МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной

безопасности мобильных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

**«Приложение для управления бюджетом»**

Выполнил студент Бруйло Александр Эдуардович

(Ф.И.О.)

Руководитель проекта асс. Уласевич Н.И. (учен. степень, звание, должность, Ф.И.О., подпись)

И.о. зав. Кафедрой к.т.н., ст. преп. Блинова Е.А.

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

Консультант: асс. Уласевич Н.И. (учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: асс. Уласевич Н.И. (учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Содержание

[Введение 5](#_Toc167082182)

[1 Анализ аналогичных решений 6](#_Toc167082183)

[1.1 Daak 6](#_Toc167082184)

[1.2 BudgetOK 6](#_Toc167082185)

[1.3 Умный счёт 7](#_Toc167082186)

[2 Проектирование приложения 9](#_Toc167082187)

[2.1 Проектирование базы данных 9](#_Toc167082188)

[2.2 Описание информационных объектов и ограничений целостности 9](#_Toc167082189)

[2.2.1. Таблица users 10](#_Toc167082190)

[2.3 Проектирование API 10](#_Toc167082191)

[2.4 Проектирование мобильного приложения 10](#_Toc167082192)

[2.5 Выводы по главе «Проектирование программного продукта» 11](#_Toc167082193)

# Введение

В последнее время наблюдается рост популярности мобильных приложений, позволяющих пользователю управлять своими финансами прямо с телефона, имея доступ к интернету. Это стало возможным благодаря быстрому развитию науки и технологий, а также стремлению людей экономить время и контролировать свои расходы без необходимости покидать дом. Разработка мобильного приложения для управления бюджетом дает ряд преимуществ: удобство доступа к информации о расходах, возможность планировать финансовые цели, уменьшение рисков, связанных с перерасходом средств. Приложение для менеджмента бюджета позволяет пользователям всегда быть в курсе своих финансовых дел, повышая их удовлетворенность и мотивацию к экономии.

Задачи курсового проекта:

* Изучение существующих мобильных приложений для управления бюджетом, анализ их достоинств и недостатков для формирования требований к новому приложению;
* Проектирование приложения, позволяющее визуализировать предварительную концепцию и спланировать разработку программного продукта, а также распределить необходимые ресурсы и время на разработку;
* Реализация программного продукта, который будет представлять собой мобильное приложение для управления бюджетом;
* Разработка руководства пользователя, чтобы помочь пользователям ознакомиться с основными функциями и принципами работы приложения. Это важный элемент разработки, поскольку некоторые пользователи предпочитают заранее изучить функционал перед его использованием.

По завершении изучения и формулирования задач курсового проекта, окончательная цель проекта заключается в предоставлении пользователям удобного и функционального инструмента для управления своими финансами с помощью мобильного приложения.

# Анализ аналогичных решений

В современном мире приложения для управления бюджетом становятся все более популярными сервисами. Их основная задача — это удобное отслеживание финансов и планирование расходов. При разработке своего приложения я обратил внимание на самые успешные примеры популярных приложений для менеджмента бюджета. Аналоги рассмотрены ниже.

## Daak

Одним из значимых аналогов в сфере мобильных приложений для управления бюджетом является «Daak». Это приложение предоставляет пользователю детальный учет финансов, включая данные о расходах, доходах и долговых обязательствах. Особенности приложения включают автоматическое присвоение категорий транзакциям, настройки для кастомизации финансовых отчетов, интуитивно понятный интерфейс для анализа бюджета, а также уведомления о превышении установленных финансовых лимитов.

Интерфейс приложения показан на рисунке 1.1.

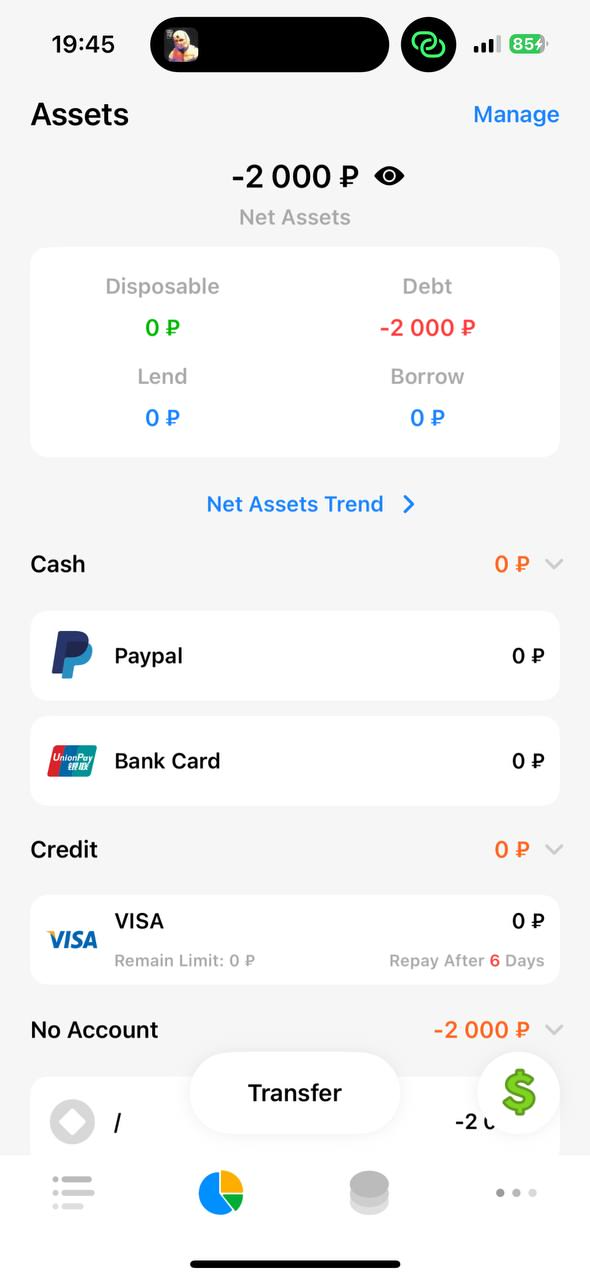
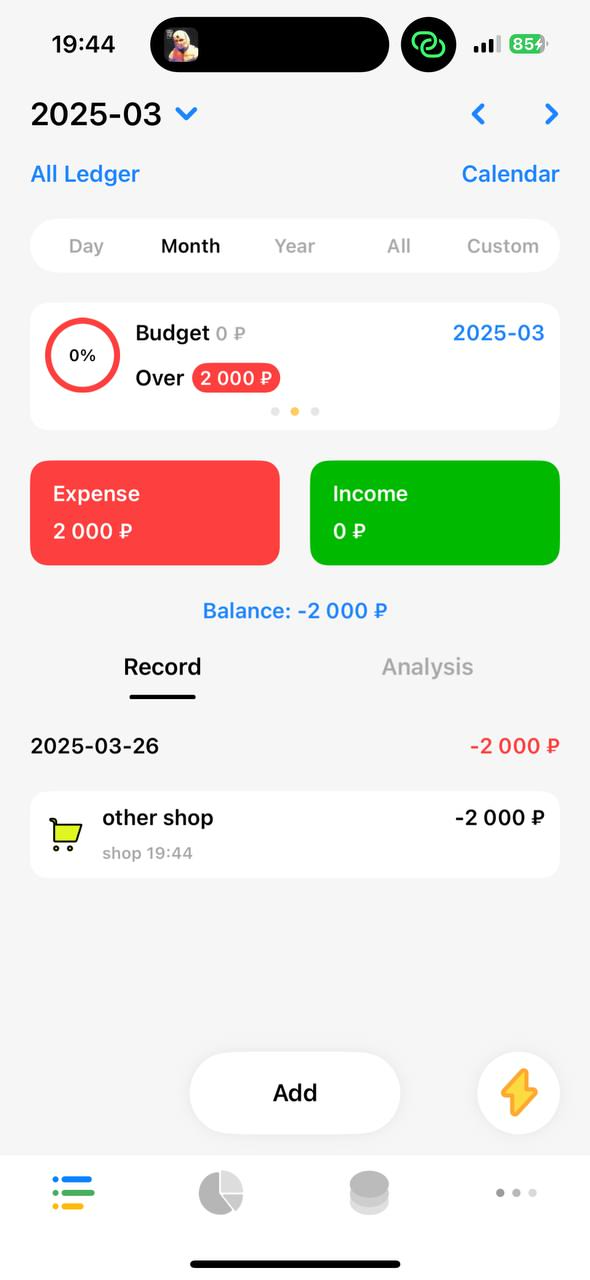


Рисунок 1.1 – Интерфейс приложения «Daak»

Однако среди недостатков приложения «Daak» можно отметить, что оно не всегда корректно обрабатывает данные о транзакциях, особенно при синхронизации с банковскими счетами, что может привести к ошибкам в учете бюджета.

## BudgetOK

Мобильное приложение «BudgetOK» предлагает широкий спектр возможностей для управления личными финансами. Оно позволяет пользователям удобно отслеживать свои расходы. Особенности приложения включают генерацию отчетов о расходах, ведение учета долгов и возможность добавления и управления доходами. Также данное приложение позволяет удобно просматривать историю трат. BudgetOK помогает пользователям эффективно планировать свой бюджет и контролировать финансовую ситуацию, предлагая интуитивно понятный интерфейс и полезные инструменты для анализа расходов.

Интерфейс приложения показан на рисунке 1.2.

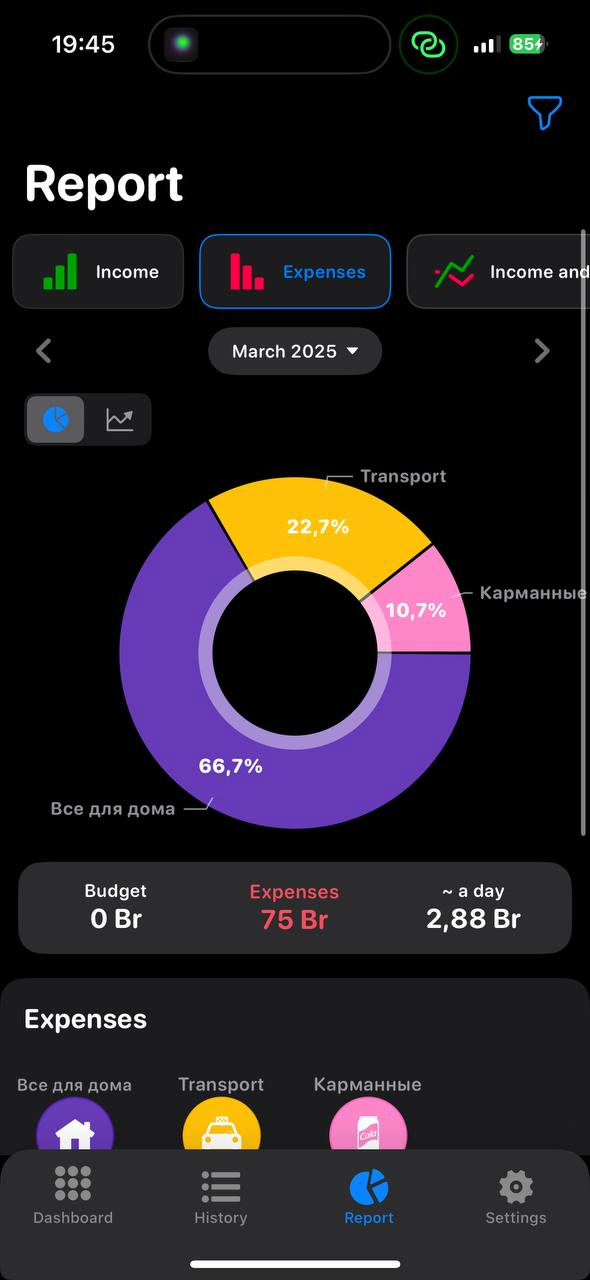
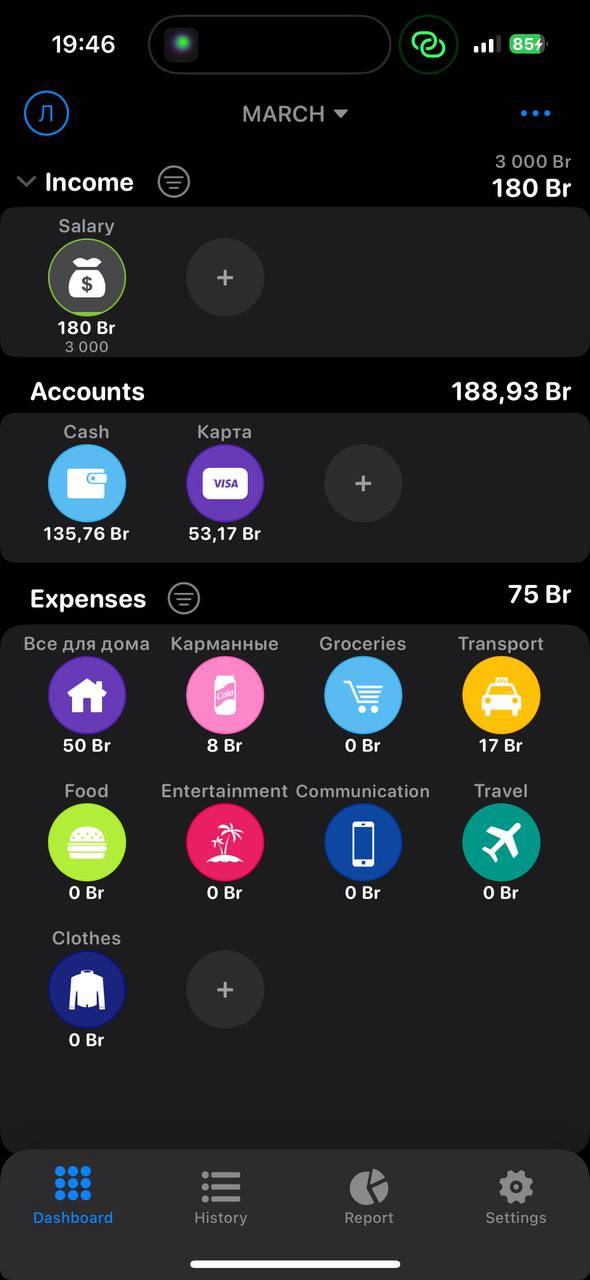


Рисунок 1.2 – Интерфейс приложения «BudgetOK»

Однако среди недостатков приложения «BudgetOK» пользователи указывают на недостаток персонализации в настройках, что ограничивает возможности адаптации приложения под индивидуальные финансовые нужды.

## Умный счёт

Мобильное приложение «Умный счёт» предоставляет пользователям подробные и актуальные данные о своих финансах, включая отчеты о расходах, доходах и долговых обязательствах. Оно отличается возможностью автоматического присвоения категорий транзакциям и генерации отчетов, что делает его особенно полезным для планирования бюджета и контроля финансовых привычек.

Интерфейс приложения показан на рисунке 1.3.

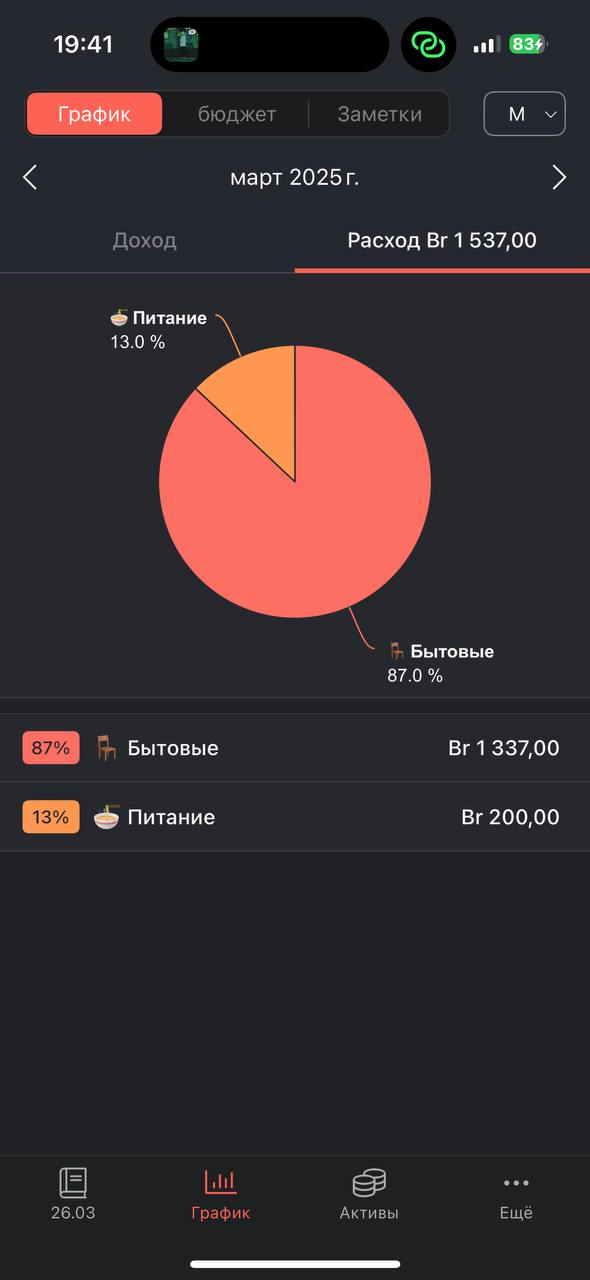
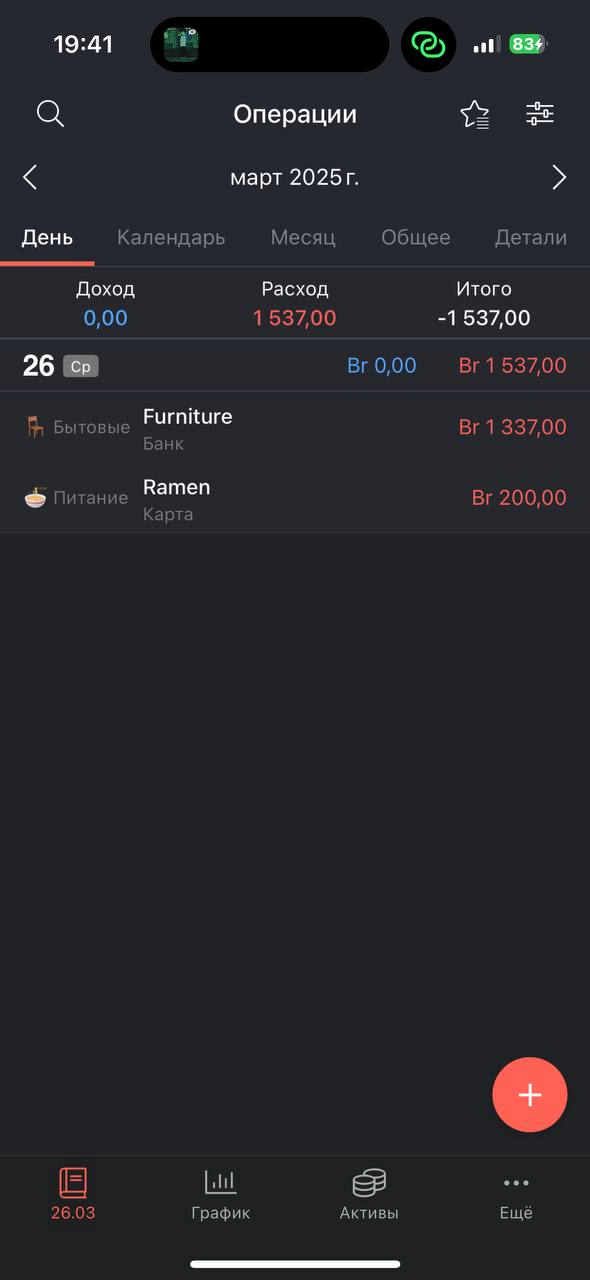


Рисунок 1.3 – Интерфейс приложения «Умный счёт»

Приложение также предлагает виджеты для быстрого доступа к финансовой информации прямо с главного экрана смартфона. Однако среди недостатков стоит отметить наличие в приложении рекламы, что может отвлекать пользователей и снижать общее удобство использования. Также пользователи иногда сообщают о проблемах с синхронизацией данных, что может вызывать недовольство и затруднять учет бюджета.

# Проектирование приложения

Проектирование программного средства — это процесс создания детального плана или концепции программного продукта до его реализации, включающий определение структуры и организации программы, разработку основных компонентов и модулей, их взаимосвязей и функций. На этапе проектирования выбираются архитектурные решения, которые могут быть монолитными, микросервисными, распределенными и т.д., а также подходящие паттерны проектирования для эффективной реализации задач. Проектирование интерфейсов, включающее создание пользовательских интерфейсов (UI) и программных интерфейсов (API), играет ключевую роль в обеспечении взаимодействия между компонентами системы и пользователем. Важно уделить внимание удобству и интуитивной понятности пользовательских интерфейсов, а также четкому и понятному взаимодействию между модулями через API. Определение функциональности системы включает разработку подробного описания всех функций и возможностей, которые должна предоставлять система, определение бизнес-логики, основных сценариев использования и требований к производительности.

## Проектирование базы данных

Модель данных — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Модель данных играет ключевую роль в проектировании базы данных, так как она определяет структуру хранения данных и способы их обработки.

Для данного курсового проекта была спроектирована база данных. Логическая модель базы данных представлена на рисунке 2.1.



Рисунок 2.1 – Модель БД

Созданная база данных содержит в себе 1 таблицу: Users.

## **Описание информационных объектов и ограничений целостности**

Далее будет описана структура таблицы. В таблице users хранятся данные для идентификации пользователей, зарегистрированных в приложении.

### **Таблица users**

Таблица users состоит из следующих полей:

* id. Хранит уникальный идентификатор пользователя;
* email. Хранит email пользователя;
* username. Хранит уникальное имя пользователя;
* password. Хранит захэшированный пароль;
* google\_id. Хранит уникальный идентификатор пользователя в системе google;
* is\_google\_account. Хранит булево значение, означающее был ли создан аккаунт до авторизации через google.

## Проектирование API

Для эффективной синхронизации работы локальной и глобальной БД на Android-клиенте было выбрано спроектировать серверное приложение. Оно должно обладать WEB API, который обеспечивает доступ к каждой сущности базы данных. Чтобы обмен данных между клиентами и сервером был успешным, необходимо передавать информацию в формате JSON.

Архитектурный стиль REST является оптимальным для разработки серверного приложения и позволяет обеспечить взаимодействие компонентов распределенного приложения в сети. Сервер разработан таким образом, чтобы иметь сервисы для взаимодействия с каждой сущностью базы данных.

При обращении к серверу с заданным маршрутом и описанным HTTP методом, сервер определяет, с какой сущностью базы данных необходимо взаимодействовать и какие действия должны быть выполнены.

Мы жестко разделяем уровни ответственности нашего приложения на 3 слоя, чтобы добиться расширяемости, лучшей тестируемости и гибкости приложения:

* Уровень доступа к данным;
* Бизнес логика приложения;
* Уровень представления. В нашем случае это URL, по которому будут обращаться Android и Web-клиенты, чтобы взаимодействовать с данными.

В целом, серверное приложение является неотъемлемой частью системы, гарантирующей эффективное и безукоризненное взаимодействие между различными компонентами в распределенной сети.

## Проектирование мобильного приложения

Клиентская часть приложения не менее важна, чем серверная. Это то, с чем будет контактировать пользователь, то, от чего в первую очередь зависят хорошие или плохие впечатления при использовании приложения. Здесь стоит понимать, что исходя из специфики проекта, в первую очередь разрабатывается мобильное приложение и самые строгие требования со стороны пользователя будут предъявляться именно к нему.

Точно так же, как и серверная часть приложения, как и практически любой другой проект, который имеет риск сильно разрастись в будущем, лучше всего разбить логику клиентской части приложения на части таким образом, чтобы на этой основе можно было каждую часть реализовывать последовательно и независимо от других частей.

Навигация – очень важный для мобильного приложения элемент, при её проектировании необходимо поддерживать логическую связь между экранами.

При первом запуске приложения пользователь попадает на Activity авторизации, с которой он может попасть в Activity регистрации. После успешной авторизации/регистрации пользователь попадет в Activity трат, где может изменить текущие траты на месяц.

В Activity настроек пользователь может просмотреть свои персональные данные и поменять их, изменить/сбросить пароль.

Activity учета займов позволяет добавить займ, в котором можно указать кто и сколько одолжил, либо же мы одолжили.

В Activity отчет пользователь может увидеть отчет его трат в процентном соотношении, сколько и куда он потратил за месяц или квартал.

Общая схема взаимодействия между страницами Android-клиента представлена на рисунке 2.2.

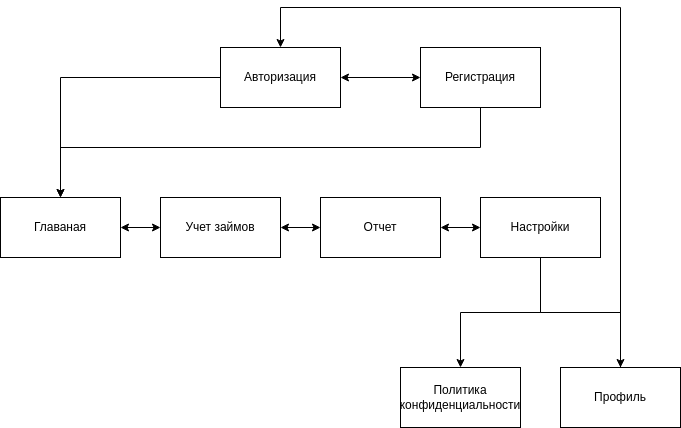


Рисунок 2.2 – Схема навигации для пользователя

Диаграмма вариантов использований представлена в приложении А.

## Выводы по главе «Проектирование программного продукта»

В данной главе было рассмотрено проектирование каждого отдельного компонента приложения.

По части мобильного приложения был определен основной функционал приложения, также были разработаны диаграмма вариантов использования приложения.

Была также спроектирована база данных, где были определены необходимые таблицы, охарактеризованы поля каждой из них. Также были установлены связи между таблицами, заключающиеся в создании первичных и внешних ключей.

При проектировании серверного приложения был выбран архитектурный стиль взаимодействия между клиентом и сервером – клиент-серверное взаимодействие. Это означает, что сервер предоставляет определенные услуги и ресурсы, которые клиент может запросить и использовать.