудоёмкости.

Общая трудоёмкость ПО (Т_{Ер}) определяется по формуле (5.1):

$$Т_{Ер} = Т_{Ен} \times K_c \times K_n \times K_T \times K_{ур}, \quad (5.1)$$

где Т_{Ен} – нормативная трудоёмкость разработки программного продукта, определяемая относительно общего количества строк исходного кода;

К_с – коэффициент сложности;

К_н – коэффициент, учитывающий степень новизны;

K_T – коэффициент, учитывающий использование стандартных модулей;

$K_{ур}$ – коэффициент, учитывающий средства разработки.

Категория сложности ПО определяется в зависимости от наличия (отсутствия) следующих характеристик:

- высокий уровень языкового интерфейса с пользователем;
- режим работы в реальном времени;
- управление удаленными объектами;
- машинная графика, многомашинные комплексы;
- существенное распараллеливание вычислений;
- нестандартная конфигурация технических средств;
- оптимизационные и особо сложные инженерные и научные расчёты;
- переносимость ПО.

Влияние фактора сложности на трудоемкость учитывается умножением нормативной трудоемкости на соответствующий коэффициент сложности.

Коэффициент сложности (K_c) учитывает дополнительные затраты труда, связанные со сложностью разрабатываемого программного продукта, при расчёте которого необходимо воспользоваться дополнительными коэффициентами сложности ПО.

Коэффициент, учитывающий степень использования при разработке ПО стандартных модулей (K_T) определяется их удельным весом в общем объеме проектируемого продукта. При определении влияния этого фактора на трудоемкость он учитывается путем умножения нормативной трудоемкости на соответствующий коэффициент.

Коэффициент новизны разрабатываемого ПО (K_n) определяется путём сравнения характеристик разрабатываемого ПО с имеющимися аналогами, позволяет определить экспертным путём, степень его новизны. Влияние фактора новизны на трудоемкость учитывается путем умножения трудоемкости на соответствующий коэффициент новизны. Для расчета данного показателя используются поправочные коэффициенты, учитывающие новизну ПО (K_n).

$$TE_p = 72 \times 0,18 \times 0,63 \times 0,65 \times 0,65 = 4 \text{ чел./дн.}$$

Основной статьей расходов на создание ПО является заработная плата разработчиков (исполнителей) проекта, в число которых принято включать инженеров-программистов, участвующих в написании кода, руководителей проекта, системных архитекторов, дизайнеров, разрабатывающих пользовательский интерфейс, разработчиков баз данных, Web-мастеров и других специалистов, необходимых для решения специальных задач в команде.

Заработная плата руководителей организации и работников вспомогательных служб (инфраструктуры) учитывается в накладных расходах.

Для расчёта основной заработной платы исполнителей используется общая трудоёмкость, плановая численность работников и плановые сроки разработки ПО являются базой для расчёта основной заработной платы разработчиков проекта. Оплата труда осуществляется на основе Единой тарифной сетки Республики Беларусь (ЕТС), в которой даны тарифные разряды и тарифные коэффициенты. Действует инструкция по распределению работников внебюджетного сектора экономики Республики Беларусь по тарифным разрядам с учетом категории, должности, образования, сложности выполняемой работы и практического опыта. По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов-исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности.

Основная заработная плата исполнителей на конкретное программное обеспечение рассчитывается по формуле (5.2):

$$ЗП_{\text{осн}} = ТС_{\text{ч}} \times ТЕ_{\text{р}} \times К_{\text{пр}}, \quad (5.2)$$

где $ТС_{\text{ч}}$ – часовая тарифная ставка соответствующего разряда, руб.;

$ТЕ_{\text{р}}$ – трудоемкость разработки, часов;

$К_{\text{пр}}$ – коэффициент премирования.

$$ЗП_{\text{осн}} = 13,12 \times 4 \times 1,15 \times 8 = 482,82 \text{ руб.}$$

Расчёт дополнительной заработной платы исполнителей за конкретное ПО ($ЗП_{\text{доп}}$) включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по нормативу в процентах к основной заработной плате, согласно формуле (5.3):

$$ЗП_{\text{доп}} = \frac{ЗП_{\text{осн}} \times \%ЗП_{\text{доп}}}{100}, \quad (5.3)$$

где $ЗП_{\text{осн}}$ – основная заработная плата исполнителя, руб.;

$\%ЗП_{\text{доп}}$ – процент дополнительной заработной платы, %.

$$ЗП_{\text{доп}} = \frac{482,82 \times 20}{100} = 96,56 \text{ руб.}$$

Обязательным условием является определение Обязательным условием является определение отчислений в фонд социальной защиты населения

согласно формуле (5.4):

$$\Phi\text{СЗН} = \frac{(\text{ЗП}_{\text{осн}} + \text{ЗП}_{\text{доп}}) \times \% \Phi\text{СЗН}}{100}, \quad (5.4)$$

где $\text{ЗП}_{\text{осн}}$ – основная заработная плата исполнителей, руб.;

$\text{ЗП}_{\text{доп}}$ – дополнительная заработная плата исполнителей, руб.;

$\% \Phi\text{СЗН}$ – установленный законодательством процент отчисления в фонд социальной защиты населения, %.

$$\Phi\text{СЗН} = \frac{(482,82 + 96,56) \times 34}{100} = 196,99 \text{ руб.}$$

Расходы на материалы и комплектующие определяются на основании сметы затрат, разрабатываемой на программное обеспечение с учетом действующих нормативов. В данные затраты включают расходы на магнитные носители, бумагу, красящие ленты и другие материалы, необходимые для разработки программного обеспечения. Нормы расхода материалов в суммарном выражении определяются в расчете на 100 строк исходного кода или по нормативу в процентах основной заработной платы разработчиков, который устанавливается организацией.

Сумма затрат на расходные материалы рассчитывается по формуле (5.5):

$$\text{МЗ} = \frac{\text{ЗП}_{\text{осн}} \times \text{Н}_\text{м}}{100}, \quad (5.5)$$

где $\text{ЗП}_{\text{осн}}$ – основная заработная плата исполнителей, руб.;

$\text{Н}_\text{м}$ – норматив расхода на материалы, %.

$$\text{МЗ} = \frac{482,82 \times 5}{100} = 24,14 \text{ руб.}$$

Расходы по статье «Машинное время» включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода машинного времени в зависимости от характера решаемых задач и типа ПК (0,5 машино-часов (5.6):

$$\text{МВ} = \text{Ц}_\text{м} \times \frac{V_{\text{строк}}}{100} \times \text{Н}_\text{мв}, \quad (5.6)$$

где $\text{Ц}_\text{м}$ – цена одного машино-часа, руб.;

$V_{\text{строк}}$ – общий объем ПО (количество строк исходного кода);

$\text{Н}_\text{мв}$ – норматив расхода машинного времени на отладку 100 строк исходного кода (машино-часов).

$$MB = 0,55 \times \frac{17210}{100} \times 0,6 = 56,79 \text{ руб.}$$

Общехозяйственные расходы связаны с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды, и относятся на конкретное программное обеспечение по нормативу в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации (30-80 %) и рассчитывается по формуле (5.7):

$$OXP = \frac{ЗП_{осн} \times \%OXP}{100}, \quad (5.7)$$

где % OXP – норматив общехозяйственных расходов, %.

$$OXP = \frac{482,82 \times 35}{100} = 168,99 \text{ руб.}$$

Прочие прямые затраты – это затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Они определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате и рассчитывается по формуле (5.8):

$$П_{рз} = \frac{ЗП_{осн} \times Н_{п.з.}}{100}, \quad (5.8)$$

где Н_{п.з.} – норматив прочих прямых затрат, %.

$$П_{рз} = \frac{482,82 \times 10}{100} = 48,48 \text{ руб.}$$

Результаты расчёта затрат внесём в таблицу 5.1.

Таблица 5.1 – Расчет полной себестоимости программного обеспечения

Наименование статей затрат	Сумма, руб.
Материальные затраты	24,14
Основная заработная плата	482,82
Дополнительная заработная плата	96,56
Отчисления в ФСЗН	196,99
Машинное время	56,79
Общехозяйственные расходы	168,99
Прочие прямые затраты	48,28
Полная себестоимость	1074,57

Для определения цены ПО необходимо рассчитать плановую прибыль. Рентабельность и прибыль создаваемого ВП определяется исходя из результатов анализа рыночных условий переговоров с заказчиком и согласования с ним отпускной цены.

Прибыль рассчитывается по формуле (5.9):

$$П_{пл} = \frac{с/с_{полн} \times P}{100}, \quad (5.9)$$

где $с/с_{полн}$ – полная себестоимость ПО, руб;

$П_{пл}$ – плановая прибыль от реализации ПО, руб.;

P – уровень рентабельности ПО, %.

$$П_{пл} = \frac{1122,41 \times 20}{100} = 224,48 \text{ руб.}$$

Отпускная цена без НДС определяется по формуле (5.10):

$$Ц_{ОТП(без НДС)} = с/с_{полн.} + П_{пл}, \quad (5.10)$$

$$Ц_{ОТП(без НДС)} = 1074,57 + 214,91 = 1\,289,58 \text{ руб.}$$

Сумма НДС определяется по формуле (5.11):

$$НДС = \frac{Ц_{ОТП(без НДС)} \times \%НДС}{100}, \quad (5.11)$$

где $\% НДС$ – ставка НДС, %.

$$НДС = \frac{1289,48 \times 20}{100} = 257,9 \text{ руб.}$$

Отпускная цена ПО с НДС ($Ц_{ОТП(с НДС)}$) рассчитывается по формуле (5.12):

$$Ц_{ОТП(с НДС)} = Ц_{ОТП(без НДС)} + НДС, \quad (5.12)$$

где НДС – сумма налога на добавленную стоимость, руб.

$$Ц_{ОТП(с НДС)} = 1289,48 + 257,9 = 1\,547,38 \text{ руб.}$$

Полученные результаты расчетов внесём в таблицу 5.2.

Таблица 5.2 – Расчет отпускной цены программного обеспечения

Показатели	Сумма, руб.
Полная себестоимость	1074,57
Прибыль	214,91
Цена без НДС	1289,48
НДС	257,9
Отпускная цена с НДС	1547,38

Экономическим эффектом от создания программного обеспечения для разработчика является чистая прибыль. Она определяется после уплаты налога на прибыль, формула (5.13):

$$\Pi_{\text{ч}} = \Pi_{\text{пл}} \times \left(1 - \frac{\%N_{\text{пр}}}{100}\right), \quad (5.13)$$

где $\%N_{\text{пр}}$ – ставка налога на прибыль, %.

$$\Pi_{\text{ч}} = 214,91 \times \left(1 - \frac{20}{100}\right) = 171,93 \text{ руб.}$$

Затраты на разработку мобильного приложения «Shabashka» составили 1074,57 руб. Отпускная прогнозируемая цена составляет 1547,38 руб. Прибыль разработчика составит 214,91 руб. Рассчитанная цена разработанного программного обеспечения позволит полностью покрыть все затраты.

Для разрабатываемого приложения важно выполнение всех функций, указанных в требованиях. Также необходимо обеспечить удобство использования для пользователя. Приложение должно быть стабильным, не часто выходить из строя и иметь минимум ошибок. Необходим тщательное тестирование на различных операционных системах и версиях популярных браузеров.

Веб-приложение должно обеспечивать быструю загрузку и отклик, даже при большом числе пользователей. Оптимизация кода и эффективное взаимодействие с БД важны. Важно обеспечить защиту личных данных пользователей, обеспечить шифрование информации и предотвратить атаки и утечку данных.

Приложение должно корректно отображаться и функционировать на различных устройствах и версиях операционной системы. Интерфейс приложения должен быть интуитивно понятным, с удобной навигацией. Лёгкость отклика на подработку, просмотр карты и создание объявлений напрямую влияет на удобство пользователей и их лояльность к приложению.

Для приблизительного расчета себестоимости программного средства рассмотрим следующие затраты:

- размещение на хостингах. Для развертывания приложения на ресурсах облачных провайдеров требуется ежемесячный платеж в размере 39 белорусских рублей;

- дополнительные расходы будут затрачены на покупку необходимых ресурсов для разработки, также расходы на интернет-соединение, плюс небольшие дополнительные затраты на поддержку ПС от 150 до 200 белорусских рублей ежемесячно.

По итогу сумма всех вышеперечисленных расходов примерно составит от 189 до 239 белорусских рублей в месяц для поддержания работы приложения.

6. ОХРАНА ТРУДА

6.1. Правовые и организационные вопросы охраны труда

Данный дипломный проект может быть использован в деятельности как на предприятиях с большой численностью работников, так и индивидуальными предпринимателями. Для безопасной реализации дипломного проекта необходимо обратить внимание на вопросы охраны труда, то есть системе обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающие правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия и средства.

Общий контроль и руководство по соблюдению правил охраны труда в ООО «Страдл» осуществляет инженер по охране труда (руководство по координации работы по охране труда возложено на Свидеровича Вальдемара Бальтазаровича). Она координирует работу по проведению всех видов инструктажей, контролю за состоянием охраны труда в ООО «Страдл», планирует и проводит мероприятия, направленные на профилактику травматизма и профессиональных заболеваний.

Перед тем как вновь принятые сотрудники или практиканты в ООО «Страдл» придут на свое рабочее место, с ними проводится вводный инструктаж. При проведении инструктажа были рассмотрены вопросы внутреннего трудового распорядка, прав и обязанностей работника и работодателя. Также сотрудники ООО «Страдл» получили общую информации об организации и описание опасных и вредных факторов по предприятию в целом.

Со всеми вновь принятыми сотрудниками или практикантами в ООО «Страдл» проводится первичный инструктаж, выделяется персональное рабочее место с установленными техническими средствами.

При проведении первичного инструктажа специалистом по охране труда особое внимание уделялось самостоятельному подключению приборов к электросети. Было дополнительно подчеркнуто, что при пользовании средствами вычислительной техники и периферийным оборудованием каждый работник должен внимательно и осторожно обращаться с электропроводкой, приборами и аппаратами и всегда помнить, что пренебрежение правилами безопасности угрожает и здоровью, и жизни человека.

В инструкции, с которой я был ознакомлен в первый рабочий день, указывалось о необходимости постоянного контроля на своем рабочем месте за исправным состоянием электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, и заземления. Делался

упор на то, что при обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

В общие правила, соблюдение которых является обязательным для всех работников организации ООО «Страдл», включены и те, которые касаются профилактики повреждения изоляции проводов и возникновения коротких замыканий, а именно не разрешается:

- вешать что-либо на провода;
- закрашивать и белить шнуры и провода;
- закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы;
- выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Среди основных правил, направленных на исключение поражения электрическим током, можно назвать следующие:

- запрещается часто включать и выключать компьютер без необходимости;
- запрещается прикасаться к экрану и к тыльной стороне блоков компьютера;
- не разрешается работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками;
- не разрешается работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе;
- запрещается класть на средства вычислительной техники и периферийном оборудовании посторонние предметы.

В действующих инструкциях ООО «Страдл» сотруднику при работе за монитором рекомендуется раз в 45 минут делать зарядку для глаз.

При проверке соблюдения правил охраны труда указывалось на необходимость поддержания своего рабочего места и инвентаря в чистоте – одна из задач работника.

6.2. Производственная санитария и гигиена труда

В ООО «Страдл» большинство сотрудников работают дистанционно. Лишь небольшое количество персонала находятся на рабочих местах в офисе. У каждого работника есть свое рабочее место, оборудованное техническими средствами, а именно: монитор, системный блок, клавиатура, мышь. Площадь помещения, выделенная одному работнику – 4,5 кв. м. Расстояние между

столами двух сотрудников составляет около двух метров. Так же каждый стол оснащен дополнительными преградами от проводов. У каждого стола размещен удобный стул с регуляторами высоты и длины сиденья.

В действующих инструкциях ООО «Страдл» указано, что работник обязан содержать в чистоте своё рабочее место, не допускать приёма пищи за рабочим столом. Тщательно мыть руки. Следить за порядком и препятствовать скоплению мусора.

Работа программиста в организациях и на предприятиях связана с вычислительными машинами, обработкой различных документов и информации, и подразумевает работу без физического напряжения, но под влиянием большого количества вредных производственных факторов.

При работе с персональным компьютером на программиста действуют следующие вредные факторы:

- ультрафиолетовое и инфракрасное излучения;
- электромагнитные излучения;
- рентгеновское излучение;
- статистическое электричество;
- блики и мерцание экрана;
- акустические шумы работающих принтеров, электродвигателей приводов винчестера, компакт – дисков, вентилятора процессора.

Кроме того, работа программиста за компьютером сопряжена с повышенными физическими и психологическими нагрузками, обусловленными следующими причинами:

- необходимость длительного напряжения, сохранение неподвижной позы во время работы. Длительная неподвижность приводит к усталости, болям в позвоночнике, шее, плечевых суставах;
- работа с клавиатурой вызывает боль в локтевых суставах, предплечьях, запястьях, кистях и пальцах рук;
- перегрузка зрительного аппарата.

Основная нагрузка на глаза обусловлена тем, что большая часть информации воспринимается визуально. Излишняя нагрузка на глаза может быть ослаблена правильным выбором монитора и его настройкой.

Поэтому в ООО «Страдл» проводится комплекс мероприятий, направленных на минимизацию воздействия вредных производственных факторов. Чтобы решить проблему с микроклиматом и шумом, в ООО «Страдл» серьезно подходят к оборудованию рабочих мест работников.

В рабочих кабинетах поддерживаются следующие показатели микроклимата:

- температура воздуха от плюс 190 до плюс 25°C;
- относительная влажность от 40 до 50%;

- скорость движения воздуха до 0,1 м/с;
- запыленность до 10 мг/м³.

Для борьбы с шумом применяются звукопоглощающие материалы.

Обязательно проводится проветривание помещений, так как в офисе становится очень душно, что может плохо сказаться на работоспособности, проветривание осуществляется три или четыре раза в течение рабочего дня.

Для создания благоприятных условий труда к организации рабочего места программиста клавиатура должны быть размещена на столе на высоте по отношению к полу 700 мм. Дисплей должен быть помещен на столе так, чтобы его экран располагался в вертикальной плоскости под углом менее 15 градусов. Во избежание перенапряжения органов зрения должно соблюдаться расстояние до монитора 40-50 см. Данные требования соблюдались мной на рабочем месте в ООО «Страдл».

В помещении, где находилось моё рабочее место имелось естественное и искусственное освещение, что помогает избежать проблемы со зрением. Освещенность на поверхности стола в зоне размещения рабочего документа составляла 300 – 500 люкс. Освещение не создавало бликов на поверхности экрана.

Для предотвращения утомления и повышения работоспособности был установлен рациональный режим труда и отдыха. В ООО «Модсен» установлен регламентированный режим труда и отдыха. Предприятие начинает работу с 9:00 утра и заканчивает в 18:00. Обеденное время начинается с 13:00 и заканчивается в 14:00. Помимо обеденного перерыва сотрудник имеет право взять технологический перерыв: через каждый час работы введён перерыв на 5-10 минут, а через два часа – на 15 минут. Рабочая неделя является пятидневной, начало недели в понедельник и окончание в пятницу.

6.3. Техника безопасности (электробезопасность)

Место проведения дипломной практики относилось к помещению без повышенной опасности. Вся оргтехника ООО «Страдл» оборудована бесперебойными источниками питания. При отключении основного электропитания источники бесперебойного питания позволяют избежать потери информации.

Периодически проводится комплексный осмотр оборудования на внешние повреждения.

А также проводится перечень мероприятий для обеспечения электробезопасности:

1) Отключают напряжение и принимают меры, исключающие его ошибочную подачу к месту работы.

2) Вывешивают предупредительные плакаты на коммутационной аппаратуре, на постоянных и временных ограждениях.

3) Проверяют, есть ли напряжение на отключенной для работы части установки и накладывают на токоведущие части установки переносное заземление.

В ООО «Страдл» оптимально рассчитана нагрузка на сеть, предусмотрены запасные электрогенераторы со своей электрической сетью, что исключает возможность перегрузки основной электрической сети, а при возникновении экстренной ситуации предоставляет возможность экстренно перейти на дополнительную сеть:

- немедленно прекращается работа, выключается аппаратура, о всех недостатках сообщается должностному лицу;

- после окончания работы приостанавливается функционирование всех использованных приложений и программ.

При проведении первичного инструктажа особое внимание уделялось самостоятельному подключению приборов к электросети. Было дополнительно подчеркнуто, что при пользовании средствами вычислительной техники и периферийным оборудованием каждый работник должен внимательно и осторожно обращаться с электропроводкой, приборами и аппаратами и всегда помнить, что пренебрежение правилами безопасности угрожает и здоровью, и жизни человека.

Василович Вероникой Викторовной, инженером по охране труда, неоднократно указывалось на то, что необходимо постоянно следить на своем рабочем месте за исправным состоянием электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, и заземления. Делался упор на то, что при обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности;

Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается:

- вешать что-либо на провода;
- закрашивать и белить шнуры и провода;
- закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы;
- выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Для исключения поражения электрическим током запрещается:

- часто включать и выключать компьютер без необходимости;
- прикасаться к экрану и к тыльной стороне блоков компьютера;
- работать на средствах вычислительной техники и периферийном

оборудовании мокрыми руками;

– работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе.

6.4. Пожарная безопасность

Среди возможных причин возникновения пожара при эксплуатации ПЭВМ (возникновение аварийных ситуаций) можно назвать следующие:

- короткие замыкания;
- перегрузки;
- повышение переходных сопротивлений в электрических контактах;
- перенапряжение;
- возникновение токов утечки.

В ООО «Страдл» принимаются меры, соблюдение которых помогает исключить с большой вероятностью возможность возникновения пожара:

1) Для понижения воспламеняемости и способности распространять пламя кабели покрывают огнезащитным покрытием;

2) При ремонтно-профилактических работах строго соблюдаются правила пожарной безопасности;

3) Каждое из помещений, где производится эксплуатация устройств ПЭВМ, оборудовано первичными средствами пожаротушения и обеспечено инструкциями по их применению. В качестве средств пожаротушения в ООО «Страдл» разрешается использование углекислотного огнетушителя типа ОУ-2, ОУ-5, а также порошковый тип. Применение пенных огнетушителей не допускается, так как жидкость пропускает ток;

4) Устройства ПЭВМ установлены вдали отопительных и нагревательных приборов (расстояние не менее 1 м и в местах, где не затруднена их вентиляция и нет прямых солнечных лучей);

5) Разработаны организационные меры по обучению персонала навыкам ликвидации пожара имеющимися в наличии средствами тушения пожара до прибытия пожарного подразделения.

К средствам тушения пожара, предназначенных для локализации небольших загораний в помещениях с ПЭВМ относятся:

- пожарные стволы;
- внутренние пожарные водопроводы;
- огнетушители;
- сухой песок;
- асбестовые одеяла.

Для профилактики действий при пожаре разработан и неукоснительно применяется специальный план эвакуации из помещения.

В каждом кабинете ООО «Страдл» расположены плакаты с памятками о мероприятиях по соблюдению правил охраны труда и техники безопасности. На этих стендах можно получить основную информацию о мероприятиях по охране труда, указаны ответственные за соблюдение пожарной безопасности и соблюдение правил охраны труда.

В ООО «Страдл» проводятся меры по устранению причин пожаров, осуществляется постоянная уборка мусора, который может вызвать возгорание, электроинструмент хранится в специальных условиях.

В ООО «Страдл» осуществляется ряд мероприятий, направленных на предотвращение огня, на устройство путей эвакуации людей и материальных ценностей, на использование средств и условий для быстрой ликвидации пожара. Присутствует план эвакуации людей при возникновении пожара из помещения, а также плакаты первой медицинской помощи.

ЗАКЛЮЧЕНИЕ

Основной целью выполненной работы было изучение информационной системы, анализ организации обмена информацией в сети, исследование уровня оснащённости и различных видов программного обеспечения, а также разработка собственных программных решений.

В ходе прохождения дипломной практики были подготовлены материалы к дипломному проекту, а также закреплены и углублены теоретические знания, полученные в процессе обучения. Полученные знания по технологии обработки информации были успешно применены на практике при разработке функционала и адаптации решений для потребностей клиентов.

Исследование уровня оснащённости и различных видов программного обеспечения способствовало лучшему пониманию современных технологий, их преимуществ и недостатков.

Выработан и отточен практический опыт по разработке, тестированию и сопровождению программных продуктов.

Профессиональное мышление было развито через решение реальных задач разработки и оптимизации программного обеспечения с учетом требований руководителя практики. Уровень квалификации в области мобильной разработки был успешно освоен, а полученные знания позволят успешно применяться в будущей профессиональной деятельности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Developer.android.com [Электронный ресурс] / Создай свое первое андроид приложение. Режим доступа: <https://developer.android.com/codelabs/basic-android-kotlin-compose-first-app#0> – Дата доступа: 15.04.2025.

2) W3schools.com [Электронный ресурс] / Обучение Java. Режим доступа: <https://www.w3schools.com/java/> – Дата доступа: 16.04.2025.

3) Skillfactory.ru [Электронный ресурс] / Android-разработчик. Режим доступа: <https://skillfactory.ru/android-razrabotchik> – Дата доступа: 12.04.2025.

4) Wikipedia.org [Электронный ресурс] / База данных. Режим доступа: https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85 – Дата доступа: 13.04.2025.

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

БД – база данных;

ПС – программное средство;

СУБД – система управления базами данных.

ПРИЛОЖЕНИЕ 1 (ОБЯЗАТЕЛЬНОЕ) – ЛИСТИНГ ПРОГРАММЫ

```
package com.example.shabashka;

import androidx.activity.EdgeToEdge;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.fragment.app.Fragment;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class BaseActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_base);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, 0);
            return insets;
        });

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);

        if (savedInstanceState == null) {
            loadFragment(new HomeFragment());
        }

        bottomNavigationView.setOnItemSelectedListener(item -> {
            Fragment selectedFragment = null;
```

```

        if (item.getItemId() == R.id.nav_home) {
            selectedFragment = new HomeFragment();
        } else if (item.getItemId() == R.id.nav_favorites) {
            selectedFragment = new FavoritesFragment();
        } else if (item.getItemId() == R.id.nav_profile) {
            selectedFragment = new ProfileFragment();
        } else if (item.getItemId() == R.id.nav_create) {
            selectedFragment = new CreateFragment();
        }

        if (selectedFragment != null) {
            loadFragment(selectedFragment);
        }

        return true;
    });
}

private void loadFragment(Fragment fragment) {
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction transaction = fragmentManager.beginTransaction();
    transaction.replace(R.id.content_frame, fragment);
    transaction.commit();
}
}

package com.example.shabashka;

import android.annotation.SuppressLint;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

```

import java.util.ArrayList;
import java.util.List;

public class HomeFragment extends Fragment {
    private JobAdapter jobAdapter;
    private List<Job> jobList;
    private JobLoader jobLoader;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_home, container, false);
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        RecyclerView recyclerView = view.findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(requireContext()));

        jobList = new ArrayList<>();
        jobAdapter = new JobAdapter(requireContext(), jobList);
        recyclerView.setAdapter(jobAdapter);

        jobLoader = new JobLoader(requireContext());
        loadJobs();
    }

    @SuppressWarnings("NotifyDataSetChanged")
    private void loadJobs() {
        jobLoader.loadJobs(jobs -> {
            jobList.clear();
            jobList.addAll(jobs);
            jobAdapter.notifyDataSetChanged();
        });
    }
}

```

```
package com.example.shabashka;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class JobDetailsFragment extends Fragment {

    private static final String ARG_TITLE = "title";
    private static final String ARG_LOCATION = "location";
    private static final String ARG_SALARY = "salary";
    private static final String ARG_HOURLY = "hourly";
    private static final String ARG_DESCRIPTION = "description";

    public static JobDetailsFragment newInstance(String title, String location, String salary, boolean hourly, String
description) {
        JobDetailsFragment fragment = new JobDetailsFragment();
        Bundle args = new Bundle();
        args.putString(ARG_TITLE, title);
        args.putString(ARG_LOCATION, location);
        args.putString(ARG_SALARY, salary);
        args.putBoolean(ARG_HOURLY, hourly);
        args.putString(ARG_DESCRIPTION, description);
        fragment.setArguments(args);
        return fragment;
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle
savedInstanceState) {
        return inflater.inflate(R.layout.fragment_job_details, container, false);
    }
}
```

```
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    TextView tvTitle = view.findViewById(R.id.tvTitle);
    TextView tvLocation = view.findViewById(R.id.tvLocation);
    TextView tvSalary = view.findViewById(R.id.tvSalary);
    TextView tvDescription = view.findViewById(R.id.tvDescription);

    Bundle args = getArguments();
    if (args != null) {
        String title = args.getString(ARG_TITLE);
        String location = args.getString(ARG_LOCATION);
        String salary = args.getString(ARG_SALARY);
        boolean hourly = args.getBoolean(ARG_HOURLY);
        String description = args.getString(ARG_DESCRIPTION);

        tvTitle.setText(title);
        tvLocation.setText(location);
        tvSalary.setText(hourly
            ? String.format("%s %s", salary, getString(R.string.salary_per_hour))
            : String.format("%s %s", salary, getString(R.string.salary_fixed)));
        tvDescription.setText(description);
    }
}
```