МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-98 01 03 “Программное обеспечение информационной

безопасности мобильных систем”

Специализация Инженер-программист

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«FinaMoon – персональный финансовый менеджер»

Выполнил студент Пилик Семён Игоревич

(Ф.И.О.)

Руководитель проекта преподаватель-стажер Колмаков М.В. (учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В .

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант: преподаватель-стажер Колмаков М.В. (учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: преподаватель-стажер Колмаков М.В. (учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2020

**Реферат**

Пояснительная записка курсового проекта содержит 25 страниц, 7 источников литературы и одного приложения.

Основными целями курсового проекта являются: разработка программного средства – «FinaMoon – персональный финансовый менеджер».

Пояснительная записка состоит из введения, пяти разделов, заключения.

Во введении представлена общая информация, дающая представление о предстоящей работе, определены цели.

В первом разделе представлена постановка задачи.

Во втором разделе рассматривается разработка архитектуры проекта.

В третьем разделе рассматривается разработка функциональной модели приложения.

В четвертом разделе представлено руководство пользователя.

В заключении представлены итоги курсового проектирования и задачи, которые были решены в ходе разработки программного средства.

Содержание

[Введение 4](#_Toc59534564)

[1. Постановка задачи. 5](#_Toc59534565)

[2. Архитектура проекта 6](#_Toc59534566)

[3. Схема базы данных 7](#_Toc59534567)

[4. Программная реализация приложения 8](#_Toc59534568)

[4.1 Реализация мобильного приложения 8](#_Toc59534569)

[Проект в себе содержит 3 классa страниц: 9](#_Toc59534570)

[5. Руководство пользователя 10](#_Toc59534571)

[5.1 Android приложение 10](#_Toc59534572)

[Заключение 19](#_Toc59534573)

[Список использованных источников 20](#_Toc59534574)

# Введение

Мобильное приложение — это программное обеспечение, специально разработанное под конкретную мобильную платформу (iOS, Android, Windows Phone и т. д.). Предназначено для использования на смартфонах, фаблетах, планшетах, умных часах и других мобильных устройствах.

Основные плюсы мобильных приложений: интерфейс программы создан конкретно под работу на мобильном устройстве через сенсорный экран или кнопки,

удобная и понятная для пользователей гаджетов навигация, мобильное меню, лучшее взаимодействие с пользователем через сообщения, пуш-уведомления, напоминания. Приложение может выполнять функции даже в фоновом режиме, чего нельзя сказать о сайте. Для работы с программой не нужно открывать браузер, а многие приложения поддерживают ряд функций и при отключенном интернете.

Темой курсового проекта является разработка программного средства «FinaMoon – финансовый менеджер» - мобильного приложения, основной задачей которого является отслеживание доходов и расходов. Отслеживание долгов.

Целью курсового проекта является проектирование и реализация программного средства.

# Постановка задачи.

**Visual Studio Code** — [редактор исходного кода](https://ru.wikipedia.org/wiki/Редактор_исходного_кода), разработанный Microsoft для [Windows](https://ru.wikipedia.org/wiki/Windows), Linux и [macOS](https://ru.wikipedia.org/wiki/MacOS). Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки [веб](https://ru.wikipedia.org/wiki/Веб-приложение)- и облачных приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/Отладчик), инструменты для работы с Git ,[подсветку синтаксиса](https://ru.wikipedia.org/wiki/Подсветка_синтаксиса), IntelliSense и средства для [рефакторинга](https://ru.wikipedia.org/wiki/Рефакторинг). Имеет широкие возможности для [кастомизации](https://ru.wikipedia.org/wiki/Кастомизация): пользовательские темы, сочетания клавиш и [файлы конфигурации](https://ru.wikipedia.org/wiki/Файл_конфигурации). Распространяется бесплатно, разрабатывается как программное обеспечение с [открытым исходным кодом](https://ru.wikipedia.org/wiki/Открытое_программное_обеспечение), но готовые сборки распространяются под [проприетарной лицензией](https://ru.wikipedia.org/wiki/Проприетарное_программное_обеспечение).

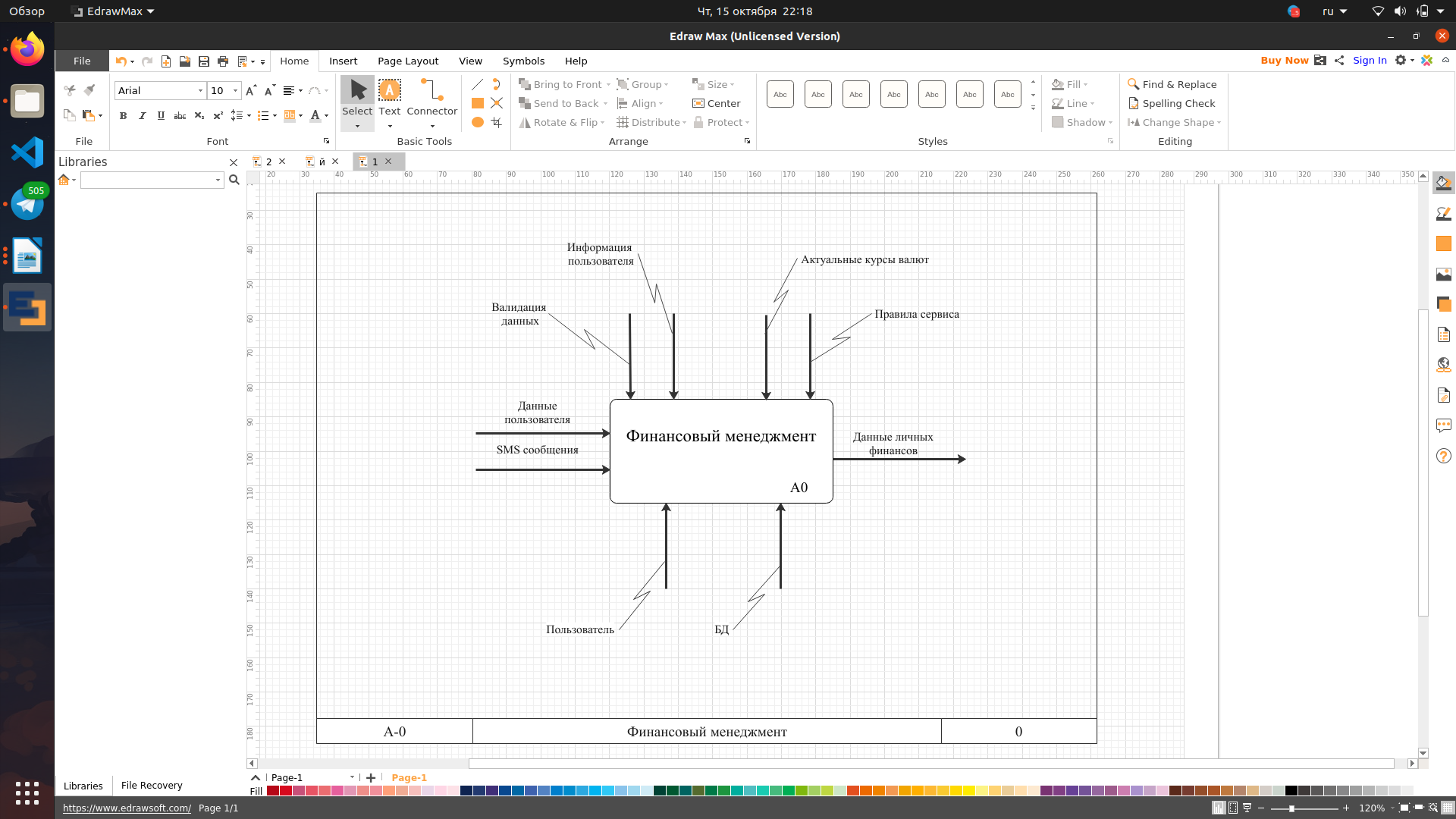
**Flutter** — SDK с открытым исходным кодом для создания мобильных приложений от компании [Google](https://ru.wikipedia.org/wiki/Google_(компания)). Он используется для разработки приложений под Android и [iOS](https://ru.wikipedia.org/wiki/IOS), а также это пока единственный способ разработки приложений под [Google Fuchsia](https://ru.wikipedia.org/wiki/Google_Fuchsia).  
 Приложения Flutter пишутся на языке **Dart**. В Android, а также под [Windows](https://ru.wikipedia.org/wiki/Windows), macOS и Linux с помощью Flutter Desktop Embedding, Flutter работает в виртуальной машине Dart с J[IT-компилятором](https://ru.wikipedia.org/wiki/JIT-компилятор). Из-за ограничений на динамическое выполнение кода в [App Store](https://ru.wikipedia.org/wiki/App_Store), под iOS Flutter использует [AOT-компиляцию](https://ru.wikipedia.org/wiki/AOT-компиляция).Одно из главных преимуществ платформы Dart — «горячая перезагрузка», когда изменение исходного кода применяется сразу в работающем приложении без необходимости его перезапуска.

Клиентом будет выступать мобильное приложение на устройстве с операционной системой Android 4.0.3 и выше.

# Архитектура проекта

## Контекстная диаграмма

Контекст предполагает управление финансами и обработку SMS солбщений, поэтому на вход поступают пользовательские данные и SMS сообщения.

В качестве механизмов используются валидация данных, актуальные курсы валют, инвормация пользователя, правила сервиса.

На выходе мы получаем структурированные данные о личных финансах, уведомления и конвертированную валюту.

Описанный ранее контекст выглядит так:

Рисунок 2.1 – контекстная диаграмма.

## Диаграмма 1-го уровня декомпозиции

Рисунок 2.2 – диаграмма 1-го уровня декомпозиции.

Функциональные блоки:

* Обработка данных – функция, подразумевающая определение типа данных и их классификацию;
* Анализ SMS сообщений – функция, получающая данные о транзакции из SMS;
* Расчёт данных — фунция математических операций над кошельками;
* Фиксирование данныз – функция удаления, создания , изменения данных .

Распределенное приложение состоит из 2-х частей:

Мобильное приложение. Эта часть состоит из Dart-классов, с помощью которых реализована основная логика мобильного приложения и осуществляется обращение к серверу.

Сервер. Данная часть предоставляет Node.js приложение, которое обрабатывает запросы с клиента с помощью контроллеров и взаимодействует с базой данных посредством библиотеки pg-promise.

Графическое представление архитектуры распределенного приложения изображено на рисунке 2.3.

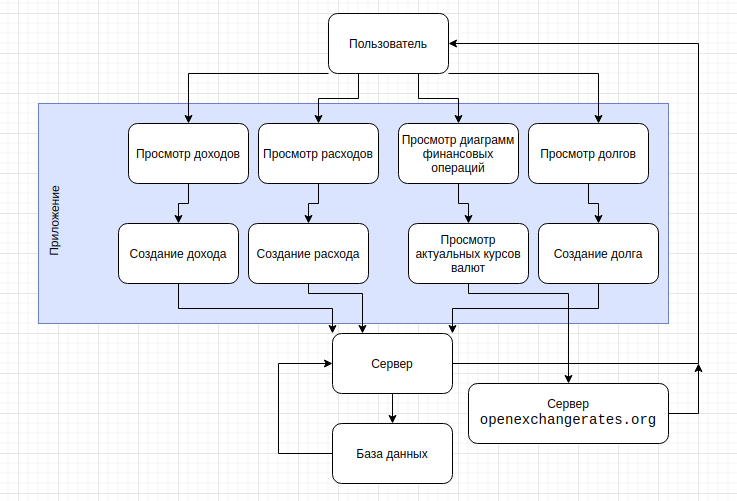


Рисунок. 2.3 – архитектура распределенного приложения

Далее происходит декомпозиция блока «Анализ SMS» по меодологии IDEF3 (рис. 2.4):

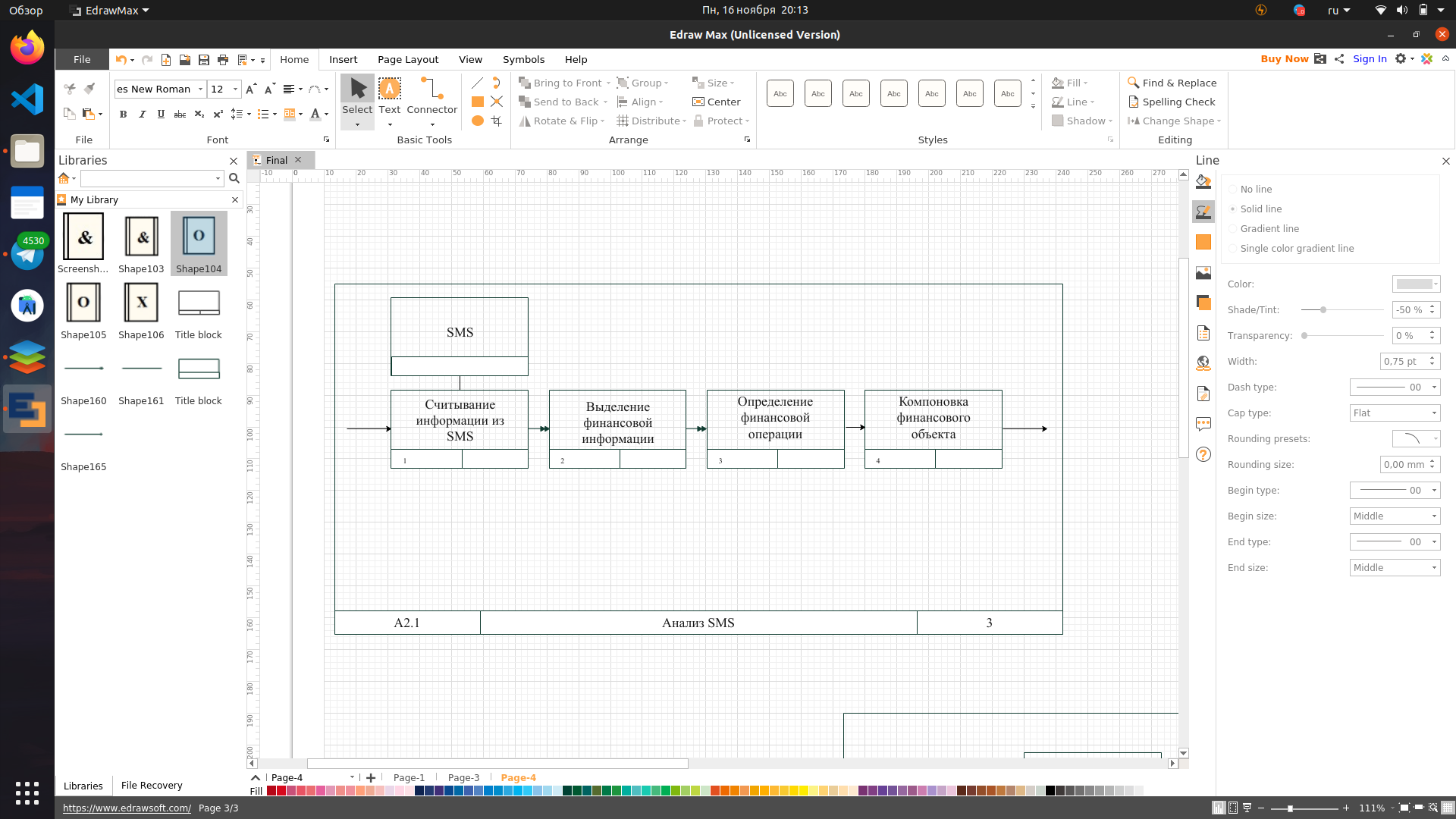


Рис 2.4 – Декомпозиция процесса «Анализ SMS»

Далее происходит декомпозиция блоков «Расчёт данных» по меодологии IDEF3 (рис 2.5):

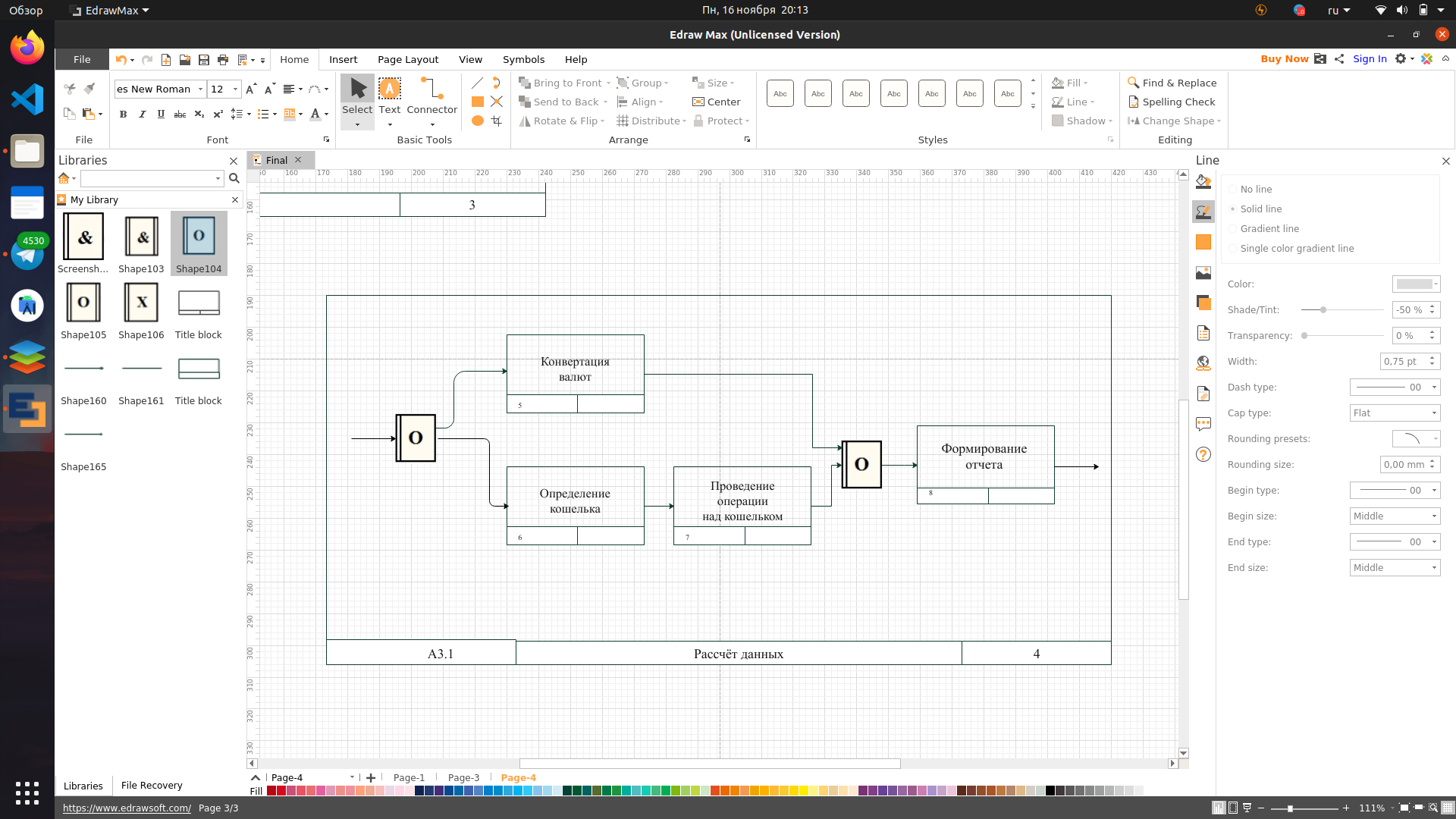


Рис 2.5 – Декомпозиция процесса «Расчет данных»

Функциональные блоки:

* Конвертация валют – функция, подразумевающая в себе преобразование валют согласно требованиям;
* Определение кошелька – функция определения и получения ссылки на требуемый кошелек
* Проведение операции над кошельком – проведение операции (добавление расхода, добавление дохода, итоговая сумма) над кошельком
* Компоновка данных – функция компоновки проведенной операции;
* Расшифровка – функция получение нужных полей SMS$
* Структурирование – функция создания финансового объекта из расшифрованной информации SMS;
* Определение финансовой операции - функция определения финансовой операции;

Потоки данных:

* Данные – текст и медиа-файлы, используемые при создании статьи
* Правила написания статей – основные правила создания статей, позволяющие им быть информативными и наглядными
* Правила модерации
* Оферта
* Пользователь

# Схема базы данных

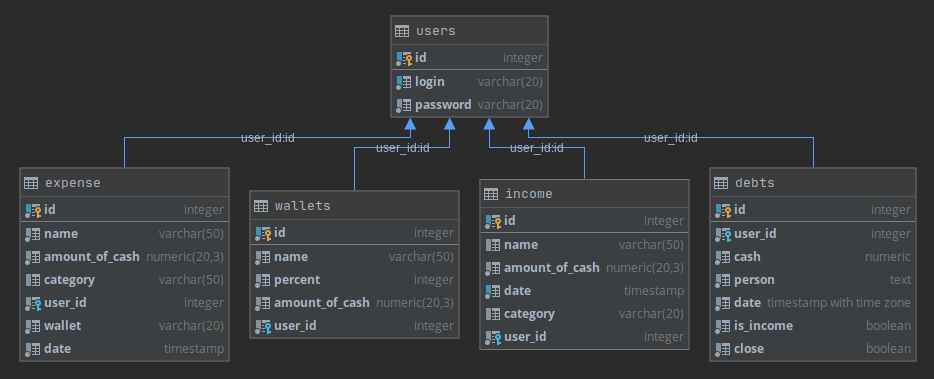
В ходе написания приложения была разработана локальная база данных «FinaMoon»,схема которой представлена на рисунке 3.1.

Рисунок 3.1 – структура базы данных

Таблица Users предназначена для хранения пользователей, зарегистрированных в приложении. Wallets – для хранении информации о кошельках Incomes и Expenses — для хранения информации о доходах и расходах соответственно. Debts — отслеживание долгов.

# Программная реализация приложения

## 4.1 Реализация мобильного приложения

Мобильное приложение реализовано для операционных систем Android и IOS. Данное приложение было разработано в среде разработки –Visual Studio Code.

Графическая структура приложения отображена на рисунке 4.2.1

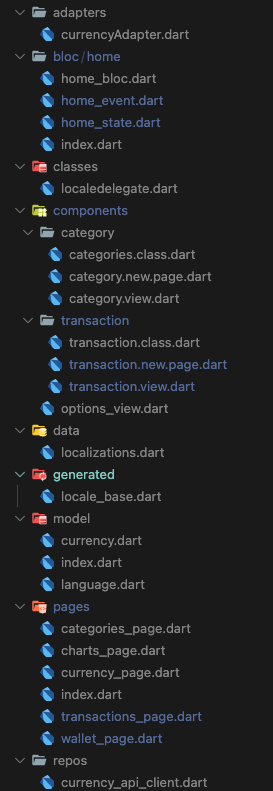


Рисунок 4.2.1 – структура проекта

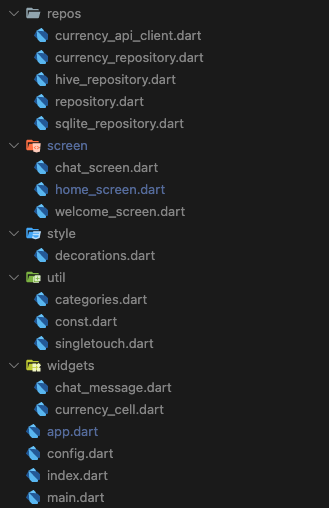


Рисунок 4.2.2 – структура проекта

### Проект в себе содержит 3 классa страниц:

1. chat\_screen – содержит страницу переписки с ботом.
2. Home\_screen – содержит страницу финансов, графиков, курсов валют.
3. Lock\_screen – содержит страницу входа в приложение

И 7 экранов на странице Home:

1. Wallet\_page – страница с основной информацией о кошельке
2. Transactions\_page – страница с информацией о проведенный транзакциях
3. Transaction\_edit\_page – страница изменения транзакции
4. Transaction\_new\_page – страница создания транзакции
5. Transactions\_operations – страница выбора операции над транзакцией
6. Charts\_page – страница графиков
7. Currency\_Page – страница конвертации валют

Остальные классы и их содержимое отвечают за конфигурацию, взаимодействие с локальной базой данных, описание основных сущностей, управление состоянием.

# Руководство пользователя

## 5.1 Android приложение

Для начала необходимо установить приложение на мобильный устройство. Запустим наше приложение. Сразу после запуска мы увидим окно ввода пароля (Рисунок 5.1.1).

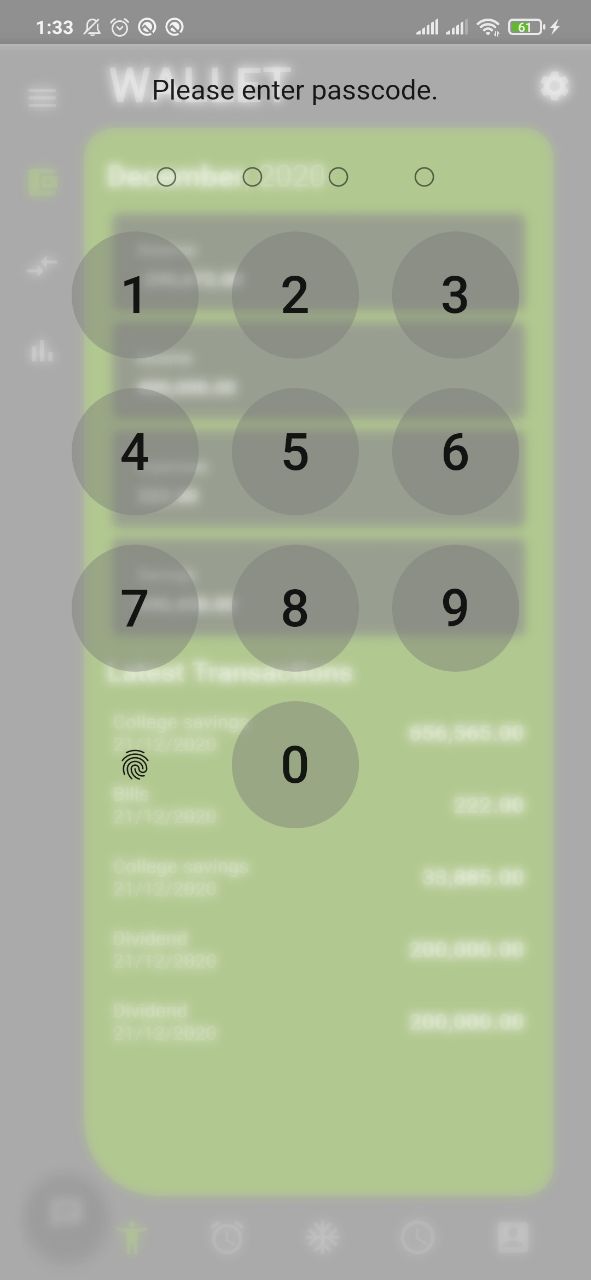


Рисунок 5.1.1 – окно ввода пароля

Также присутствует авторизация с помощью отпечатка пальца. Если успешно пройти авторизацию на первой странице, то произойдет переход на главную страницу где можно просмотреть списки расходов, доходов и сбережени (Рисунок 5.1.2).

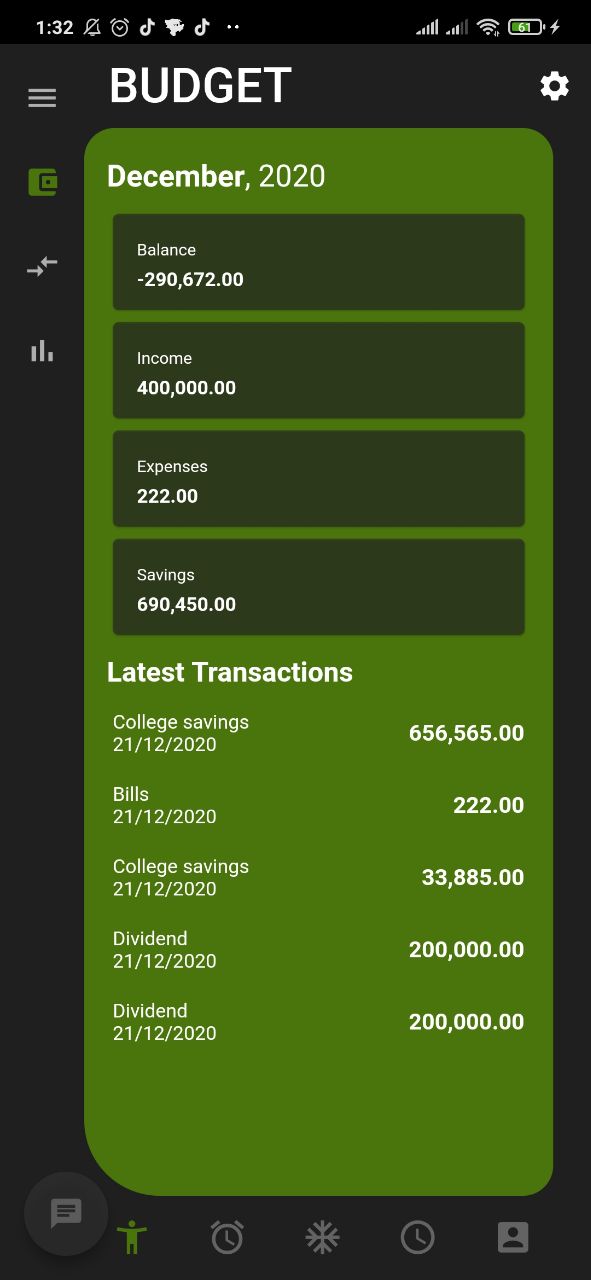


Рисунок 5.1.2 – главная страница

При смахивании вниз происходит переход на страницу создания транзакции, где пользователь может ввести такие данные как: название дохода или расхода, сумма, выбрать категорию. Также запланировать транзакцию, и определить срок е> действия (Рисунок 5.1.3).

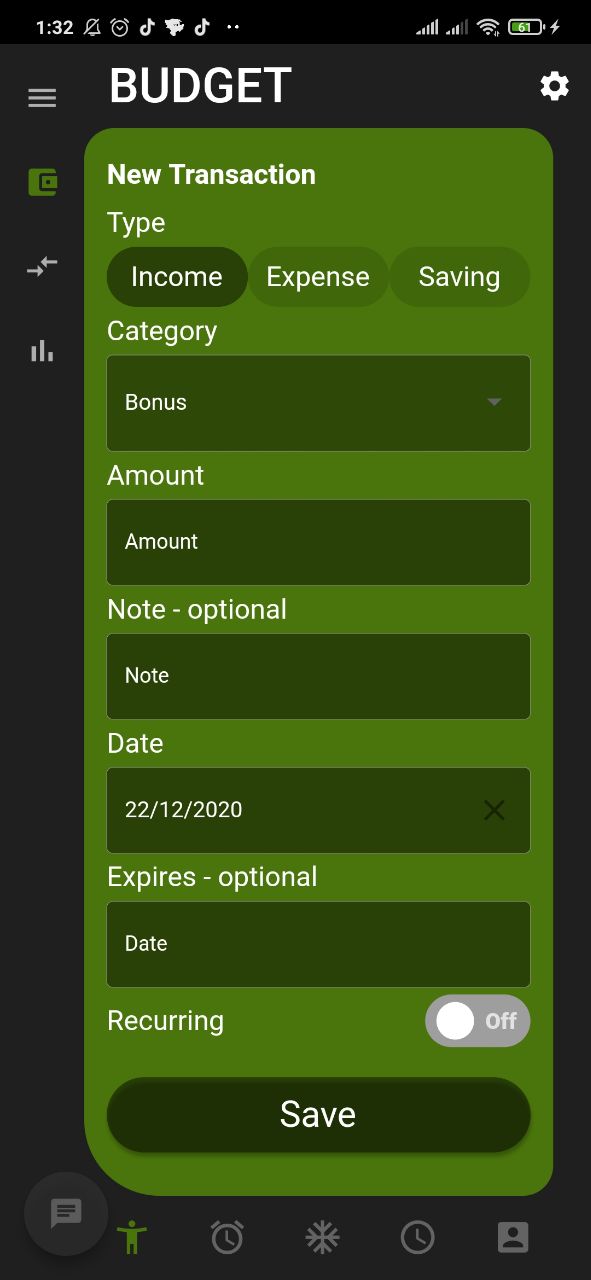


Рисунок 5.1.3 – создание транзакции

После ввода данных и нажатии кнопки «создать» произойдет создание события и переход на главный экран приложения. При смахивании вправо происходит переход на страницу создания просмотра проведенных транзакций по месяцам (Рисунок 5.1.4), где пользователь может выбрать нужную ему транзакцию из списка и открыть страницу выбора операции над транзакцией (Рисунок 5.1.5).

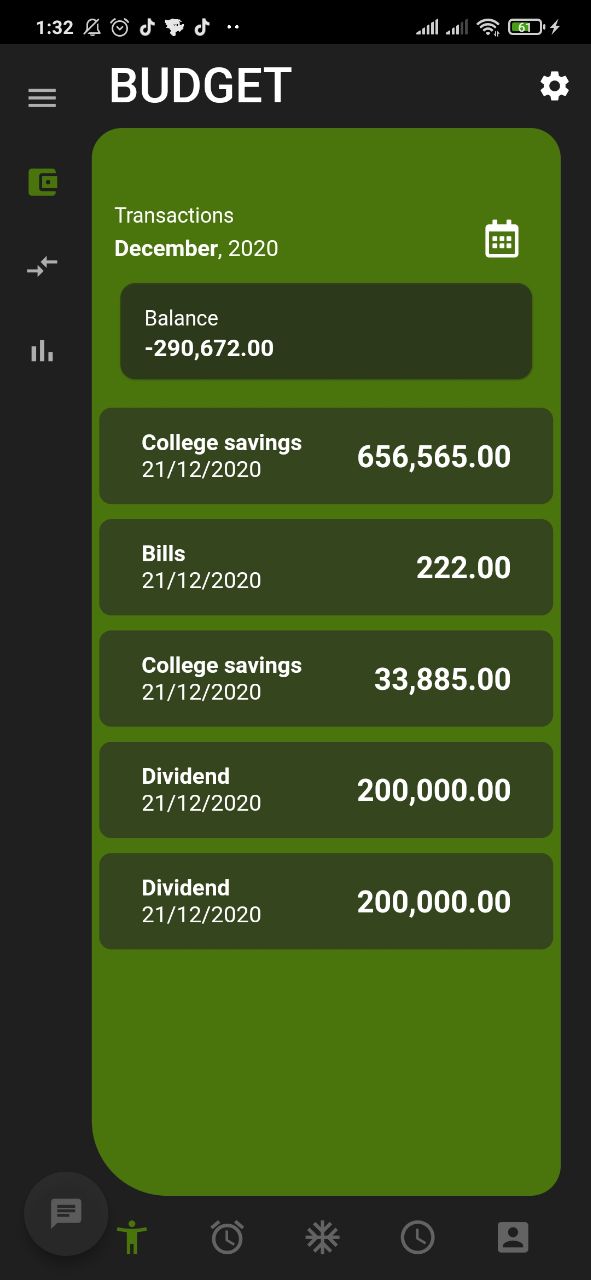


Рисунок 5.1.4 – страница создания просмотра проведенных транзакций

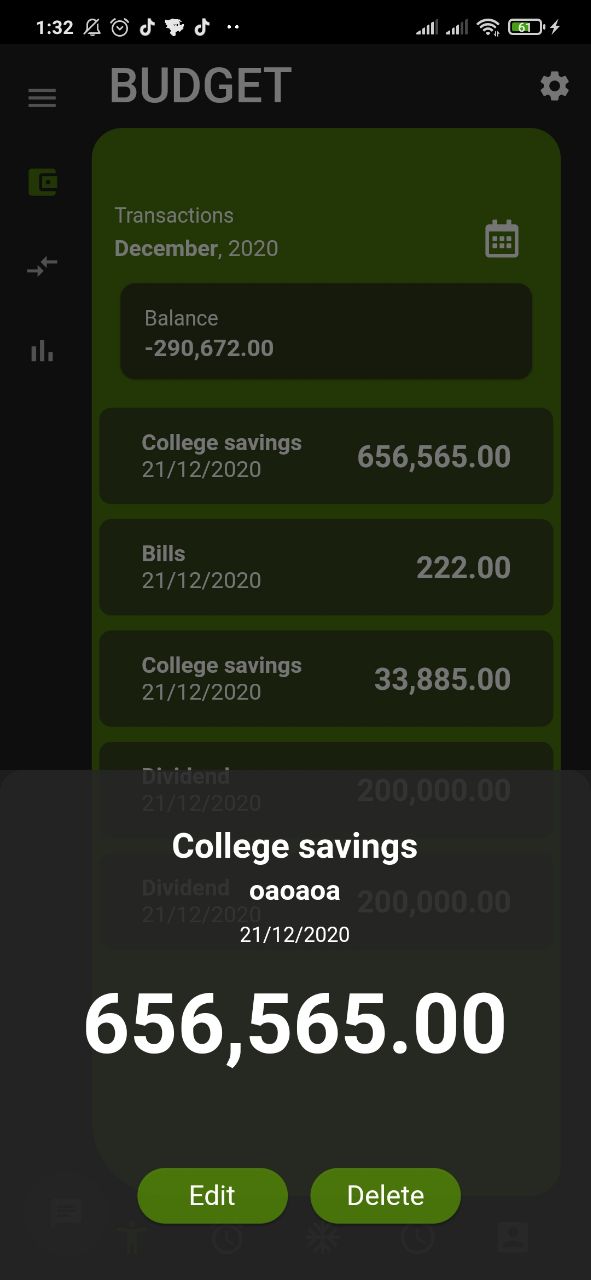


Рисунок 5.1.5 – страница выбора операции над транзакцией

Если выбрать пункт удаления транзакции, то откроется диалоговое окно для подтверждения (Рисунок 5.1.6), если же пункт изменения, то откроется окно редактирования транзакции (Рисунок 5.1.7)

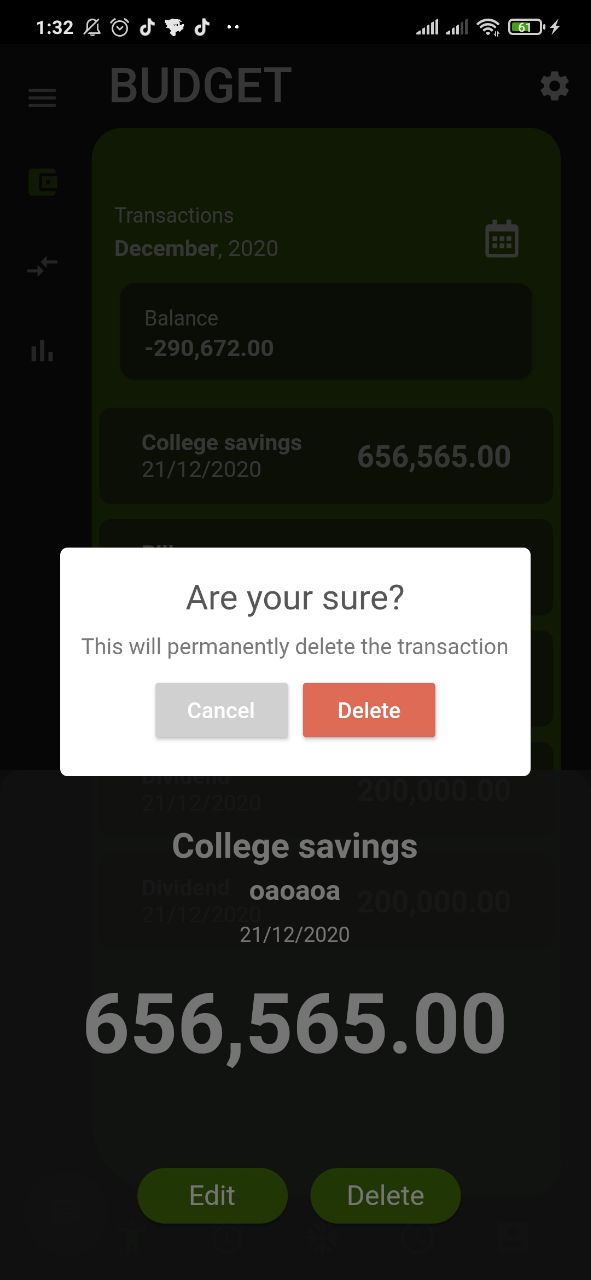


Рисунок 5.1.6 – диалоговое окно для подтверждения удаления

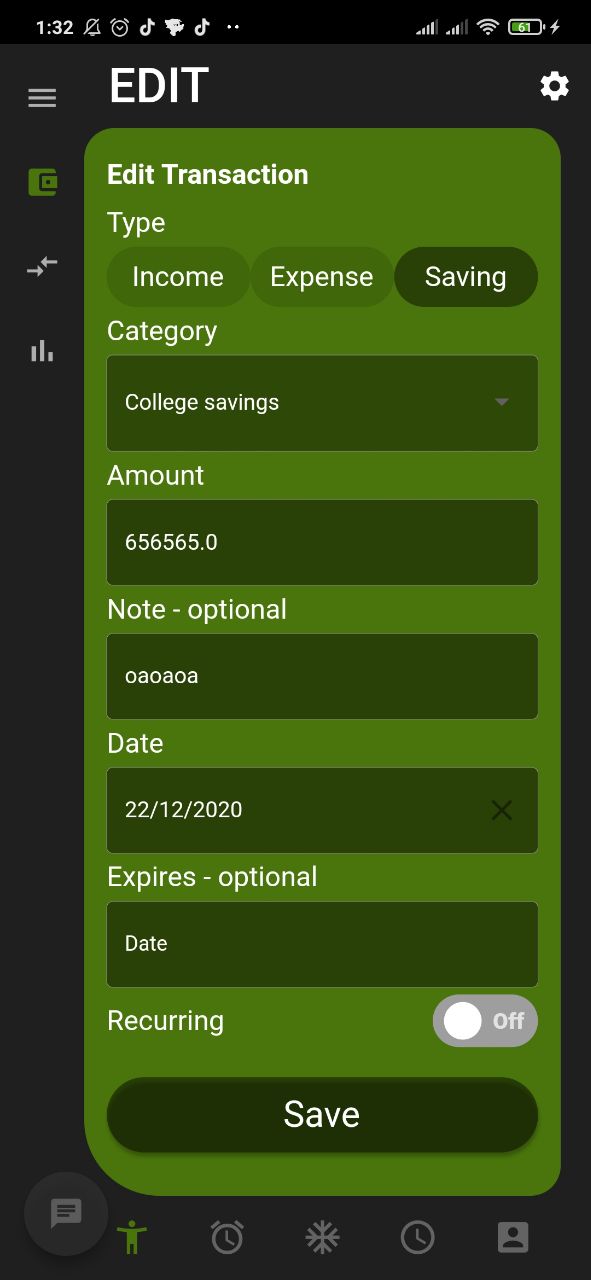


Рисунок 5.1.7 – окно редактирования транзакции

Если заполнить приложение данными то в окне графиков появляются графики транзакций (Рисунок 5.1.8).

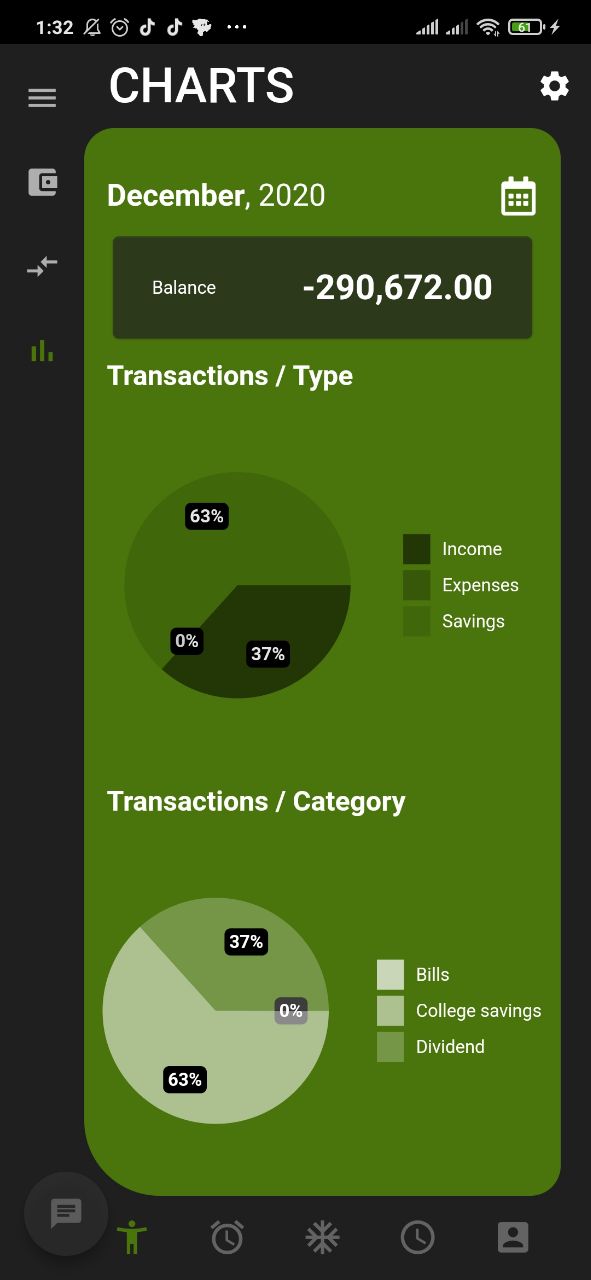


Рисунок 5.1.8 – окно кошельков

Если в меню выбрать пункт «Валюта» то откроется окно конвертации валют, приложение получает актуальные курсы из открытой АПИ и сохраняет в локальную базу данных из-за чего даже при неработающем интернете можно будет получит приблизительные значение

я (Рисунок 5.1.9)



Рисунок 5.1.9 – окно курсов валют

Если обобщить все вышесказанное и показанное, то на данных скриншотах приведена работа приложение и с помощью текста изложены все нюансы при работе с ним.

При нажатии на кнопку в левой нижней части экрана открывается переписка с ботом на Dialogflow, на данный момент он не сильно обучен но в перспективе все описанные ранее операции можно будет проводить через него(Рисунок 5.1.10)

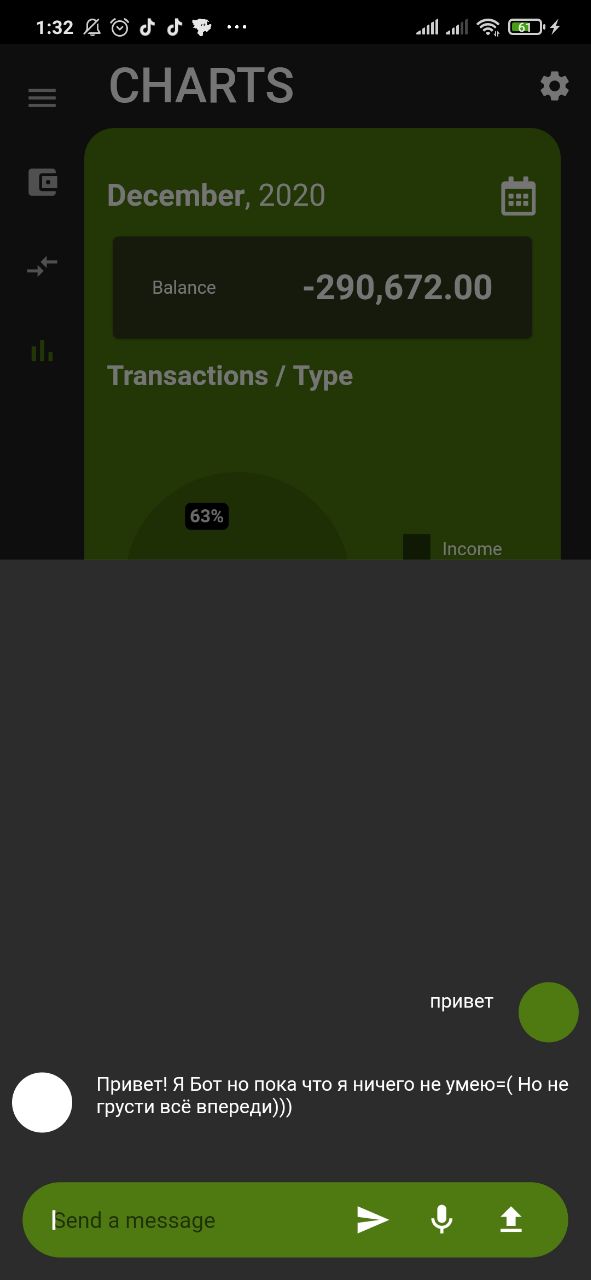


Рисунок 5.1.10 – не выплаченный долг

Так же хотелось бы сказать, что в приложении присутствует синхронизация локальной и удалённой баз данных. Локальная база данных повторяет удаленную и создана для локального хранения данных (без сети). Все данные при авторизации выгружаются с удалённого сервера на локальную базу данных и последующее взаимодействие происходит с локальной бд. В момент следующей авторизации базы данных обмениваются данными.

# Заключение

Решая поставленную задачу, я пришел к удовлетворительному результату. Таким образом, была достигнута цель, и был создан проект ‘FinaMoon’, который включает в себя android / ios приложение.

Программная система имеет простой, интуитивно понятный пользователю интерфейс.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанный продукт работает корректно, а требования технического задания выполнены в полном объеме.

В результате приложение обладает следующими особенностями:

* реализована отправка / получение данных на удалённый сервер по технологии REST API;
* реализован чат-бот;

# Список использованных источников

1. Stackoverflow - https://stackoverflow.com;

2. Cyberforum - https://cyberforum.ru;

3. Github - https://github.com;

4. Android documentation - <https://developer.android.com/docs>.

5. Java documentation - <https://docs.oracle.com/en/java/>

6. ASP.NET documentation - <https://docs.microsoft.com/en-us/aspnet/core/>

7. SQLServer documentation - <https://docs.microsoft.com/en-us/sql/sql-server/>