

Содержание

Введение.....	3
1. Программные средства и технологии, применяемые на предприятии.....	4
1.1 Платформа ASP .NET Core.....	4
1.2 JetBrains WebStorm.....	4
1.3 JetBrains Rider	5
1.4 GitHub	5
1.5 Язык программирования C#.....	6
2. Аналитический обзор.....	8
2.1 Анализ аналогов	8
2.1.1 Приложение «Money Lover».....	8
2.1.2 Приложение «Spendee»	9
3. Проектирование, дизайн и функционал создаваемого приложения.....	11
3.1 Цели и задачи проекта	11
3.2 Целевая аудитория	11
3.3 Дизайн приложения	11
Заключение	23
Список использованных источников	24

Введение

Преддипломная практика для специалистов высшей квалификации — это возможность продемонстрировать свою теоретическую подготовку и практические навыки, полученные в ходе обучения. Получение обратной связи со стороны компаний и организаций, принимающих студентов на практику, приобретение опыта и знаний над чем нужно поработать, чтобы соответствовать современным требованиям рынка труда.

Цель данной преддипломной практики заключается в демонстрации своих теоретических знаний и практических навыков на примере разработки дипломного проекта.

Целью дипломного проекта является проектирование и разработка веб-приложения для управления личными финансами пользователей, включающего серверную и клиентскую части.

В ходе преддипломной практики были выполнены следующие задачи:

- 1) анализ предметной области, приложений-аналогов;
- 2) выбор дизайна, методов и технологий;
- 3) проектирование структуры приложения, структуры базы данных, осуществление навигации по приложению;
- 4) выполнение программной реализации проекта;

Преддипломная практика проходила в Белорусском государственном технологическом университете с 22 марта 2021 года по 16 апреля 2021 года.

Белорусский государственный технологический университет осуществляет подготовку специалистов различных специальностей. В сфере информационных технологий это: информационные системы и технологии, программное обеспечение информационной безопасности мобильных систем, программное обеспечение информационных технологий, дизайн электронных и веб-изданий.

1. Программные средства и технологии, применяемые на предприятии

Итогом данной преддипломной практики является веб-приложение для управления личными финансами. Каждый этап создания расписан в дневнике практики с указанием даты выполнения. В первые дни было проведено ознакомление с техникой безопасности и правилами работы.

Для прохождения преддипломной практики в университете были выделены места в аудитории. Аудитория была оснащена компьютерами с некоторым необходимым программным обеспечением.

1.1 Платформа ASP .NET Core

Для написания сервера приложения была выбрана платформа ASP .NET Core. Для обоснования выбора отметим отличительные особенности данной технологии. ASP .NET Core — фреймворк с открытым исходным кодом для платформы .NET. Данный фреймворк предоставляет огромный набор инструментов для проектирования различного рода серверов, сервисов и других веб-приложений на базе платформы .NET. На данный момент более широко используется версия для .NET Core по ряду причин: платформа с открытым исходным кодом, кроссплатформенность, возможность self-хостинга и многое другое.

Разрабатывать приложения на основе ASP .NET Core можно с помощью таких языков программирования как VB и C#. В моем случае я вел разработку на языке C#, т.к. он имеет привычный C-подобный синтаксис, множество функций для удобной разработки на уровне языка, а так же огромную фан-базу, что позволило быстро находить ответы на проблемы в процессе реализации проекта.

Для построения серверной части приложения я выделил для себя GraphQL подход. Принцип разработки GraphQL API отличается от привычного REST, однако предоставляет массу преимуществ в разработке и поддержке API.

С помощью NuGet пакетного менеджера была найдена библиотека HotChocolate для построения GraphQL сервиса на базе ASP .NET Core. Библиотека достаточно хорошо документированна и активно поддерживается самой компанией Microsoft.

1.2 JetBrains WebStorm

JetBrains WebStorm — интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами (в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на

Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) — то есть в таких конструкциях поддерживается корректный рефакторинг. Данная среда разработки доступна для Windows, OS X и Linux.

Особенности данного продукта:

1. Сборка приложений на Node.js.
2. Модификация js файлов с одновременным просмотром результатов
3. Рефакторинг кода
4. Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
5. Интеграция с системами управления версиями.
6. Возможность расширения плагинами.
7. Отладка кода на JavaScript.

1.3 JetBrains Rider

JetBrains Rider — кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#. Проект анонсирован в январе 2015 года. В его основе лежит другой продукт JetBrains — ReSharper. Среда поддерживает платформы .NET Framework, .NET Core и Mono. Работает на операционных системах Windows, MacOS, Linux. Дизайн среды ориентирован на продуктивность работы программистов, позволяя сконцентрироваться на функциональных задачах, в то время как Rider берёт на себя выполнение рутинных операций.

Среди прочих возможностей, среда хорошо совместима со многими популярными свободными инструментами разработчиков, такими как VCS, Entity Framework, NuGet и XUnit.

Rider поддерживает .NET Framework, новую платформу .NET Core и проекты на основе Mono. IDE позволяет разрабатывать десктопные приложения, .NET-сервисы и библиотеки, игры на движке Unity, мобильные приложения Xamarin, веб-приложения ASP.NET и ASP.NET Core.

Rider предоставляет более 2200 инспекций кода, сотни контекстных действий и рефакторингов, заимствованных из ReSharper, в сочетании с продвинутой функциональностью сред разработки на основе платформы IntelliJ. Несмотря на большой набор функций, Rider — быстрая и отзывчивая IDE, поэтому мой выбор пал именно на эту IDE.

1.4 GitHub

GitHub — крупнейший веб-сервис для хранения исходного кода IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc.

Сервис абсолютно бесплатен для проектов с открытым исходным кодом и предоставляет им все возможности, а для частных проектов предлагаются различные платные тарифные планы.

Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева.

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок. Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

- На платных тарифных планах можно создавать приватные репозитории, доступные ограниченному кругу пользователей.
- Есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса.
- Код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта.
- Кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial.
- На сайте есть pastebin-сервис gist.github.com для быстрой публикации фрагментов кода.

1.5 Язык программирования C#

C# — современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать множество типов безопасных и надежных приложений, работающих в экосистеме .NET.

C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. Здесь представлен обзор основных компонентов языка C# 8 и более ранних версий.

C# — это объектно- и компонентно-ориентированный язык программирования, который предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО.

Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения:

- Сборка мусора — автоматически освобождает память, занятую недоступными неиспользуемыми объектами.
- Типы, допускающие значение `null` — обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты.
- Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них.

- Лямбда-выражения поддерживают приемы функционального программирования.
- Синтаксис LINQ создает общий шаблон для работы с данными из любого источника.
- Все типы C#, включая типы-примитивы, такие как `int` и `double`, наследуют от одного корневого типа `object`. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений.
- Динамическое выделение объектов и хранение упрощенных структур в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность.
- Итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

2. Аналитический обзор

2.1 Анализ аналогов

Для создания приложения с уникальным дизайном, удобным интерфейсом и всеми необходимыми техническими средствами, нужно тщательно изучить уже представленные в интернете аналоги. Это даст возможность выделить успешные дизайнерские решения, организовать структуру страниц, определить необходимые модули, выделить все достоинства и недостатки, что в результате поможет создать лучший вариант представления для разрабатываемого клиента.

Одними из важнейших аспектов при оценке аналогов были функциональность и юзабилити, так как это определяет дальнейшее отношение пользователей к системе. Функциональность подразумевает набор возможностей (функций), которые предоставляет данная система или устройство.

Юзабилити, также удобство использования, пригодность использования, эргономичность — способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях.

2.1.1 Приложение «Money Lover»

В качестве одного из прототипов было выбрано приложение «MoneyLover». MoneyLover — настоящем комбайне для всех любителей считать деньги. В отличие от многих других утилит, MoneyLover — не просто про учет расходов или планирование бюджета. Программа охватывает практически все возможные сценарии использования денег, а стало быть — большинство ситуаций, которые могут возникнуть на жизненном пути.

Существуют версии этого приложения для всех платформ, в том числе и веб-версия.

Приложение предназначено для учёта и анализа собственных расходов и доходов. Приложение имеет довольно красивый и понятный любому пользователю интерфейс и функционал.

Пользовательский интерфейс приложения продемонстрировано на рисунке 1.1.

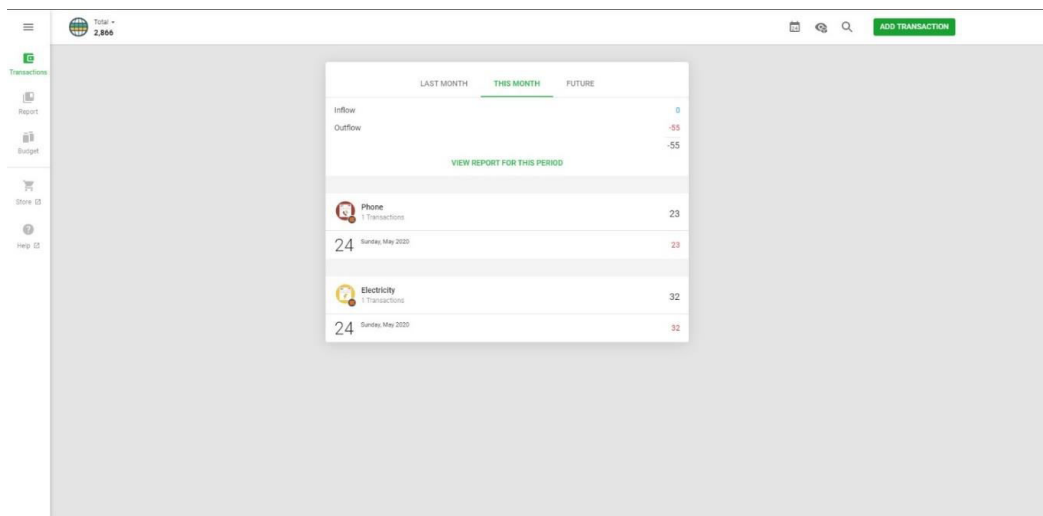


Рисунок 1.1 – Пользовательский интерфейс приложения «MoneyLover»

Достоинства приложения:

- богатые функциональные возможности;
- продвинутое управление счетами и категориями;
- функция сканирования чеков;
- учет долгов и постановка финансовых целей;
- учет задолженности и доступного лимита по кредитным картам;
- совместный учет финансов;
- синхронизация транзакций с PayPal;
- автоматическая и ручная конвертация валюты;
- информативный виджет.

Недостатки приложения:

- средний уровень безопасности;
- рекламные окна, мешающие работе;
- необходимость регистрации.

2.1.2 Приложение «Spendee»

Spendee выделяется на фоне остальных приложений красивым интерфейсом. Здесь нет унылых таблиц, напоминающих о скучной бухгалтерской работе. Вместо этого Spendee предлагает удобный и привлекательный UI, чем-то напоминающий ленту в соцсети. Ваши доходы и расходы будут представлены в виде красивой инфографики, так что вы сможете с лёгкостью понять, что происходит с вашими деньгами.

Приложение Spendee существует с версиями под все мобильные устройства, в том числе и веб-версия.

Отличительной особенностью является возможность привязки банковских счетов для автоматического заполнения транзакций по счетам.

Пользовательский интерфейс приложения продемонстрирован на рисунке 1.2 ниже.

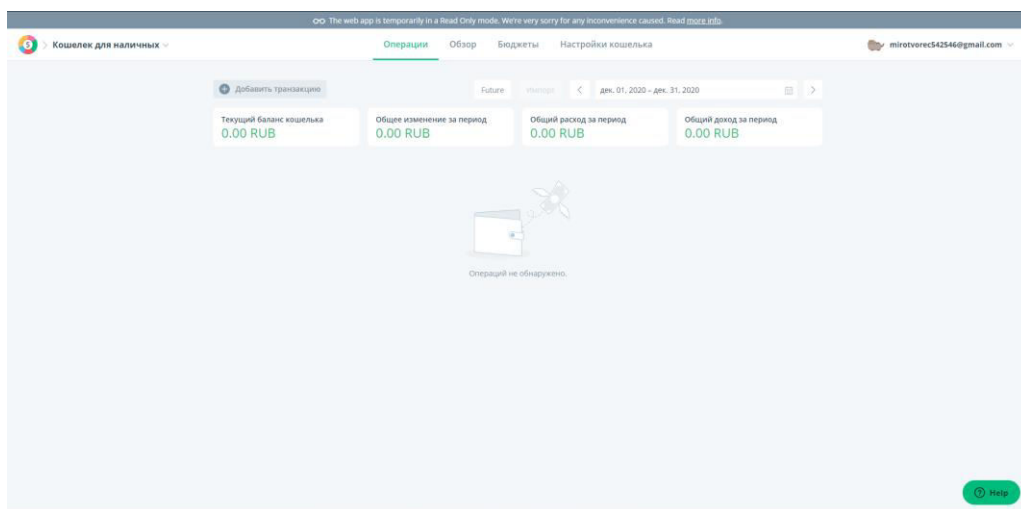


Рисунок 1.2 – Пользовательский интерфейс приложения «Spendee»

Так же приложение имеет возможность создавать кошельки в разной валюте, добавлять транзакции вручную, а так же просматривать статистику за определенный промежуток времени.

Достоинства приложения:

- в приложении можно спланировать расходы по каждой отдельной категории;
- можно вести совместный бюджет с другими аккаунтами.

Недостатки приложения:

нет возможности привязать к аккаунту больше одного кошелька.

3. Проектирование, дизайн и функционал создаваемого приложения

3.1 Цели и задачи проекта

Перед разработкой модуля следует выделить перечень задач для выполнения:

- 1) Обеспечить регистрацию одиночных и повторяющихся денежных транзакций.
- 2) Обеспечить создание и управление кошельками и категориями транзакций.
- 3) Обеспечить визуализацию статистики транзакций пользователя за период времени.
- 4) Обеспечить возможность настройки параметров работы приложения пользователя.

Таким образом, цель данного проекта заключается в разработке сервиса, удовлетворяющего вышесказанным критериям.

3.2 Целевая аудитория

Целевая аудитория — это люди, удовлетворяющие ту потребность, которую решает ваш продукт или услуга.

Основной аудиторией являются платежеспособные люди в возрасте от 18 до 40 лет, активно использующие социальные сети и сеть Internet.

Исходя из данных, собранных в процессе анализа аналогов, был разработан дизайн приложения и архитектура модуля. Скриншоты будут предоставлены в соответствии с функционалом выполненной работы.

3.3 Дизайн приложения

На рисунке 3.1 представлена стартовая страница приложения.

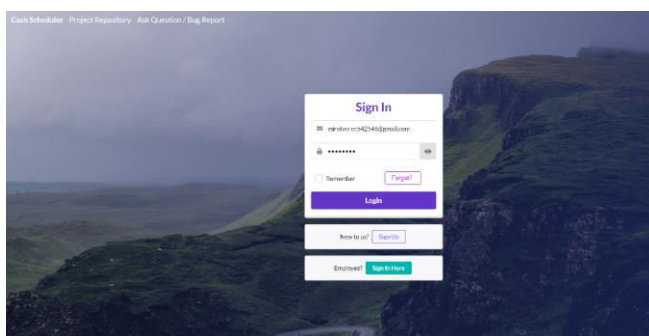


Рисунок 3.1 – Стартовая страница приложения

Здесь можно войти в систему, зарегистрироваться, оставить обратную связь, а также восстановить забытый пароль путём подтверждения аккаунта по электронной почте.

На рисунке 3.2 представлена возможность создать транзакцию.

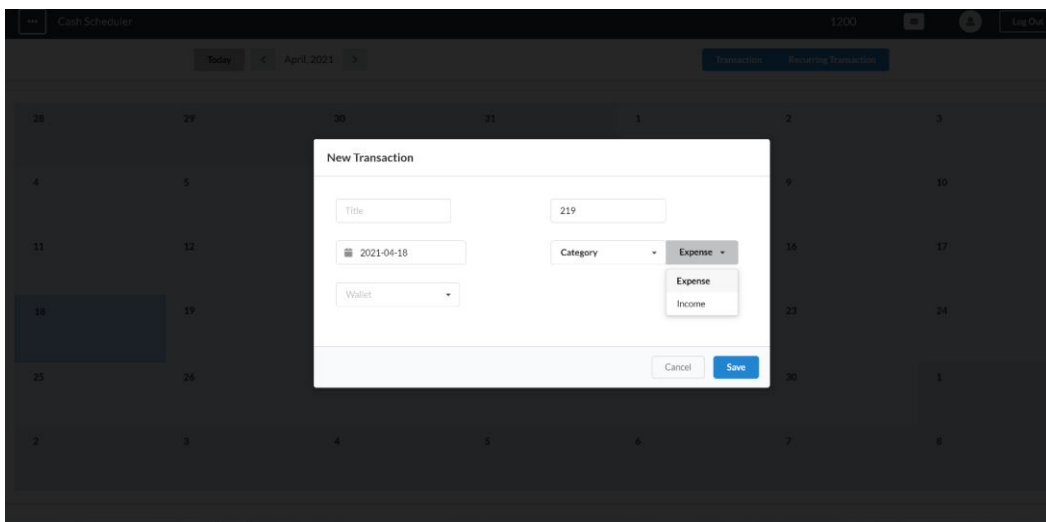


Рисунок 3.2 – Создание транзакции

Здесь можно выбрать вид транзакции расход это, либо доход, в зависимости от этого будет выбор категории. Возможность выбора категории продемонстрирована на рисунке 3.3.

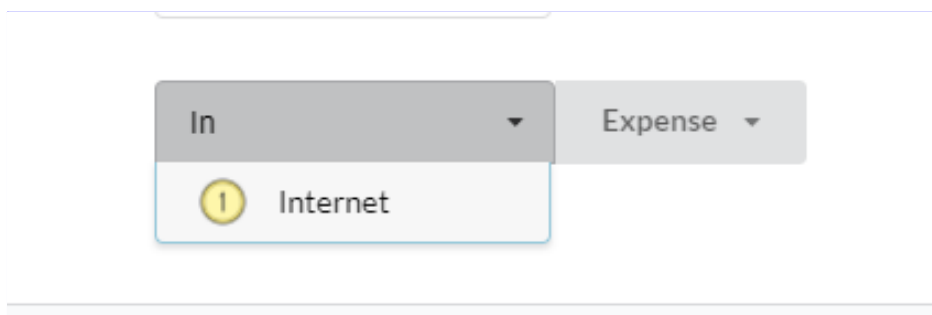


Рисунок 3.3 – Категории

Существуют стандартные категории, но пользователь также может создавать свои (рисунок 3.4).

Рисунок 3.4 – Создание категории

Так же, при создании транзакций, необходимо привязать её к одному из ваших кошельков. По умолчанию при регистрации создается кошелёк по умолчанию, но вы можете создавать их сколько угодно в какой угодно валюте.

Пример создания нового кошелька представлен на рисунке 3.5.

Рисунок 3.5 – Создание кошелька

При создании кошелька, вы можете пометить его как основной, указать нужную валюту, баланс, а также задать ему имя.

Иногда вам может понадобиться перекинуть какую-то сумму с одного кошелька на другой. И такая возможность представлена на рисунке 3.6.

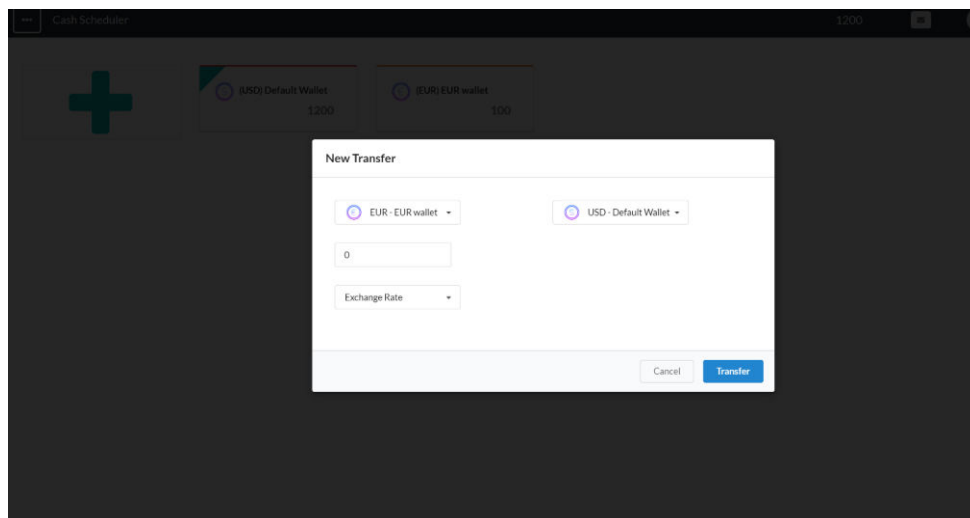


Рисунок 3.6 – Трансфер с одного кошелька на другой

Здесь пользователь указывает откуда производить трансфер и куда, сумму в изначальной валюте и курс обмена валюты. По умолчанию курс обмена получается через публичный API европейского банка, однако, если курс пользователя отличается, он может добавить свой курс (рисунок 3.7) и в будущем переиспользовать его для такой же пары валюты (откуда - куда).

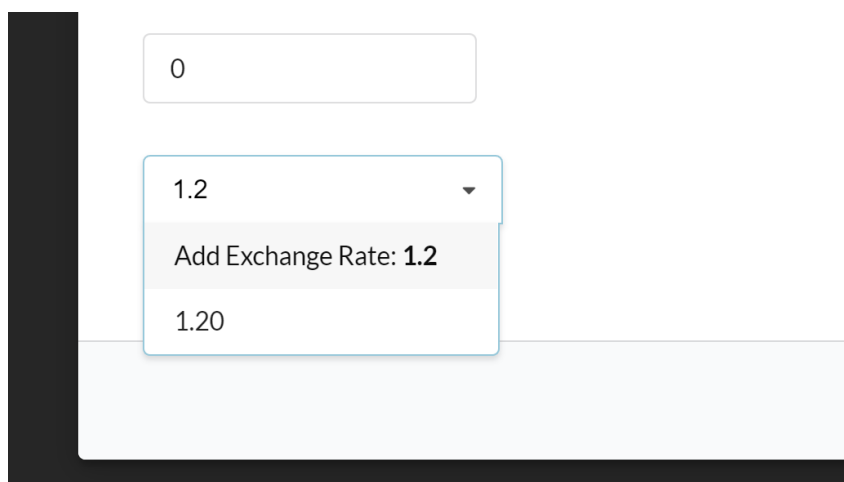


Рисунок 3.7 – Добавление курса обмена

После совершения трансфера, балансы кошельков автоматически обновятся.

Вместе с тем, присутствует возможность поменять валюту уже существующего кошелька с возможностью конвертации средств в соответствии с курсом (рисунок 3.8).

Edit Wallet

EUR wallet 100

ETB Is Default?

50.09 Convert Balance?

Cancel Delete Save

Рисунок 3.8 – Смена валюты у существующего кошелька

В таком случае пользователю будет предложено выбрать курс обмена валют по той же логике, что и ранее.

Присутствует возможность просмотра отчетности по тому, на какие категории и в каком количестве были созданы транзакции за определенный месяц. Чтобы перейти на данную страницу просто нужно нажать в меню слева на раздел «Transactions». На этой странице пользователь может просматривать отчеты трат, доходов и их соотношения по месяцам (рисунок 3.9)

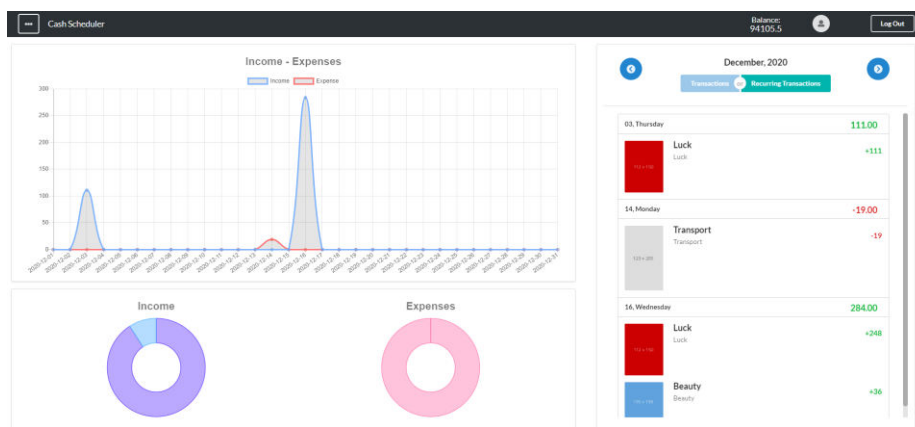


Рисунок 3.9 – Страница транзакций

Можно посмотреть, как много было потрачено или заработано на определенную категорию за выбранный месяц. Это можно увидеть на графике внизу страницы, показано на рисунке 3.10.



Рисунок 3.10 – Соотношение расходов и доходов по категориям

Следующий же график показывает траты и доходы за каждый день выбранного месяца. Он находится вверху страницы транзакций, пример его продемонстрирован на рисунке 3.11.

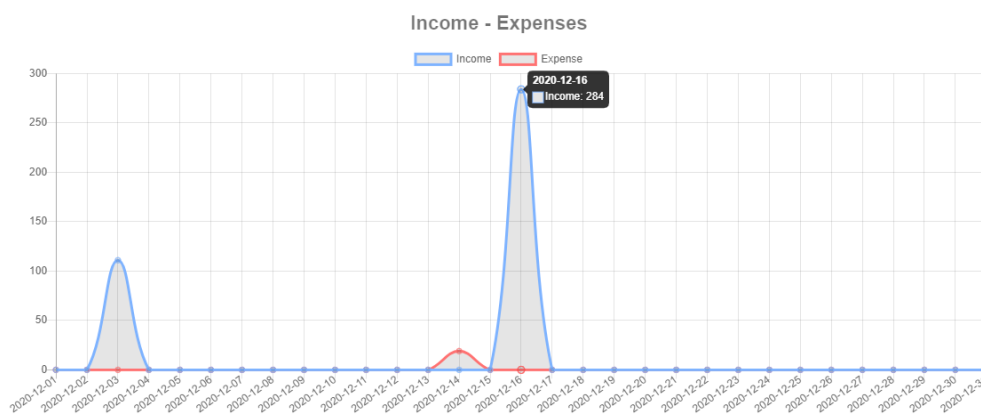


Рисунок 3.11 – Количество расходов и доходов по дням

На странице транзакций справа можно увидеть список транзакций за выбранный месяц сгруппированный по дням месяца.

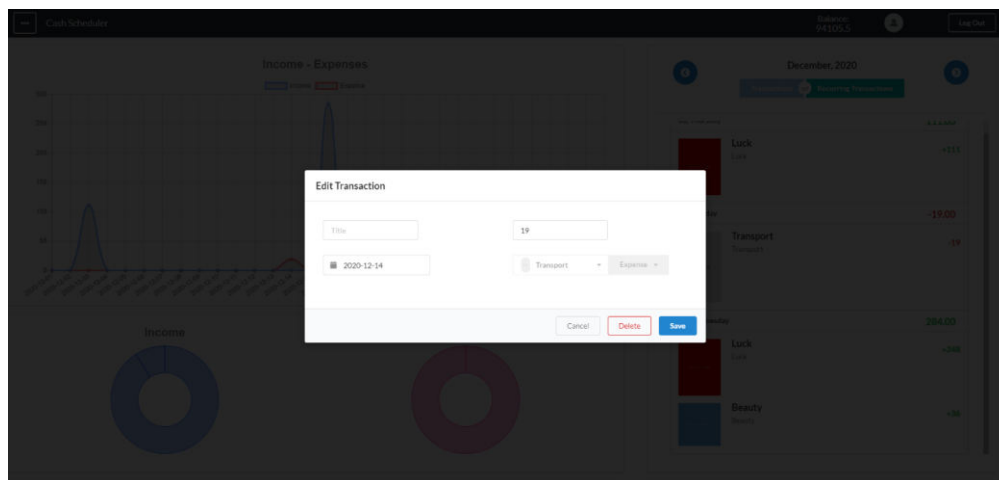


Рисунок 3.12 – Редактирование транзакций

Реализована возможность для пользователя выбрать какую-либо транзакцию из списка и отредактировать либо ее заголовок, либо сумму, либо дату, изменять категорию, к которой относится транзакция, нельзя, однако предусмотрено удаление транзакции (рисунок 3.12).

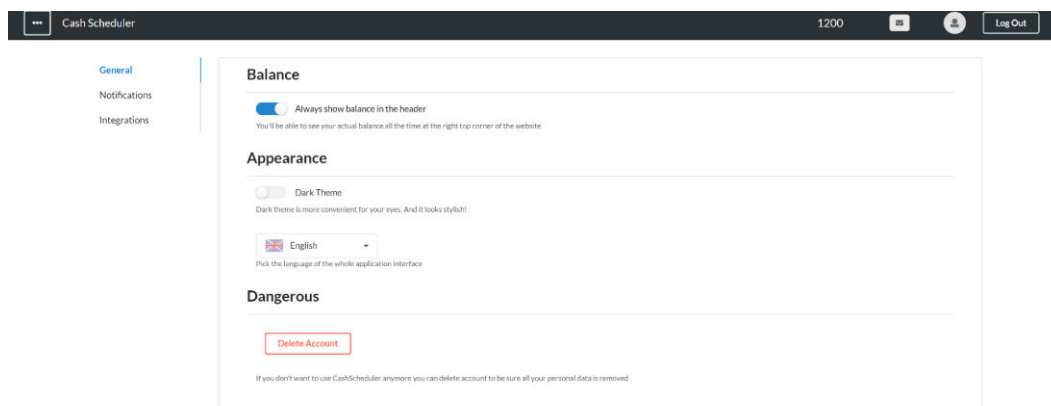


Рисунок 3.13 – Первый раздел настроек

Для удобства пользования приложением, пользователю предоставлена возможность редактировать некоторые параметры работы его аккаунта. Чтобы перейти на данную страницу просто нужно нажать в меню слева на раздел «Settings». На этой странице реализовано три раздела: general, notifications, integrations.

В первом разделе предусмотрено включение и выключение отображения баланса, активация тёмной темы интерфейсы, смена локализации приложения, а также возможность полностью стереть аккаунт пользователя (рисунок 3.13).

Во втором разделе есть такие настройки как включение уведомлений, возможность получать копию уведомлений на почту, включение и выключение звука в браузере при оповещении (рисунок 3.14).

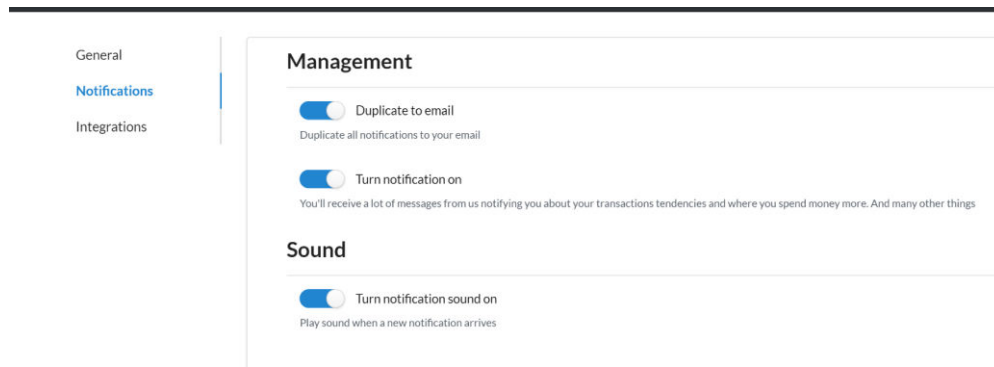


Рисунок 3.14 – Второй раздел настроек

Третий раздел включает в себя лишь одно поле, которое хранит в себе специальный токен, с помощью которого можно дать доступ к некоторым (не всем) данным пользователя (рисунок 3.15). Предполагается, что если в будущем будут возможны интеграции с моим веб-приложением, то с помощью этого токена можно будет иметь возможность создавать и обновлять категории и транзакции (нет возможности удаления каких-либо данных).

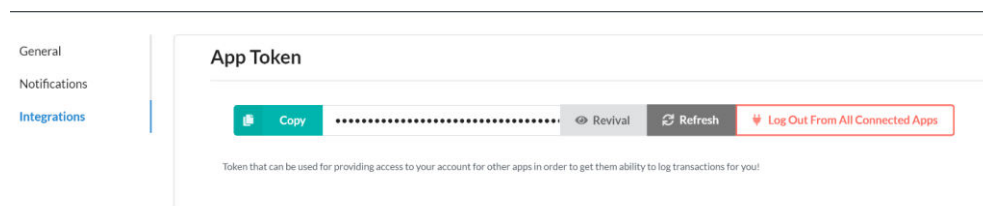


Рисунок 3.15 – Третий раздел настроек

Если это необходимо — есть возможность разлогиниться из всех интегрированных приложений.

Если пользователь захотел поменять свою личную информацию (имя, основной баланс), он может это сделать на любой странице приложения, кликнув на иконку пользователя в заголовке страницы (рисунок 3.16).

The screenshot shows a mobile application interface. At the top, there is a dark header bar with a balance of '1200', an envelope icon, a user profile icon, and a 'Log Out' button. Below the header, a modal form is displayed. The form has a title 'Ilya Matsuev'. It contains four input fields: 'First Name' with the value 'Ilya', 'Last Name' with the value 'Matsuev', 'Email' with the value 'ilya.matsuev.personal@gmail.com', and 'Balance' with the value '1200' and a dollar sign. A blue 'Save' button is located at the bottom of the form.

Рисунок 3.16 – Изменение личных данных

Редактирование электронной почты пользователя, собственно, его логина, на данный момент не предусмотрено.

В приложении присутствует возможность обратной связи, например, если вы хотите оставить отзыв, сделать пожелание или поделиться багом. Сделать это можно на стартовой странице приложения, нажав на кнопку в заголовке страницы «Ask Question / Bug Report». После чего откроется модальное окно, в которое пользователь должен ввести всю необходимую информацию, чтобы сотрудники кол-центра могли помочь пользователю. Модальное окно представлено на рисунке 3.17.

Cash Scheduler Project Repository Ask Question / Bug Report

Ask Question / Bug Report

Name: John Wick Email: ilya.matsuev.personal@gmail.com Phone: +375292033067

Subject: Some Topic...

Description: Hi there! I've got a question about...

☐ I'm not a robot

Close Send

Рисунок 3.17 – Форма обратной связи

Case 00001026 00001027

Test question

Priority: Medium Status: New Case Number: 00001027

Feed Related

Close Case Email

Create new... Add

Most Recent Activity Search this feed...

Details

Case Number: 00001027 Status: New

Case Owner: Cash Scheduler Priority: Medium

Contact Name: Ilya Matsuev Case Origin: Web

Type: Other Case Reason:

Web Name: Ilya Matsuev Date/Time Opened: 4/18/2021, 3:01 PM

Web Email: ilya.matsuev.perso... Date/Time Closed: 4/18/2021, 3:04 PM

Web Phone:

Internal Comments:

Created By: System, 4/18/2021, 3:01 PM Last Modified By: Cash Scheduler, 4/18/2021, 3:04 PM

Рисунок 3.18 – Принятие обратной связи со стороны агента

После отправки запроса пользователя, он может быть принят в работу одним из агентов поддержки в кол-центре, как показано на рисунке 3.18. Здесь агент может проставить всю необходимую информацию, которую он считает нужной, оставить комментарии и ответить пользователю по электронной почте или по телефону, если он есть.

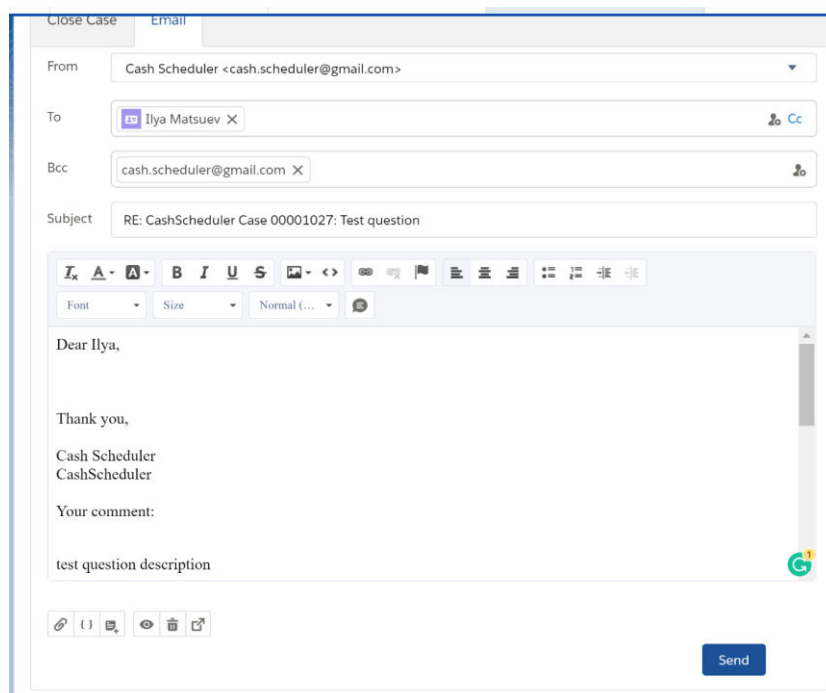


Рисунок 3.19 – Шаблон ответа пользователю по почте

После того, как агент ответит пользователю, ему автоматически в реальном времени придет оповещение в приложении, помимо того, что оно было отправлено на почту (рисунок 3.20).

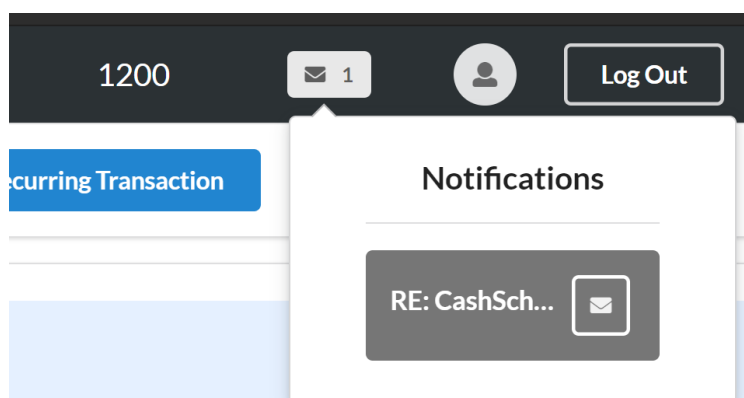


Рисунок 3.20 – Список уведомлений пользователя

Уведомление можно пометить как прочитанное, а можно открыть, как показано на рисунке 3.21.

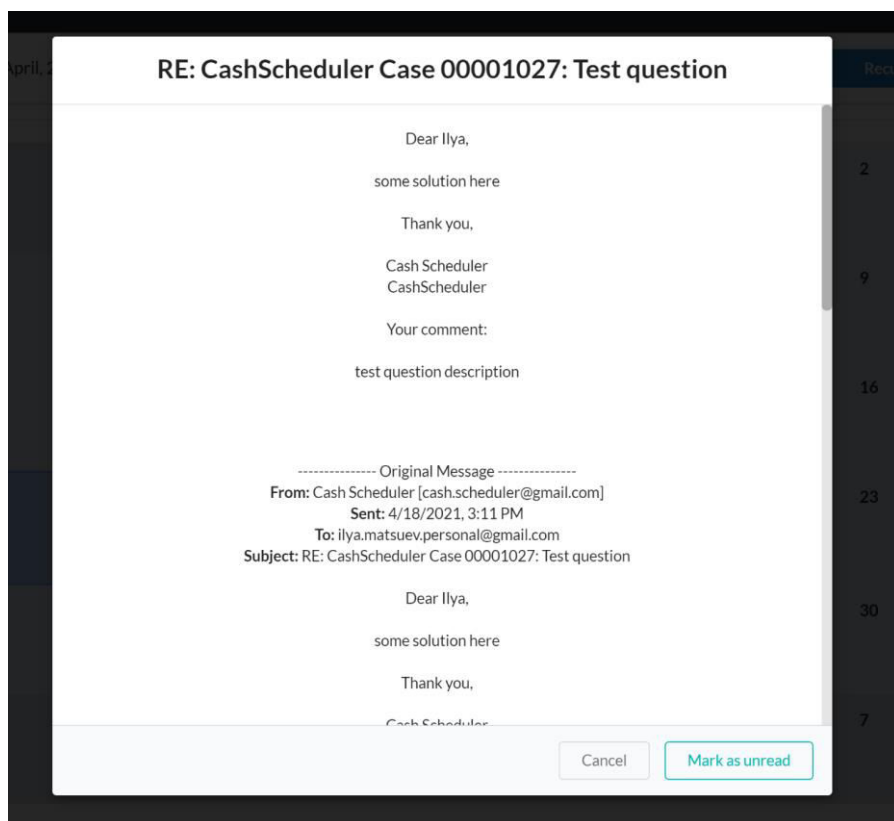


Рисунок 3.20 – Уведомление от агента

Таким образом пользователь в реальном времени может получать уведомления без надобности проверять электронную почту.

Заключение

За время прохождения преддипломной практики было выполнено ознакомление со структурой и с комплексом технических и программных средств Белорусского государственного технологического университета, выполнен аналитический обзор аналогов по тематике дипломного проектирования, выполнено проектирование, разработан дизайн, выполнена программная реализация ресурса, а именно: разработано веб-приложение для управления личными финансами.

Выполненная работа основана на официальной документации используемых технологий, выбранные методы реализации практичны и современны. Основные цели и задачи были выполнены.

В ходе прохождения практики регулярно поступала информация от руководителя, контролирующего этапы разработки.

Список использованных источников

1. Документация ASP .NET Core [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0> — Дата доступа: 28. 03. 2021
2. Документация Hot Chocolate для ASP .NET Core [Электронный ресурс]. Режим доступа: <https://chillicream.com/docs/hotchocolate/v10> — Дата доступа: 29. 03. 2021.
3. GraphQL клиент для React приложения [Электронный ресурс]. Режим доступа: <https://www.apollographql.com/docs/react/> — Дата доступа: 02. 04. 2021
4. Настройка Service Cloud на Salesforce [Электронный ресурс]. Режим доступа: <https://trailhead.salesforce.com/en/content/learn/modules/omni-channel-readiness-and-digital-engagement/defining-omni-channel> — Дата доступа: 11. 04. 2021