**МИНОБРНАУКИ РОССИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра информационных систем и технологий

Веб-приложение “Finance” Курсовой проект

09.03.02 Информационные системы и технологии Программирования и информационных технологий

Зав.кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Д. Махортов, профеcсор

Руководитель В.С. Тарасов, ст. преподаватель Обучающийся К. Саидов, 4 курс, д/о

Обучающийся К. Чашемов, 4 курс, д/о

Воронеж 2024

Содержание

[Введение 3](#__RefHeading___Toc765_3588212081)

[1. Анализ предметной области 5](#__RefHeading___Toc872_1689325553)

# [1.1. Обзор предметной области 5](#__RefHeading___Toc9427_2178085446)

[2. Постановка задачи](#__RefHeading___Toc944_999031852) 6

# [2.1. Этапы разработки 6](#__RefHeading___Toc9436_2178085446)

# [2.2. Функциональные требования 6](#__RefHeading___Toc9438_2178085446)

# [2.3. Средства реализации 7](#__RefHeading___Toc9440_2178085446)

## [2.3.1. СУБД 7](#__RefHeading___Toc9452_2178085446)

## [2.3.2. Сервер 7](#__RefHeading___Toc9454_2178085446)

## [2.3.3. Архитектура серверной части 8](#__RefHeading___Toc9456_2178085446)

## [2.3.4. Клиент 8](#__RefHeading___Toc9475_2178085446)

## [2.3.5. Архитектура клиентской части 9](#__RefHeading___Toc9477_2178085446)

[3. Реализация](#__RefHeading___Toc874_1689325553) 9

# [3.1. Диаграмма вариантов использования](#__RefHeading___Toc9540_2178085446) 9

# [3.2. Диаграмма классов 1](#__RefHeading___Toc9762_3834807095)0

# [3.3. Диаграмма таблиц базы данных 1](#__RefHeading___Toc9764_3834807095)1

# [3.4. Реализация интерфейса 1](#__RefHeading___Toc9766_3834807095)2

[Заключение 2](#__RefHeading___Toc950_999031852)6

Введение

В истории человечества деньги использовались для осуществления торговых операций между людьми, которые были связаны с различными регионами, странами и так далее. Появились такие науки как экономика, благодаря этому изобретению. В процессе исторического развития были созданы различные структуры, занимающиеся финансовыми операциями. Можно привести в пример банковские учреждения, биржи, различные финансовые организации, фонды и т.д. С того момента как возникли первые государства, начали появляться разные валюты и их курсы соответственно.

В современном мире деньги могут принимать самые разнообразные формы. Их можно встретить в различных видах: они могут быть как материальными, так и виртуальными. Например, это могут быть средства, которые находятся на наших банковских счетах, или же криптовалюты, которые используются для оплаты товаров и услуг.

На сегодняшний день существует множество разнообразных форм денег и способов их хранения. В связи с этим возникает необходимость в грамотном ведении учета своих средств. Это включает в себя проведение операций по счету средств в банках, их наличных средств и так далее. При учете подразумевается контроль за движением денежных средств по счетам, анализ этого движения и выработка стратегии и тактики дальнейшего использования.

До появления различных цифровых устройств, управление денежными средствами представляло собой сложную задачу из-за необходимости ручного подсчёта каждой финансовой операции. В связи с этим человеческий фактор часто приводил к ошибкам, что негативно сказывалось на финансовом контроле.

С появлением цифровых устройств и программ, способных выполнять различные математические операции, управление финансами стало более простой задачей. Однако многие из этих программ не предлагали удобного пользовательского интерфейса, не были адаптированы для работы с финансовыми данными и не обеспечивали надежное хранение информации. Из всего вышесказанного следует, что хотя автоматизация расчетов значительно упростила учет финансов для многих людей, она также усложнила процесс из-за необходимости разбираться в программах и обеспечивать безопасное хранение данных. Сложный пользовательский интерфейс также делал учет финансов непростой задачей, требующей навыков в работе с компьютером.

С развитием информационных технологий появились инструменты, которые позволяют разработчикам создавать более удобные для обычных пользователей программы и системы. Теперь пользователю не нужно беспокоиться о том, где и как хранить данные, разбираться в сложных интерфейсах и понимать, что делает каждый пункт меню, потому что современные интерфейсы стали более понятными и простыми, не перегруженными множеством разнообразных компонентов.

Следовательно, наша цель заключается в разработке веб-приложения для финансового учета, которое будет легким в использовании и будет обеспечивать все основные операции с банковскими счетами и имеющимися на них средствами.

1. Анализ предметной области
   1. Обзор предметной области

Существует ряд значимых мотивов для ведения учета и планирования бюджета как для семьи, так и для отдельного человека.

Это выполняется с целью:

* Отслеживания фактических доходов и расходов в определенный период (например, месяц или год). Ведение бюджета помогает понять, сколько вы действительно зарабатываете и на что каждый месяц тратятся ваши деньги. За несколько месяцев можно проанализировать структуру доходов и выявить основные источники (например, заработная плата, доход от предпринимательства, хобби). Кроме того, вы получите полную информацию о своих расходах, что поможет найти способы их сокращения.
* Вы сможете выявить "финансовые утечки", которые приводят к тому, что ваши деньги уходят напрасно. Это могут быть импульсивные и необдуманные покупки или инвестиции, которые не приносят никакой пользы. Теперь, когда вы ясно видите, сколько денег тратится впустую каждый месяц, вы можете легче от них отказаться. Ведение бюджета поможет вам более эффективно использовать заработанные средства и уменьшать расходы.

Начать планирование семейного (или личного) бюджета - это важный шаг в управлении финансами. Планирование помогает использовать деньги осознанно, избегать дорогостоящих кредитов и внимательно присматриваться к расходам. Когда покупка запланирована заранее, у вас будет определенное время, чтобы собрать необходимую сумму. Для тех, кто имеет стабильный доход, разработка четкого плана расходов открывает новые возможности для инвестирования свободных денежных средств, погашения долгов, установления финансовых целей и накопления средств на них.

Оптимизация расходов и поиск дополнительных источников выгоды - ключевые шаги в управлении бюджетом семьи. Анализ доходов и расходов за предыдущие месяцы позволит определить основные и второстепенные расходы, выявить обязательные платежи (например, ЖКХ, обучение детей, налоги) и исключить ненужные траты. Многие расходы можно частично компенсировать, используя карты с кэшбэком, сервисы кэшбэка, приложения для экономии, акции и скидки, а также купоны с сайтов-купонаторов. Эти возможности предоставляют как крупные, так и маленькие торговые точки, и использовать их следует обязательно, чтобы сделать покупки более выгодными.

Начинать накопления важно даже при небольшом доходе. Планирование бюджета поможет понять, что даже маленькие суммы можно откладывать. Оптимизация расходов поможет освободить деньги путем отказа от ненужных трат, а эти средства можно направить на накопления.

Важно создать финансовую подушку безопасности - неприкосновенный резерв на случай неожиданных обстоятельств. Никто не знает, что может произойти завтра, особенно в периоды кризиса и экономических нестабильностей. Инфляция, новые налоги, сокращения в организациях могут существенно ухудшить финансовое положение. Для сохранения привычного уровня жизни и минимизации потерь важно иметь финансовый резерв.

1. Постановка задачи
   1. Этапы разработки

Весь процесс разработки можно поделить на четыре этапа:

Проектирования базы данных, которая является фундаментом для работы сервера приложения. Здесь будут храниться все данные пользователей, включая учетные записи, счета, доходы, расходы по категориям и переводы между счетами.

Разработка серверного приложения является ключевым этапом процесса. Именно сервер обеспечивает непрерывную работу необходимых функций с базой данных и гарантирует надежное изменение данных.

Описание API. Это требуется для того чтобы конечный пользователь сервера мог с легкостью понять как пользоваться его функциями и как это делать корректно чтобы получить ожидаемый результат.

Разработка клиентского приложения. Также основная задача вкр и заключается она в том, чтобы создать приложение реализующее пользовательский интерфейс и взаимодействие с сервером согласно описанному API.

* 1. Функциональные требования

Итоговое приложение должно включать следующий функционал:

Регистрация и аутентификация: Эта функция позволяет пользователям создавать учетные записи и аутентифицироваться в системе. Это позволяет иметь доступ к одним и тем же данным пользователя с разных устройств, что особенно удобно для семейного учета, где каждый член семьи может вносить изменения данных независимо.

Создание и удаление счетов. Позволяет вносить основные данные по счетам: сумма на счете, наименование счета. Также имеется возможность удалить счет, с сохранением всех транзакций. Это требуется для корректного сбора статистики.

Возможность добавления транзакций. По сути основная функциональность приложения без которой оно не имеет смысла. Можно вносить переводы между счетами, так и доходы и расходы по отдельным счетам.

Вывод анализ доходов и расходов является одной из ключевых функций приложения, позволяющей производить анализ финансов и планировать бюджет. Для получения статистики необходимо указать период времени, за который будут собраны данные.

Реализация добавления собственных категорий доходов и расходов необходима для получения более детальной статистики. При выводе статистики в процентном соотношении будет отображено распределение доходов и расходов по этим категориям. По умолчанию, при регистрации, для каждого пользователя создается категория доходов и расходов с названием "другое".

* 1. Средства реализации
     1. СУБД

В качестве СУБД было решено использовать PostgreSQL т. к. это один из наиболее популярных инструментов.

* + 1. Сервер

Для написания серверной части было принято решение использовать Java версии 17, поскольку это один из самых распространенных языков программирования, который часто выбирают для создания серверов.

Для упрощения разработки было принято решение использовать Spring Boot Framework, поскольку у него уже есть готовый набор инструментов для создания веб- и enterprise-приложений, таких как Spring контейнер, Spring Security и многие другие. Кроме того, с помощью Spring Boot удобно разрабатывать приложения, поддерживающие REST.

Для средства миграции базы данных было принято решение использовать LiquiBase, так как он удобен в использовании и изначально разрабатывался для языка Java.

Было принято решение использовать MyBatis для работы с данными, поскольку, по сравнению с аналогами, он лучше подходит для написания сложных запросов к базе данных и не генерирует запросы автоматически, что в некоторых случаях является преимуществом.

* + 1. Архитектура серверной части

Было решено придерживаться стандартной архитектуры Spring приложения (контроллеры, сервисы и репозитории), чтобы соответствовать требованиям Spring Boot. Для доступа к базе данных используются мапперы, описанные в виде интерфейсов, с реализацией, описанной в XML файлах.

* + 1. Клиент

Для написания клиентской части приложения был также использован язык Java 17, так как помимо инструментов для создания серверных приложений, он обладает и инструментами для написания клиентских приложений, что в свою очередь обеспечивает удобство поддержки как серверной, так и клиентской части системы.

Также для удобного и сложного парсинга запросов, было решено использовать библиотеку Gson, так как она позволяет с легкостью разбирать сложные JSON объекты, например списки.

Выбором фреймворка для разработки клиентской части стал Vaadin версии 23.2.6. Этот фреймворк позволяет создавать веб-приложения полностью на Java, что обеспечивает использование одного и того же удобного инструмента для разработки.

* + 1. Архитектура клиентской части

Фреймворк Vaadin также удобен тем, что любое Vaadin-приложение по умолчанию является Spring Boot приложением, что означает использование архитектуры Spring. В случае клиентского приложения, где нет прямого взаимодействия с базой данных и контроллеры не обрабатывают запросы, стандартная архитектура меняется. Поэтому приложение будет состоять из представлений (views), слушателей (listeners) — основных частей приложения. Также предполагается наличие сущностей (entities) и объектов DTO.

1. Реализация
   1. Диаграмма вариантов использования

На Рисунке 1 изображена диаграмма прецедентов для неавторизованного пользователя. Единственное, что может сделать такой пользователь — это войти в систему (указать свои учетные данные), либо зарегистрироваться в системе, для дальнейшего входа в систему.

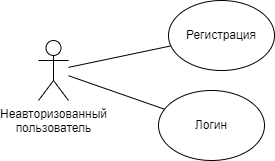
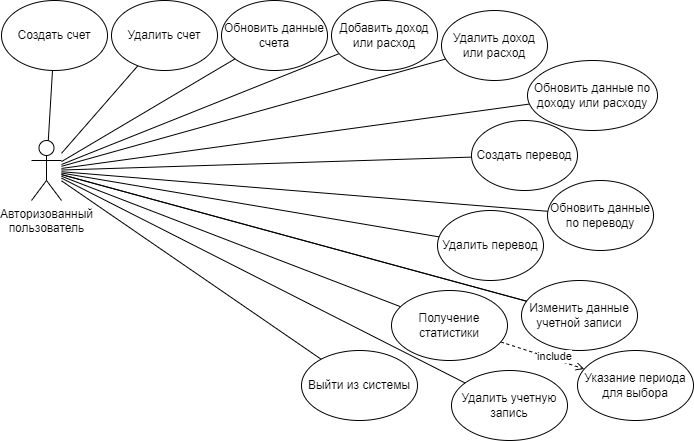


Рисунок 1 - Диаграмма прецедентов для неавторизованного пользователя

На Рисунке 2 изображена диаграмма прецедентов для авторизованного в системе пользователя. Ему доступен весь функционал приложения.

Рисунок 2 - Диаграмма прецедентов для авторизованного пользователя

* 1. Диаграмма классов

На Рисунке 3 изображена диаграмма классов приложения. На ней указаны все классы являющиеся сущностями. Они прямо отражают структуру базы данных и являются представлением всех основных таблиц.

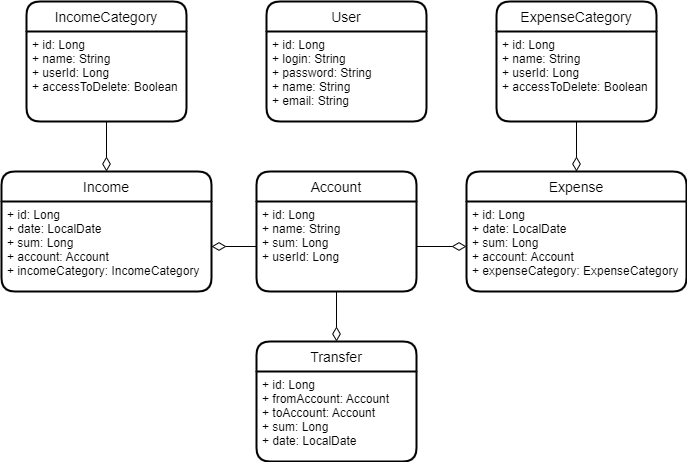


Рисунок 3 - Диаграмма классов приложения

* 1. Диаграмма таблиц базы данных

На Рисунке 4 изображена диаграмма базы данных. На ней указаны как основные таблицы, реализующие бизнес-логику приложения (user, account, transfer, income\_category, expense\_category, income, expense), так и таблицы отвечающие за работу дополнительных систем (databasechangelog, databasechangeloglock нужны для работы LiquiBase).

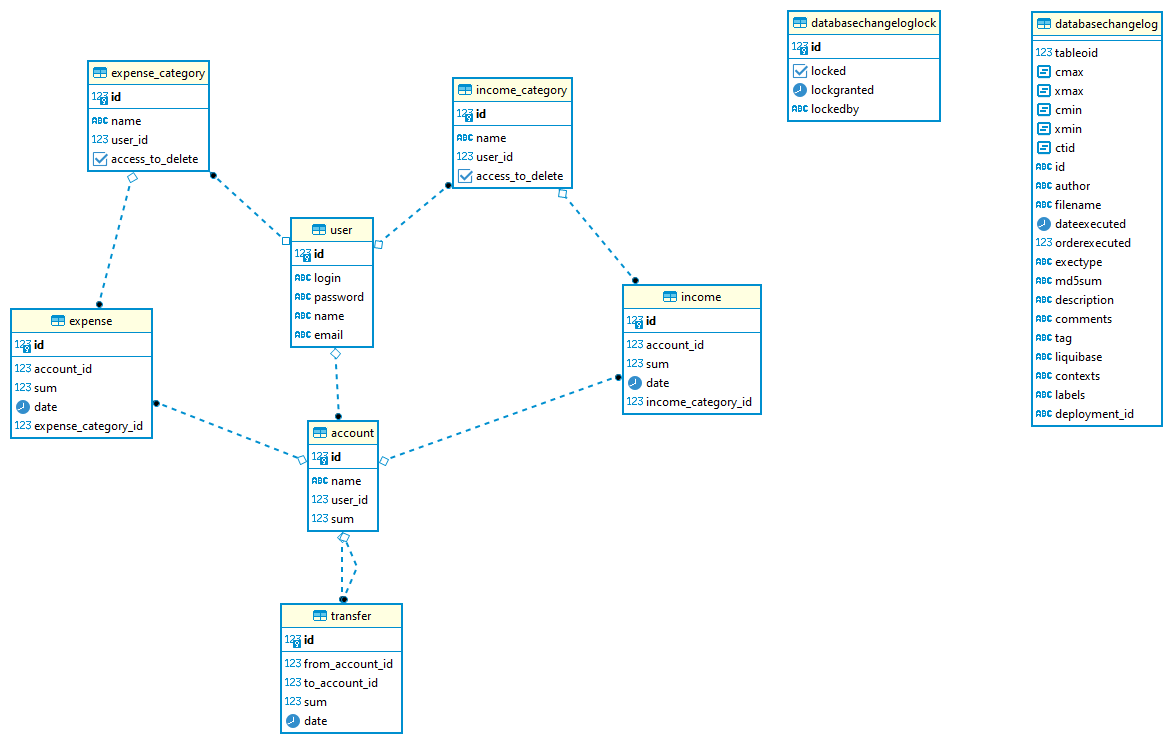


Рисунок 4 - Диаграмма базы данных

* 1. Реализация интерфейса

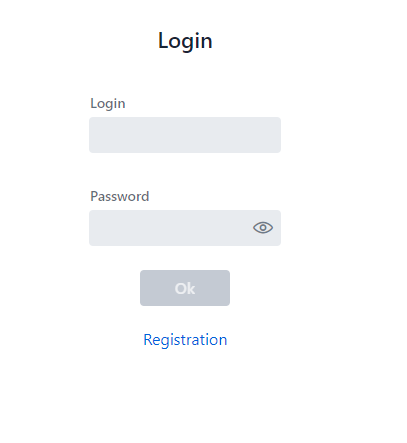


Рисунок 5 - Страница входа

Первое, что видит пользователь, когда заходит в приложение, это страница входа (Рисунок 5).

По умолчанию кнопка "ОК" не активна и недоступна для нажатия. Для ее активации необходимо заполнить поля "Логин" и "Пароль". После этого кнопка становится доступной для нажатия, и пользователь может войти в систему, если введенные данные верны. (Рисунок 6).

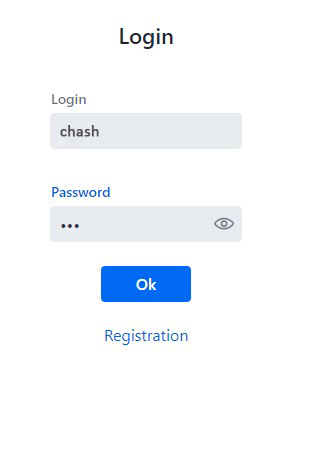


Рисунок 6 - Страница входа после активации кнопки Ok

Также можно перейти на страницу регистрации (Рисунок 7) по указанной ниже ссылке Registration.

Также как и на странице входа, по умолчанию кнопка для подтверждения регистрации Ok – неактивна. Соответственно для ее активации необходимо заполнить все пустые поля(Email, Password, Confirm password, Login, Name). Итоговый вид на Рисунке 8.

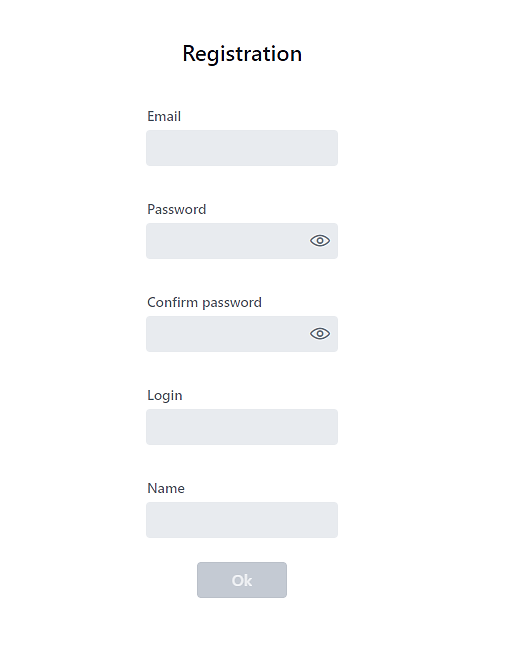


Рисунок 7 - Страница регистрации

После успешного входа в систему, пользователь перенаправляется на страницу своего профиля (Рисунок 9). Вверху страницы указано имя пользователя, указанное при регистрации. Далее расположены кнопки для взаимодействия со своим профилем: "Edit data" для редактирования данных пользователя, "Change password" для смены пароля, "Exit" для выхода из учетной записи (уничтожается сессия на стороне сервера и пользователь перенаправляется на страницу входа) и "Delete profile" для безвозвратного удаления пользователя и всех его данных.

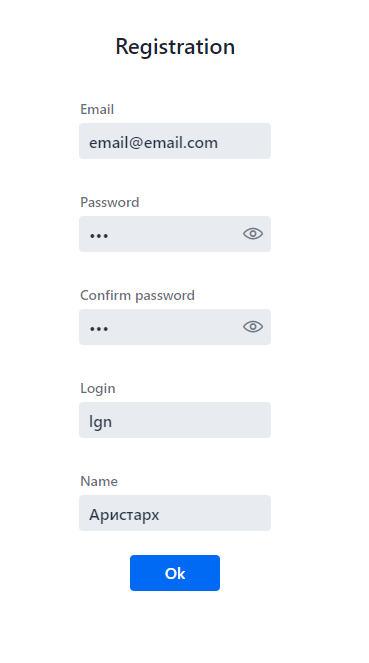


Рисунок 8 - Страница регистрации с активной кнопкой "Ok"

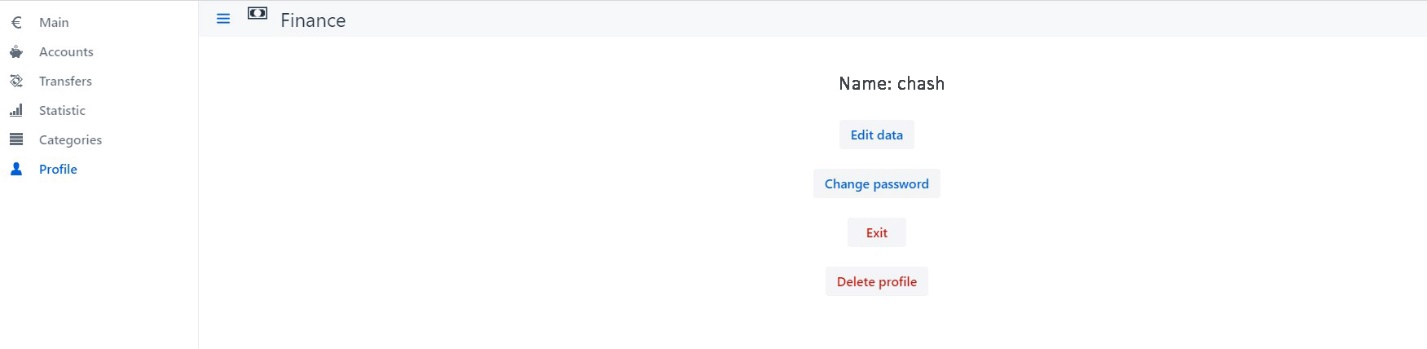
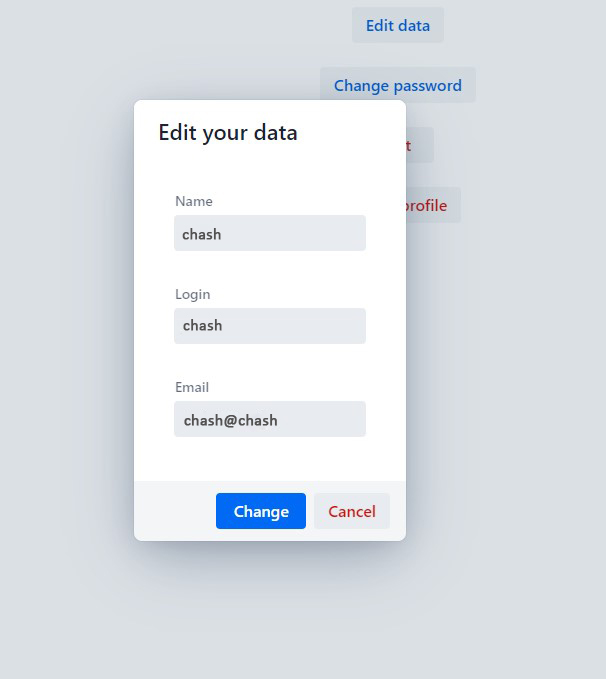


Рисунок 9 - Страница профиля пользователя

При нажатии кнопки Edit data появляется диалоговое окно (Рисунок 10) на котором указаны все доступные для редактирования пользовательские данные(Email, Login, Name). Кнопка Change изначально активна, но в том случае если какое-то из полей будет пустым или значение в поле Email не будет соответствовать формату почты, то кнопка перестанет быть активной.

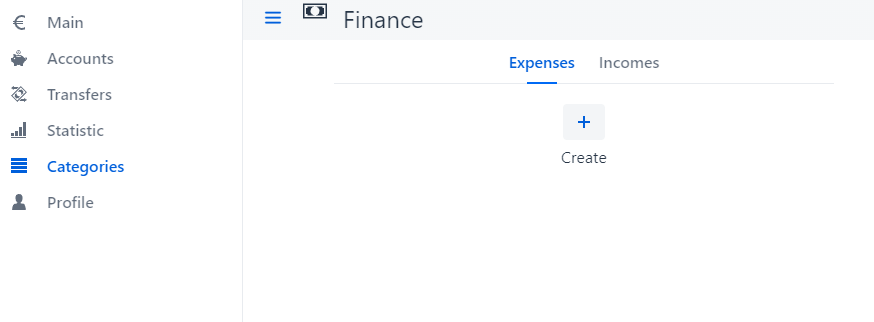
Рисунок 10 - Диалоговое окно для редактирования данных пользователя 

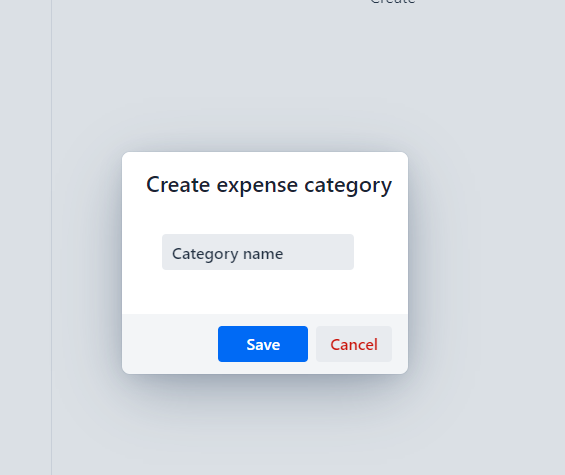
На левой стороне расположено меню, где доступна свободная навигация, позволяющая перемещаться между всеми страницами приложения.

Также имеется страница категорий для расходов (Expense) и доходов (Income), изображенная на Рисунке 11. Изначально у пользователя отсутствуют собственные категории (не считая созданных по умолчанию категорий "Other" для доходов и расходов, которые не отображаются пользователю в списке категорий). Для создания собственных категорий необходимо нажать кнопку "Create".

Рисунок 10 - Диалоговое окно для редактирования данных пользователя

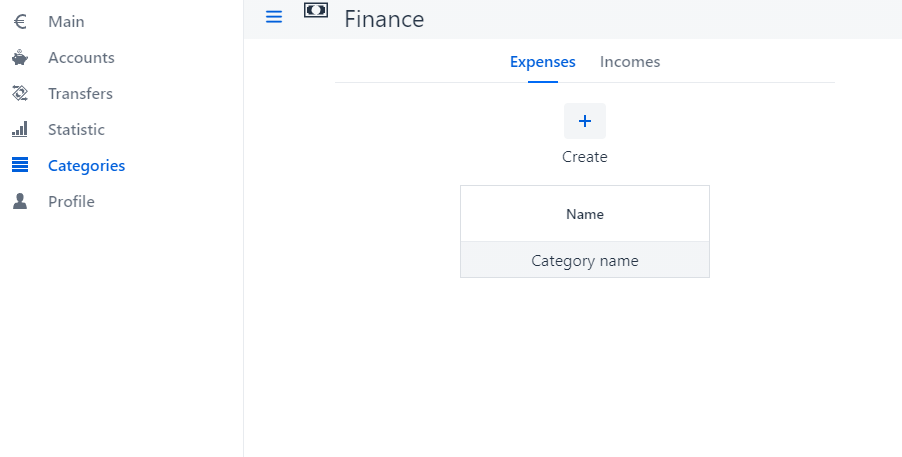
При нажатии на эту кнопку появляется диалоговое окно для создания новой категории (Рисунок 12). Здесь требуется указать только одно поле "Category name". Как и в случае с другими диалоговыми окнами, кнопка "Save" изначально активна, однако если поле с названием категории будет пустым, то и кнопка станет неактивной.

Рисунок 11 - Страница доходов и расходов

Рисунок 12 - Диалоговое окно для создания новой категории

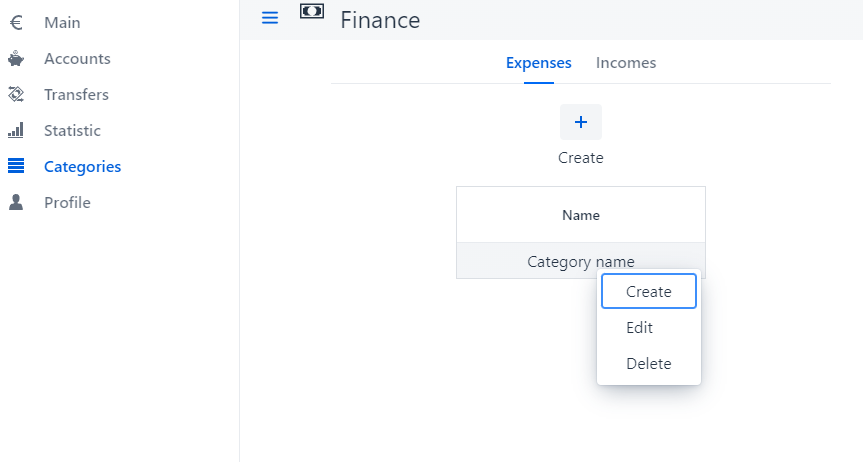
После создания новой категории расходов, страница категории будет выглядеть следующим образом (Рисунок 13).

Как можно увидеть, появляется таблица категорий, с единственным столбцом — название категории. Чтобы тем или иным образом взаимодействовать с данными в этой таблице необходимо нажать правой кнопкой мыши по таблице(если страница открыта на телефоне то удерживать ту строку, с которой необходимо произвести манипуляции).

Рисунок 13 - Страница категорий после создания своей категории расходов

После этого появится контекстное меню с тремя кнопками (Рисунок 14). "Create" для создания новой категории, "Edit" для редактирования данных выбранной категории и "Delete" для удаления категории. При выборе опции "Edit" появляется аналогичное диалоговое окно, как и для создания, но вместо значений по умолчанию отображается текущее название категории. Для окна редактирования также действуют те же правила, что и для создания(кнопка активна только тогда, когда все поля заполнены).

На странице счетов (Рисунок 15) есть таблица, в которой указаны все счета пользователя. Имеются столбцы с названиями счетов и суммой на счете. Для создания, редактирования и удаления также по нажатию правой кнопки мыши, открывается контекстное меню, с кнопками для создания(Create), изменения(Edit) и удаления(Delete) счетов.

Рисунок 14 - Контекстное меня для таблицы категорий

По нажатию на кнопки создания и редактирования открываются диалоговые окна (Рисунок 16) в которых указано два поля: название счета и сумма на счете. Аналогично другим окнам, кнопка для подтверждения операции активна только если все поля непустые.

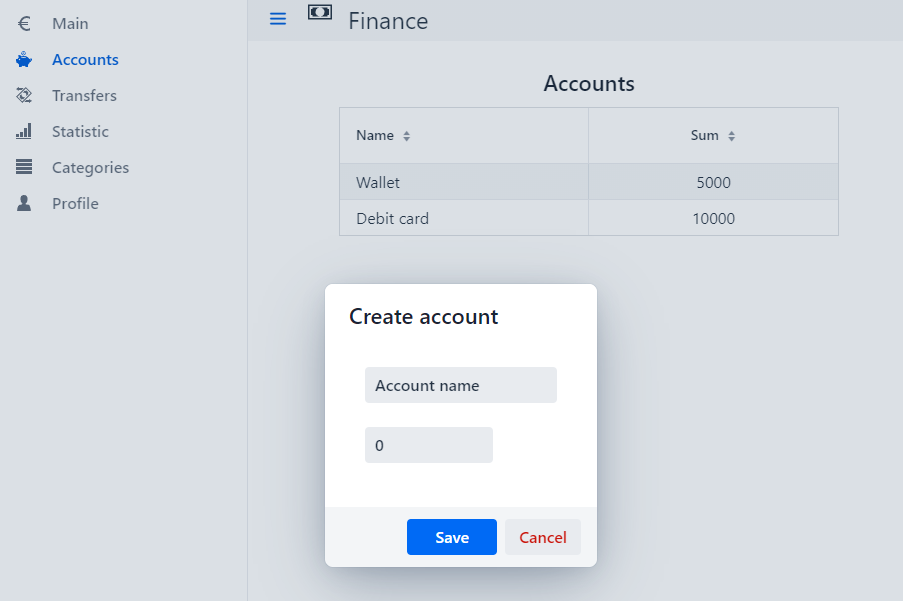
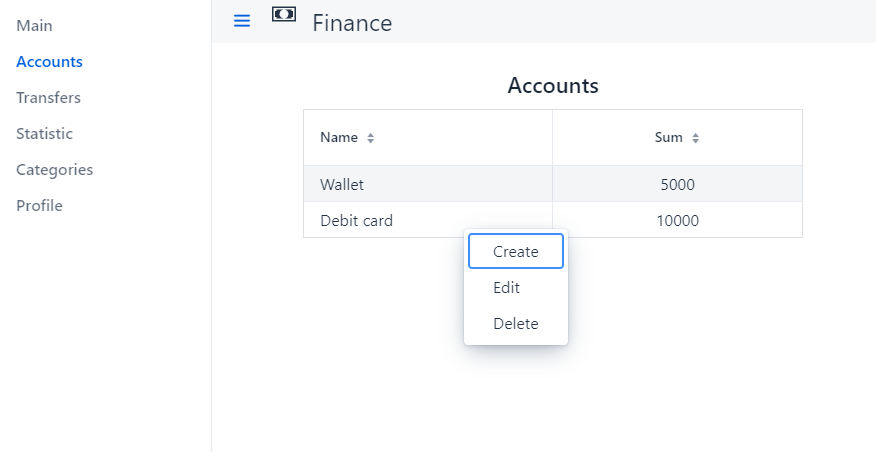


Рисунок 16 - Диалоговое окно для создания счета

Рисунок 15 - Страница счетов пользователя с открытым контекстным меню

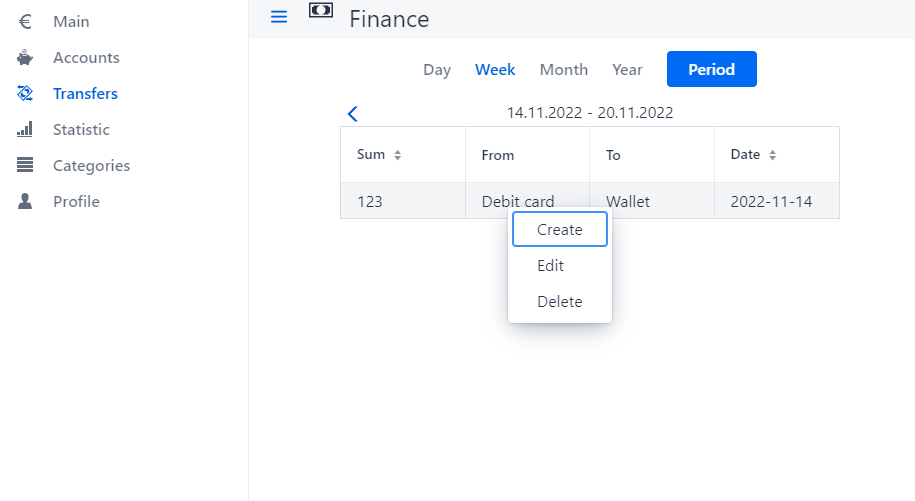
Следующей страницей является страница переводов (Рисунок 17). На ней есть таблица со всеми переводами. Также имеется возможность навигации по временным периодам (день, неделя, месяц, год и свободно настраиваемым периодом). Для работы с переводами необходимо также нажать правой кнопкой мыши по таблице и в открытом контекстном меню выбрать желаемое действие. При создании и редактировании открывается диалоговое окно (Рисунок 18) работающее по тем же принципам, что и предыдущие. Разница заключается в том, что если у пользователя имеется всего один счет или их вообще нет, то переводы становятся недоступны и при переходе на эту страницу пользователь получает соответствующее сообщение.

Рисунок 17 - Страница переводов с открытым контекстным меню

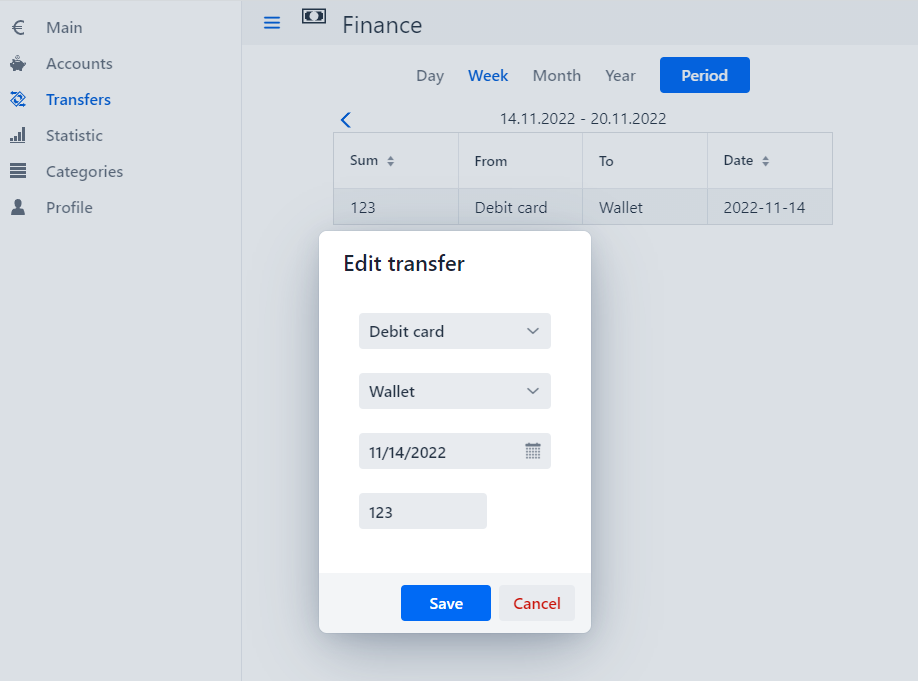


Рисунок 18 - Диалоговое окно для изменения данных перевода

Самой главной страницей является страница Main (Рисунок 19), на которой происходит вся работа с доходами и расходами, с использованием разнообразных виджетов. Имеется навигация как между доходами и расходами, так и между временными промежутками аналогично переводам. Также присутствует таблица, в которой указаны расходы и доходы, в зависимости от выбранной вкладки(Expenses, Incomes). Для работы с ней используется сценарий аналогичный предыдущим (использование контекстного меня вместе с диалоговым окном (Рисунок 20)).

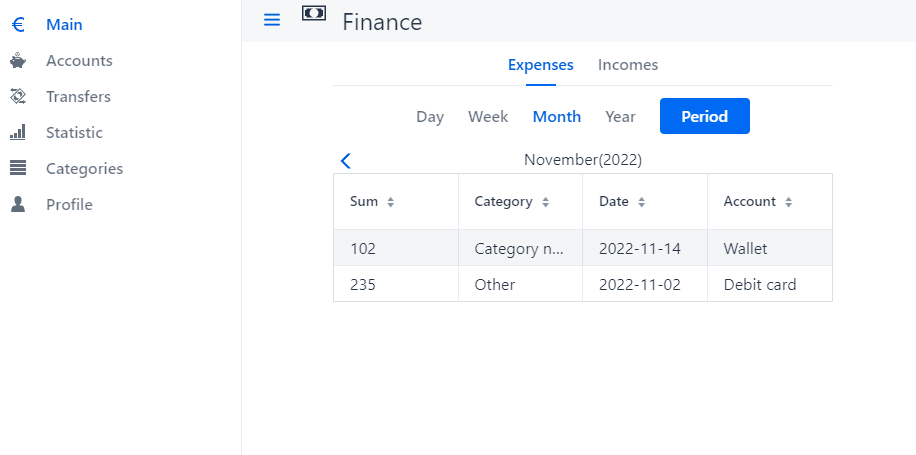


Рисунок 19 - Главная страница (доходов и расходов)

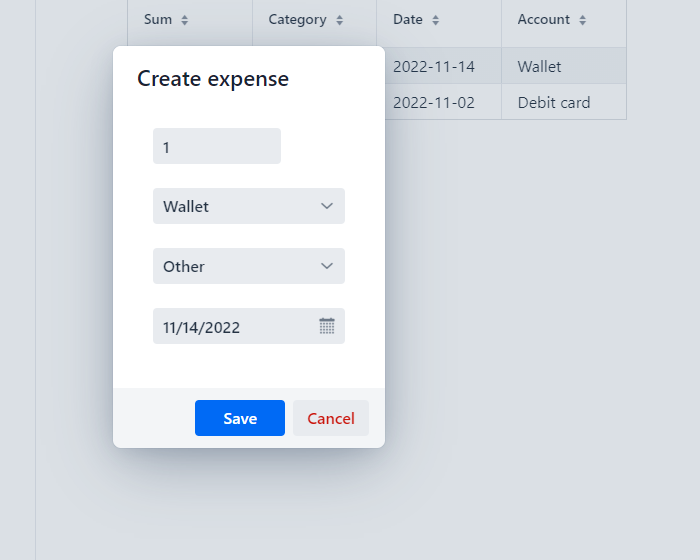
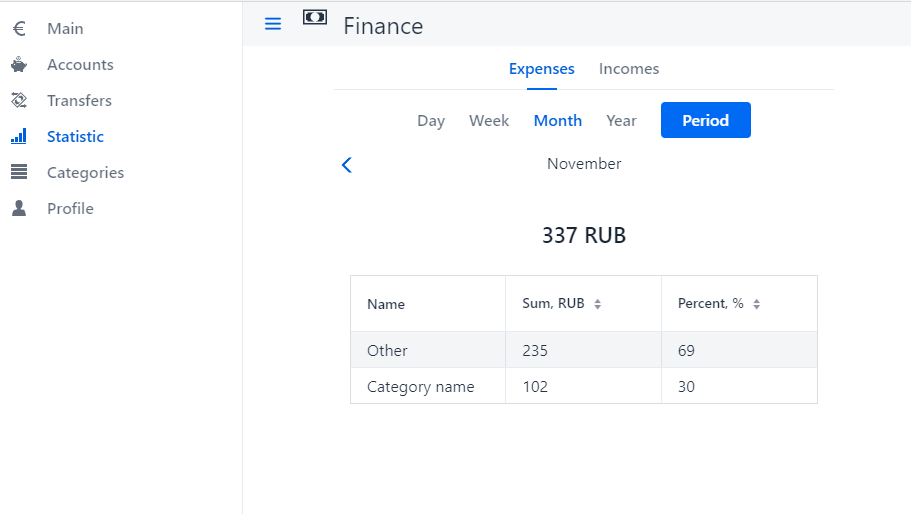


Рисунок 20 - Диалоговое окно для создания расхода

На странице статистики (Рисунок 21) можно смотреть статистику по доходам и расходам за любой указанный период времени, аналогично переводам. Статистика выводится в виде таблице, в которой три столбца: название категории дохода или расхода, сумма (весь доход или расход по данной категории), процент от общей суммы доходов или расходов. Поскольку это данные являющиеся результатом расчетов, то поэтому вся таблица неизменяемая, и является доступной только для просмотра.

Рисунок 21 - Страница статистики по доходам и расходам

Заключение

Перед началом разработки был проведен анализ предметной области, в результате которого были выявлены преимущества и недостатки найденных аналогов. Эта информация была учтена в процессе разработки данного приложения.

Также в процессе разработки приложения были расширены и углублены знания в области использования различных инструментов разработки. Были детально изучены методы их применения для создания подобных приложений.

В ходе разработки были успешно завершены все этапы проекта, в результате чего были выполнены все предъявленные требования.