МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программной инженерии в информационных системах

Курсовая работа по курсу «Технологии программирования»

«MoneyKeeper»

Выполнили: студенты 3 курса, группы 3.1

Борисов А.Д., Никонов И. Е.

Воронеж 2019

Оглавление

Введение

Грамотное управление личными финансами в современном мире – это задача, которая лежит на каждом человеке. Сегодня все сложнее держать в голове все многочисленные расходы, а также сопоставлять их с доходами. Можно управляться с этой задачей, делая записи в блокнот, что не всегда удобно, или пользуясь онлайн-сервисами, что так же имеет свои недостатки, такие как наличие доступа к интернету и неуверенность в безопасности данных. В итоге наиболее удобным инструментом является мобильное приложение. Смартфон всегда под рукой и добавление новых записей о доходах и расходах – минутное дело.

Желаемое приложение должно облегчать жизнь пользователю, а не усложнять, это, к сожалению, могут далеко не все приложения подобного плана. Приложение должно быть легковесным и предоставлять только необходимую функциональность:

Систематизация расходов и доходов

Просмотр истории финансовой деятельности

Ненагруженный, интуитивно понятный интерфейс также является необходимой особенностью хорошего инструмента.

Данная курсовая работа посвящена разработке именно такого, простого в освоении, но в то же время выполняющего самые необходимые функции, приложения, способного облегчить финансовые сложности человека, не нагружая его сложным интерфейсом, а его телефон жесткими системными требованиями.

1 Анализ предметной области

1.1 Глоссарий

Доход — денежные средства, полученные пользователем, в результате какой-либо деятельности за определённый период времени.

Расход — денежные средства, потраченные пользователем на какую-либо категорию товаров или услуг.

Категория расходов — группа товаров или услуг.

Порог — максимальная сумма, которую пользователя намерен потратить в выбранной категории.

Акция — мероприятие компании по предоставлении скидочной цены на тот или иной продукт.

Валидность — соответствие требованиям, допустимость, правильность.

1.2 Анализ существующих решений

1.2.1 CoinKeeper

Является одним из ключевых игроков на рынке мобильных приложений по учёту расходов. При первом запуске сразу можно заметить большое количество настойчивой рекламы и навязывание акций компании, пример которых изображен на рис.1.

Приложение распространяется в GooglePlay на условно-бесплатной основе, что означает бесплатное использование всех функций в течение 15 дней, а в дальнейшем, при желании продолжать работу с приложением, необходимо приобрести полную версию.

Достоинства:

- Наличие основных функций;
- Подсказки по управлению;
- Автоматическая распланировка бюджета;
- Резервное копирование данных;
- Синхронизация с другими устройствами;
- Возможность установки пароля.

Недостатки:

- Невозможность представления отчета в виде графика;
- Медленная анимация, которая тормозит даже на современных устройствах.
- Отсутствие возможности получить визуальное представления о состоянии расходов\доходов в обобщенном плане в виде диаграммы на главном экране.

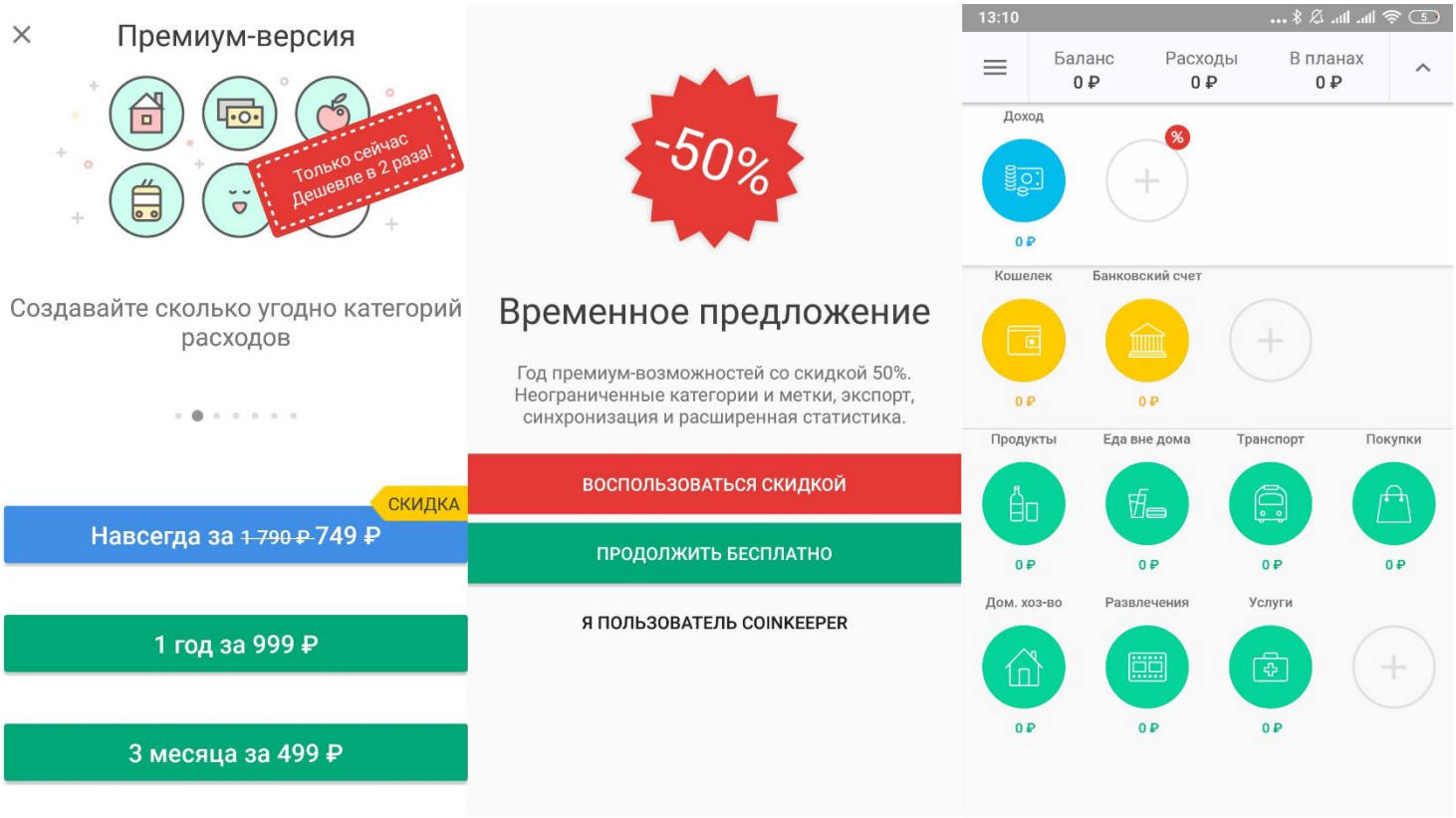

Рис. 1. Акции CoinKeeperи главный экран приложения

1.2.2 Monefy

Является самым визуально информативным из рассматриваемых приложений. При первом запуске сразу можно увидеть всю необходимую информацию в визуальном представлении (рис 2). Приложение не навязывает акции и дополнительные сервисы, а также распространяется по бесплатной модели с возможностью расширения имеющихся функций за дополнительную плату.

Достоинства:

- Наличие основных функций;
- Отсутствие необходимости в информационной справке;
- Синхронизация с другими устройствами;
- Возможность установки пароля.

Недостатки:

- Невозможность представления отчета в виде графика;

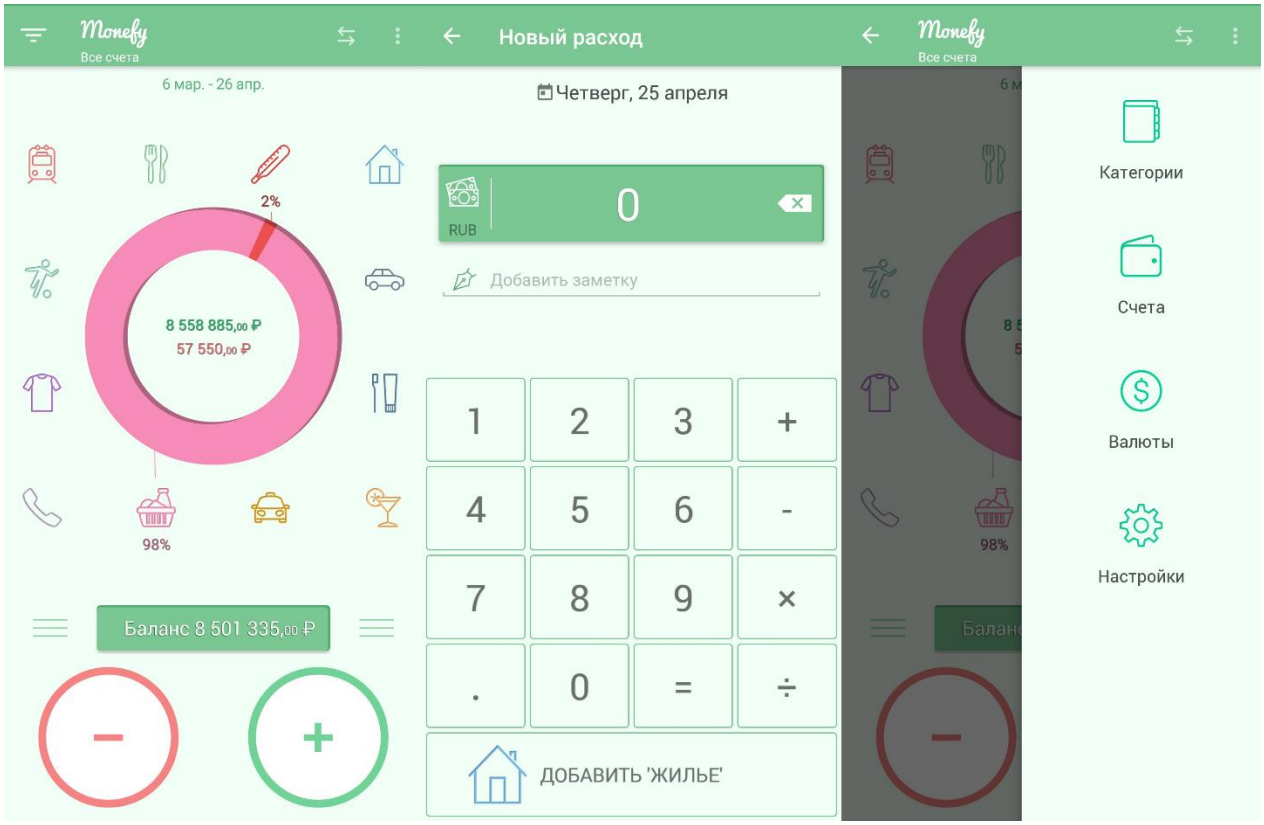

Рис.2. Интерфейс приложения Monefy

1.2.3 Fingen

Является приложениемс самым большим количеством функций из рассматриваемых. Оно может сканировать и анализировать чеки, проводить автоматический учет расходов и доходов. Имеет перегруженный интерфейс. На данные момент распространяется по бесплатной модели с возможностью платного расширения в виде отчетов.

Достоинства:

- Максимально-возможный функционал;
- Возможность установки пароля.
- Сканирование чеков;
- Наличие справки;

- Автоматический учет доходов/расходов;

Недостатки:

- Невозможность представления отчета в виде графика;
- Отчеты только в платной версии;
- Сложный интерфейс

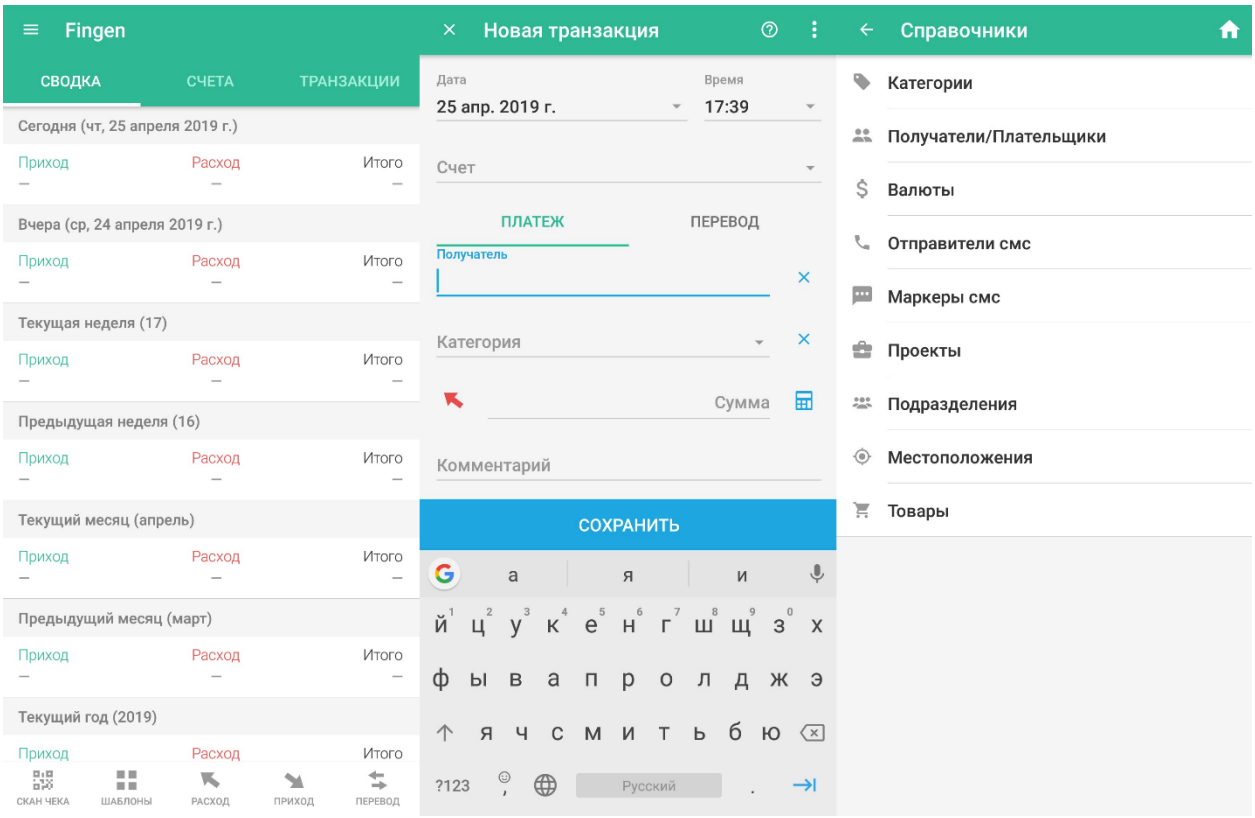

Рис.3. Интерфейс приложения Fingen.

1.3 Постановка задачи

Цель курсовой работы: реализовать Android приложение, которое позволяет в отличии от рассмотренных выше аналогов осуществлять предоставление статистики в виде отчета, а также иметь визуально информативный интерфейс по типу Monefy, который позволяет получать всю необходимую информацию в виде диаграммы с главного экрана. Приложение должно отслеживать личные финансы и иметь следующие основные функции:

- Добавление Доходов
- Удаление Доходов
- Добавление Расходов
- Удаление Расходов
- Добавление Категорий Расходов
- Удаление Категорий Расходов
- Добавление порогов расходов
- Удаление порогов расходов
- Просмотр статистики

1.4 Анализ задачи

1.4.1 Варианты использования приложения

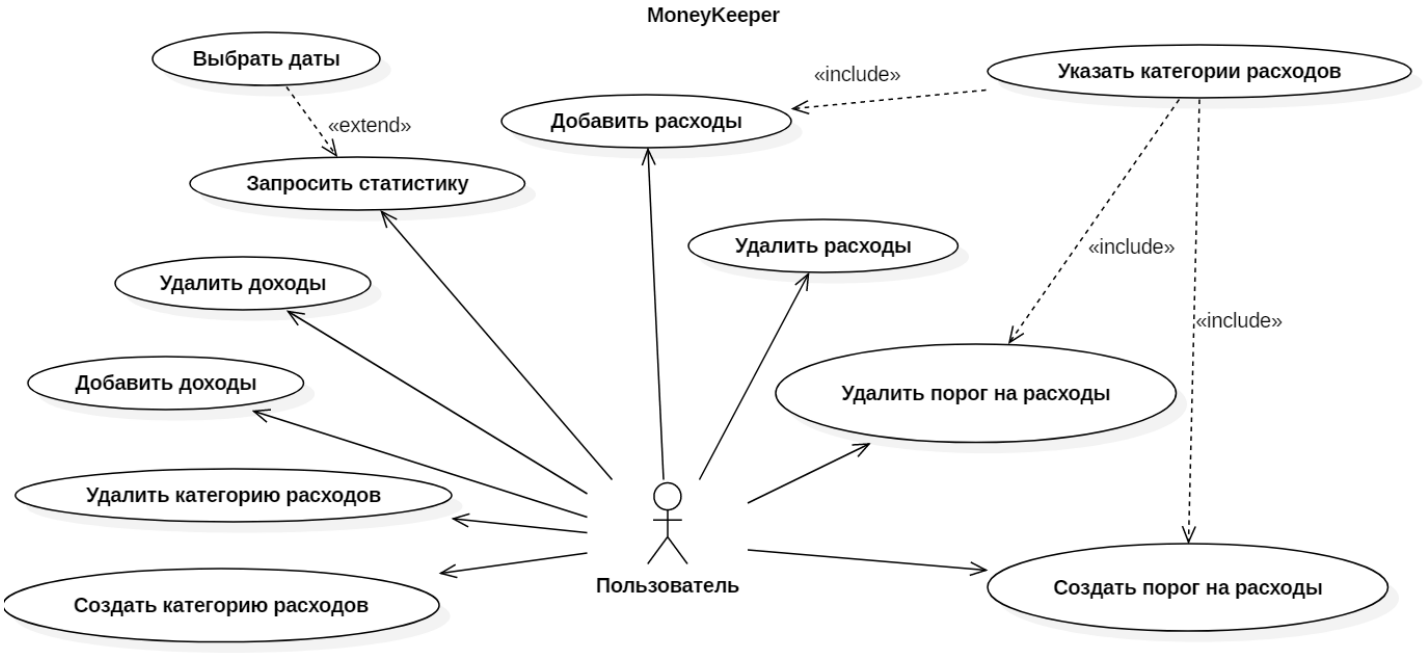

Рисунок 1. Диаграмма прецедентов.

При взаимодействии с системой у пользователя есть определенный список возможностей, который более наглядно изображен на рисунке 1:

- Добавление Доходов
- Удаление Доходов
- Добавление Расходов
    - При добавлении расходов пользователь должен указать категорию
- Удаление Расходов
- Добавление Категорий Расходов
- Удаление Категорий Расходов
- Добавление порогов расходов

    - При добавлении порога на расход пользователь должен указать конкретную категорию расходов
- Удаление порогов расходов
    - При удалении порога на расход пользователь должен выбрать конкретную категорию расходов
- Просмотр статистики
    - В качестве расширения пользователь может указать дату по которой он хочет получить статистику

1.4.2 Взаимодействие компонентов системы

Рисунок 2. Диаграмма последовательности для просмотра статистики.

Рисунок 3. Диаграмма коммуникации для просмотра статистики.

На рисунке 3, показана диаграмма коммуникации, на которой явно указываются отношения между объектами при просмотре статистики в приложении, а на рисунке 2 изображена диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов.

Для просмотра статистики пользователь обращается к графическому интерфейсу приложения, который в свою очередь запрашивает статистику у менеджера статистики. Менеджер статистики делает запрос к базе данных, она же в свою очередь возвращает данные менеджеру статистики, который отправляет эти данные на графический интерфейс, который в свою очередь показывает пользователю запрашиваемую статистику.

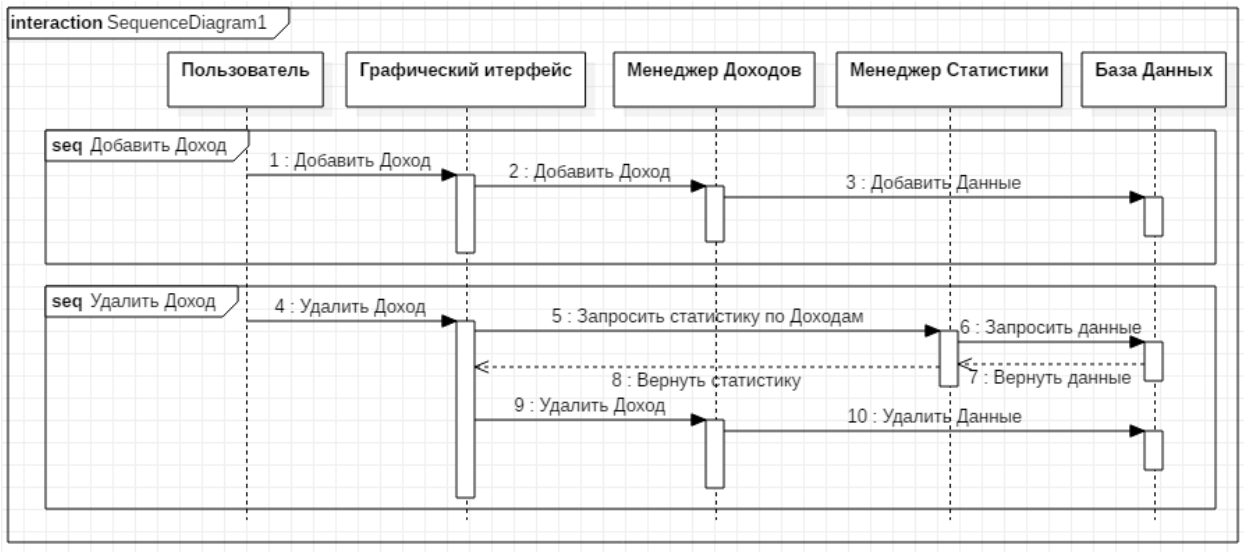

Рисунок 4. Диаграмма последовательности для добавления/удаления дохода.

Рисунок 5. Диаграмма коммуникации для добавления/удаления дохода.

На рисунке 5, показана диаграмма коммуникации, на которой явно указываются отношения между объектами при добавлении/удалении дохода в приложении, а на рисунке 4 изображена диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов.

Для добавления нового дохода пользователь обращается к графическому интерфейсу приложения, который в свою очередь просит менеджер доходов добавить соответствующий доход. Менеджер доходов осуществляет добавление данных в базу данных.

Для удаления существующего дохода пользователь обращается к графическому интерфейсу приложения, который в свою очередь делает запрос

по доходам к менеджеру статистики. Менеджер статистики делает запрос по поводу данных к базе данных, которая возвращает данные менеджеру статистики. Менеджер посылает статистику по доходам обратно графическому интерфейсу, который выводит пользователю его, после чего просит менеджер доходов удалить выбранный пользователем доход. Менеджер дохода в свою очередь удаляет данные соответствующего дохода из базы данных.

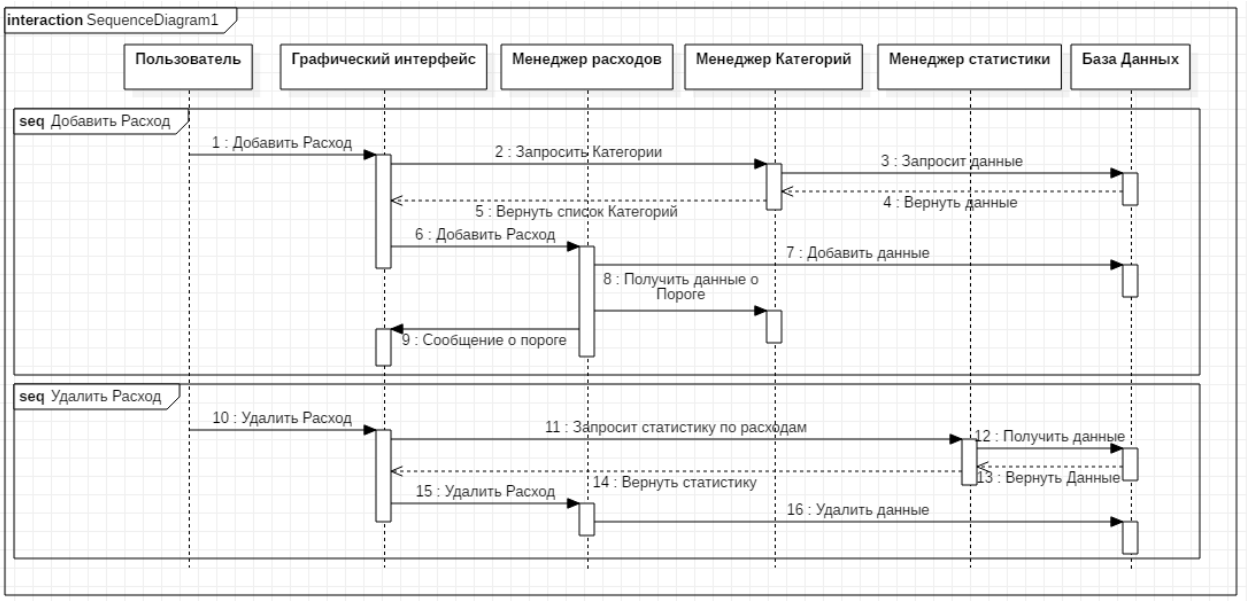

Рисунок 6. Диаграмма последовательности для добавления/удаления расхода.

Рисунок 7. Диаграмма коммуникации для добавления/удаления расхода.

На рисунке 7, показана диаграмма коммуникации, на которой явно указываются отношения между объектами при добавлении/удалении расхода в приложении, а на рисунке 6 изображена диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов.

Пользователь обращается к графическому интерфейсу с целью добавить расход, интерфейс запрашивает у менеджера категории, который в свою очередь запрашивает их у базы данных, после чего база данных возвращает их менеджеру категорий, а он графическому интерфейсу. После чего графический интерфейс просит менеджер расходов добавить расход пользователя в базу данных. После чего менеджер расходов сообщает менеджеру категорий информацию для порога расходов для данной категории и выводит сообщение о пороге в графический интерфейс приложения.

Если пользователь обращается к графическому интерфейсу с целью удалить расход, интерфейс в свою очередь делает запрос по расходам к менеджеру статистики. Менеджер статистики делает запрос по поводу данных к базе данных, которая возвращает данные менеджеру статистики. Менеджер посылает статистику по расходам обратно графическому интерфейсу, который выводит пользователю его, после чего просит менеджер расходов удалить выбранный пользователем расход. Менеджер расходов в свою очередь удаляет данные соответствующего расхода из базы данных.

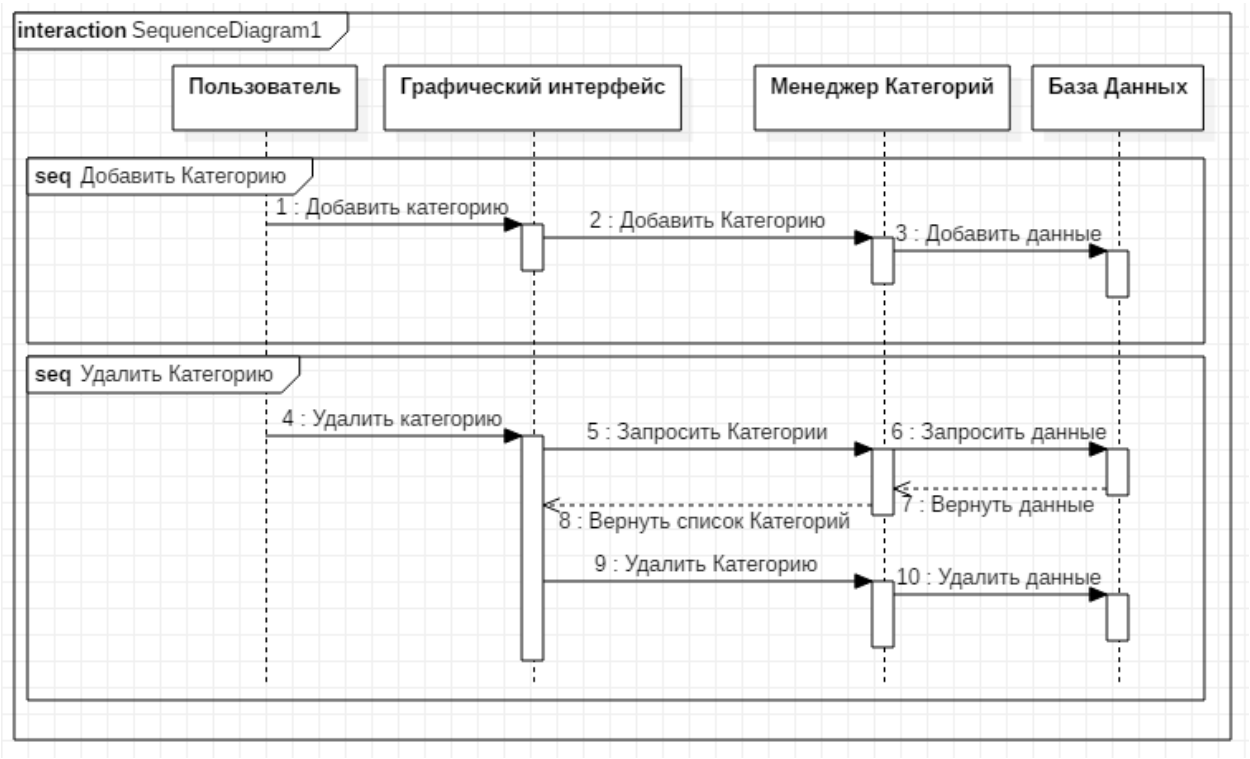

Рисунок 8. Диаграмма последовательности для добавления/удаления категории.

Рисунок 9. Диаграмма коммуникации для добавления/удаления категории.

На рисунке 9, показана диаграмма коммуникации, на которой явно указываются отношения между объектами при добавлении/удалении категории в приложении, а на рисунке 8 изображена диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов.

Для добавления новой категории пользователь обращается к графическому интерфейсу приложения, который в свою очередь просит менеджер категорий добавить соответствующую категорию. Менеджер категорий осуществляет добавление данных в базу данных.

Для удаления существующей категории пользователь обращается к графическому интерфейсу приложения, который в свою очередь делает запрос по категориям к менеджеру категорий. Менеджер категорий делает запрос по поводу данных к базе данных, которая возвращает данные менеджеру категорий. Менеджер посылает категории обратно графическому интерфейсу, который выводит пользователю его, после чего просит менеджер удалить выбранный пользователем категорию. Менеджер категорий в свою очередь удаляет данные из базы данных.

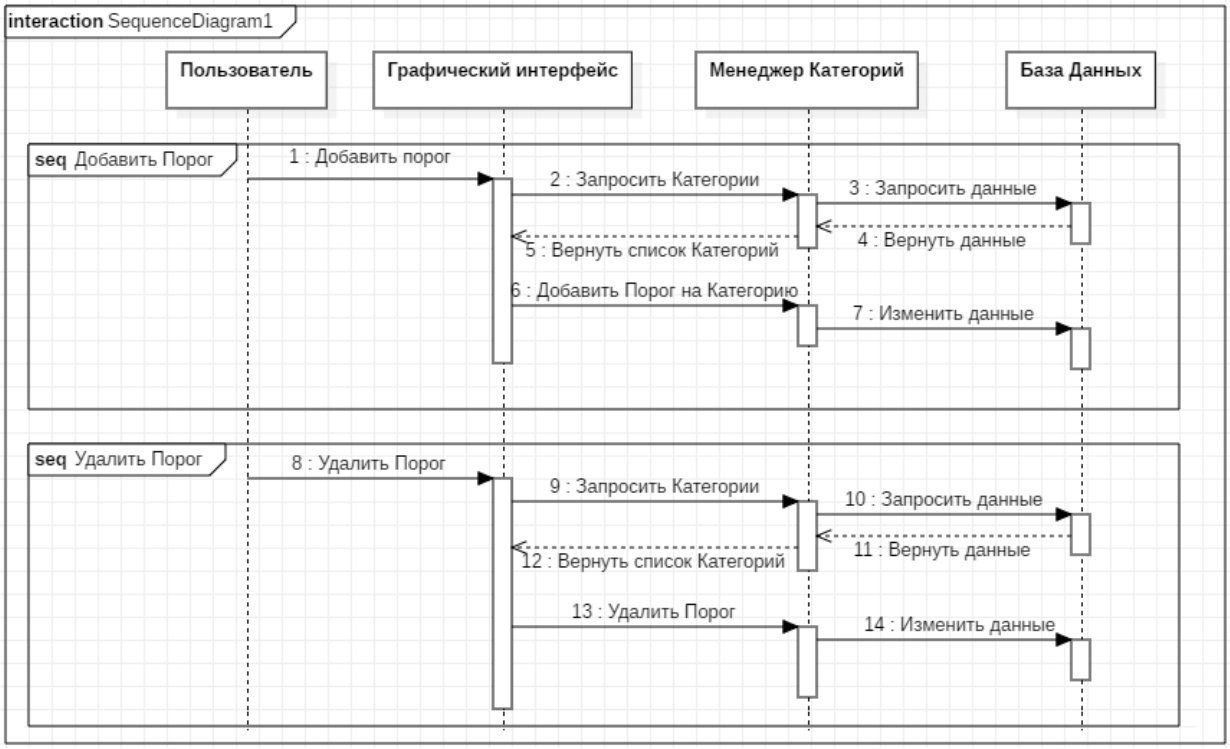

Рисунок 10. Диаграмма последовательности для добавления/удаления порога.

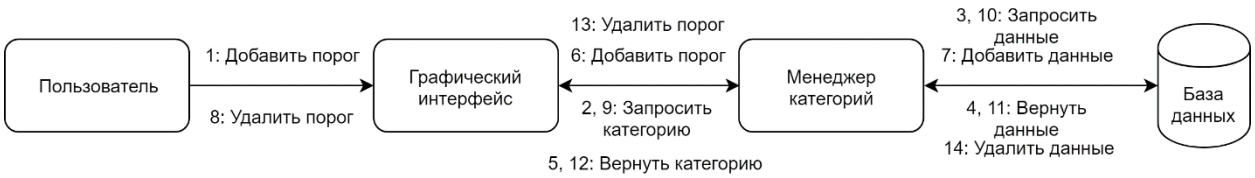

Рисунок 11. Диаграмма коммуникации для добавления/удаления порога.

На рисунке 11, показана диаграмма коммуникации, на которой явно указываются отношения между объектами при добавлении/удалении порога в приложении, а на рисунке 10 изображена диаграмма последовательности, на которой изображено упорядоченное во времени взаимодействие объектов.

Для добавления нового порога пользователь обращается к графическому интерфейсу приложения, который в свою очередь просит менеджер категорий добавить соответствующую категорию. Менеджер категорий делает запрос по поводу данных к базе данных, которая возвращает данные менеджеру категорий. Менеджер посылает категории обратно графическому интерфейсу, который выводит пользователю его, после чего просит менеджер добавить выбранный пользователем порог на расходы для данной категории. Менеджер в свою очередь добавляет данные в базу данных.

Для удаления существующего порога пользователь обращается к графическому интерфейсу приложения, который в свою очередь делает запрос по категориям к менеджеру категорий. Менеджер категорий делает запрос по поводу данных к базе данных, которая возвращает данные менеджеру категорий. Менеджер посылает категории обратно графическому интерфейсу, который выводит пользователю его, после чего просит менеджер удалить выбранный пользователем порог для категории. Менеджер категорий в свою очередь удаляет данные из базы данных.

1.4.3 Варианты состояния системы

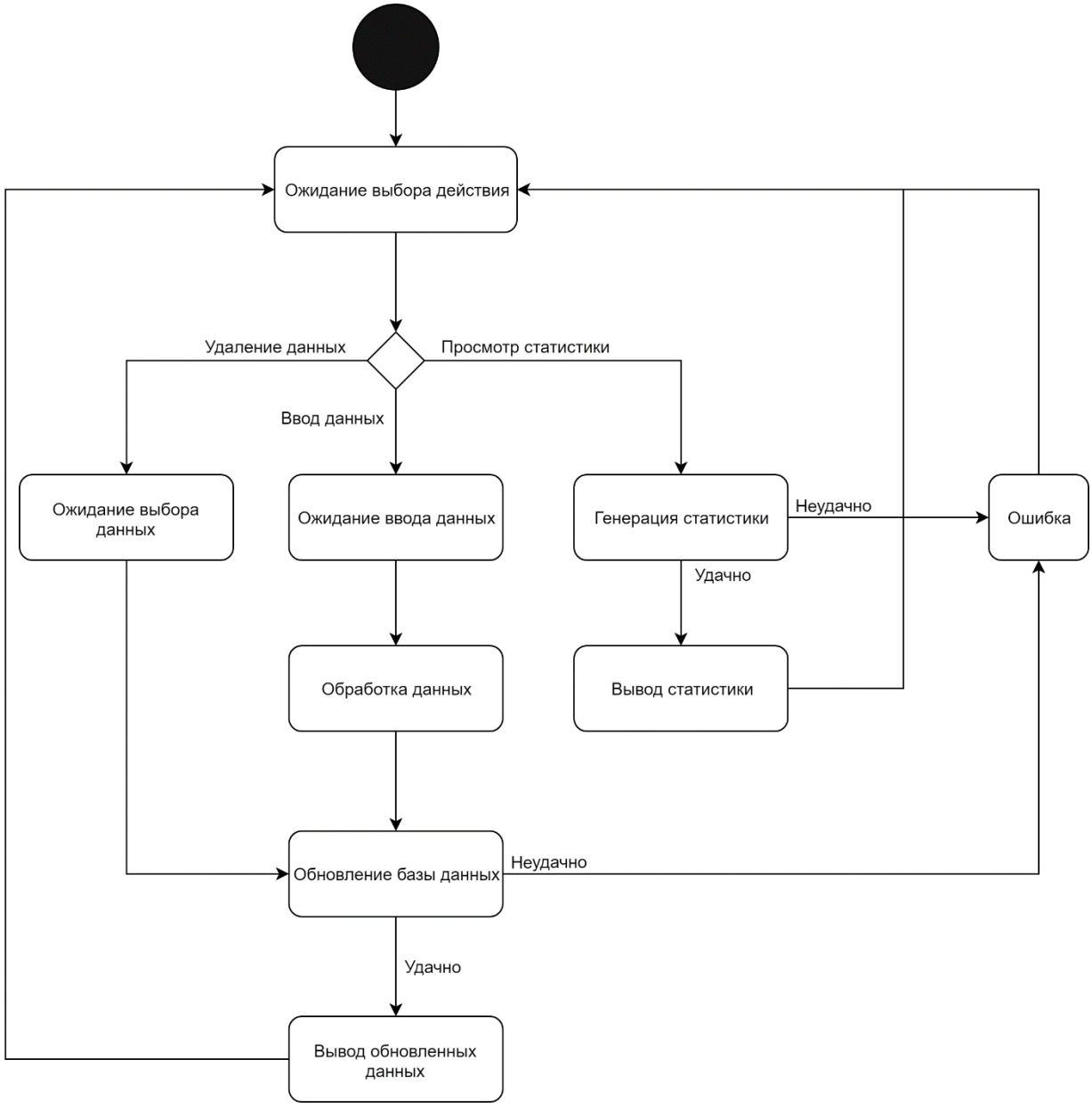

Рисунок 12. Диаграмма состояния.

Диаграмма состояний, изображенная на Рисунке 12, отражает возможные состояния системы. При запуске приложения система находится в ожидании выбора действия. В зависимости от выбора пользователя возможны 3 основные цепочки состояний:

- Ввод данных
- Удаление данных

- Просмотр статистики

Если пользователь выбирает опцию ввода данных, система переходит в состоянии ожидания ввода данных, после ввода данных пользователем система переходит в состояния обработки введенных данных, затем переходит в состояние обновления базы данных. Если все данные прошли проверки на валидность успешно, то система переходит в состояние вывода обновленных данных и переходит в состояния ожидания выбора новой опции от пользователя. Если же данные не прошли проверки, то система переходит в состояние вывода ошибки после чего переходит в ожидание выбора новой опции от пользователя.

Если пользователь выбирает опцию удаления данных, то система переходит в состояние ожидание выбора данных для удаления, после чего переходит в состояние обновления базы данных, в случае не успешного обновления переходит в состояние отображения ошибки и после переходит в состояние ожидания выбора действия. Если же обновление базы данных прошло успешно, то переходит в состояние отображения обновленных данных, после чего переходит в состояние ожидания выбора действия.

Если пользователь выбирает опцию просмотра статистики, то система переходит в состояние генерации статистики, в случае не успешной генерации переходит в состояние отображения ошибки и затем ожидания выбора действия. Если же статистика была сгенерирована без ошибок, то переходит в состояние вывода статистики и переходит в состояние ожидания выбора действия пользователя.

1.4.4 Варианты действия в системе

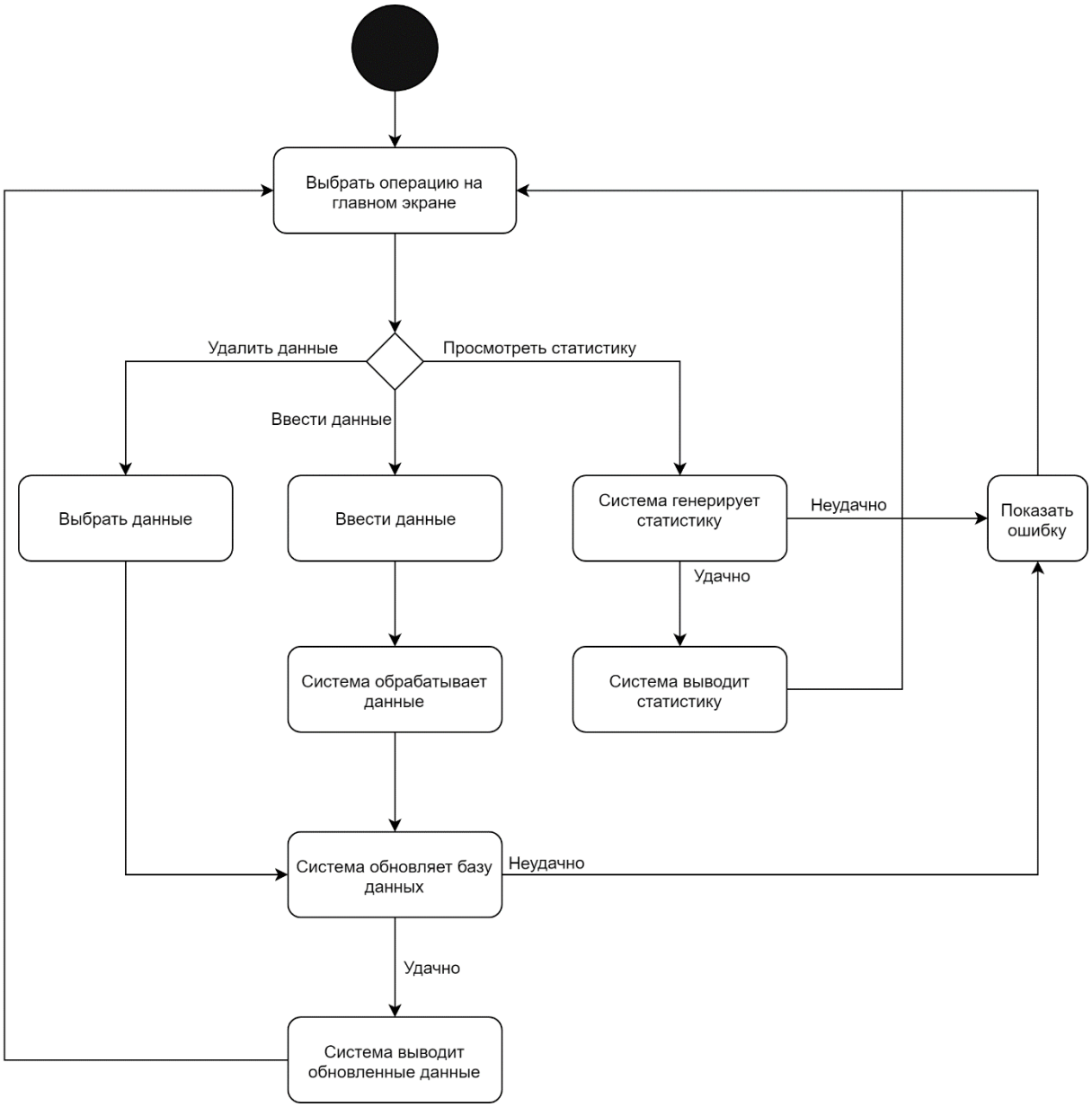

Рисунок 13. Диаграмма активности.

Диаграмма активности, изображенная на Рисунке 13, отражает возможные действия, состояния которых описаны на диаграмме состояния (Рисунок 12). При запуске приложения пользователь должен выбрать действие над системой. В зависимости от выбора пользователя возможны 3 основные цепочки действий:

- Ввод данных

- Удаление данных
- Просмотр статистики

Если пользователь выбирает опцию ввода данных, то ему необходимо ввести данные для добавления, после чего система занимается обработкой введенных пользователем данных и в последствие осуществляет обновление базы данных. В случае ошибки, система показывает пользователю окно ошибки и пользователь переходит на начальный экран выбора операции. Если же обновление базы данных прошло успешно, то система выводит пользователю обновленные данные и пользователь переходит на выбор операции на главном окне приложения.

Если пользователь выбирает удаление данных на главном экране, то ему необходимо выбрать данные, которые он хочет удалить из системы, после чего система обновляет базу данные, в случае возникновения ошибки система выведет окно ошибки пользователю с информацией, после чего пользователь переходит на главный экран, для выбора действия. Если обновление базы данных прошло успешно, то система выведет пользователю обновленные данные и пользователь перейдет на главный экран для выбора следующего действия.

Если пользователь выберет операцию просмотра статистики, то система сгенерирует ему статистику, в случае возникновении ошибки система покажет пользователю окно с ошибкой после чего пользователь перейдет на главный экран для выбора следующего действия. В случае успешной генерации статистики система выведет пользователю статистики, после чего пользователь переходит на главный экран для дальнейшего выбора действия.

1.4.5 Модель базы данных

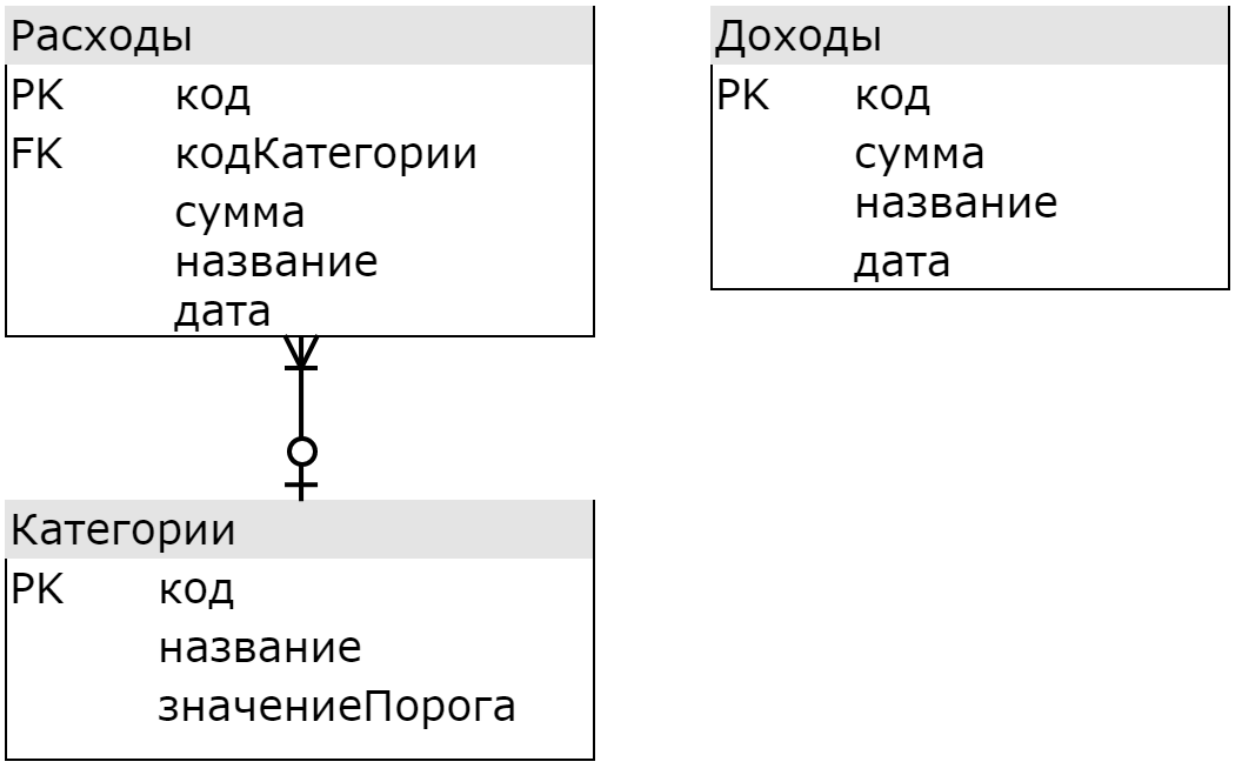

Рисунок 15. Схема базы данных

На рисунке 15 представлена схема базы данных для разрабатываемого приложения по контролю за расходами.

Данная база данных состоит из 3 таблиц:

- Расходы
- Доходы
- Категории

Таблица “Расходы” содержит следующие атрибуты:

- Код, который является первичным ключом
- Код категории, который является вторичным ключом, ссылающимся на таблицу “Категории”
- Сумма, которая определят размер расхода
- Название, которое является неким описанием расхода
- Дата, определяющие время записи расхода

Таблица “Категории” содержит следующие атрибуты:

- Код, который является первичным ключом
- Название самой категории
- Значение порога, которое определяет максимальное значение суммарного расхода по данной категории

Между таблицами “Расходы” и таблицей “Категории” имеется связь многие ко многим. Каждый расход обязательно относится к какой то одной категории и к каждой категории может относится один и более расход, а может не одного не быть.

Таблица “Доходы” содержит следующие атрибуты:

- Код, который является первичным ключом
- Сумма, которая определят размер дохода
- Название, которое является неким описанием дохода
- Дата, определяющие время записи дохода

1.4.6 Интерфейсная реализация приложения

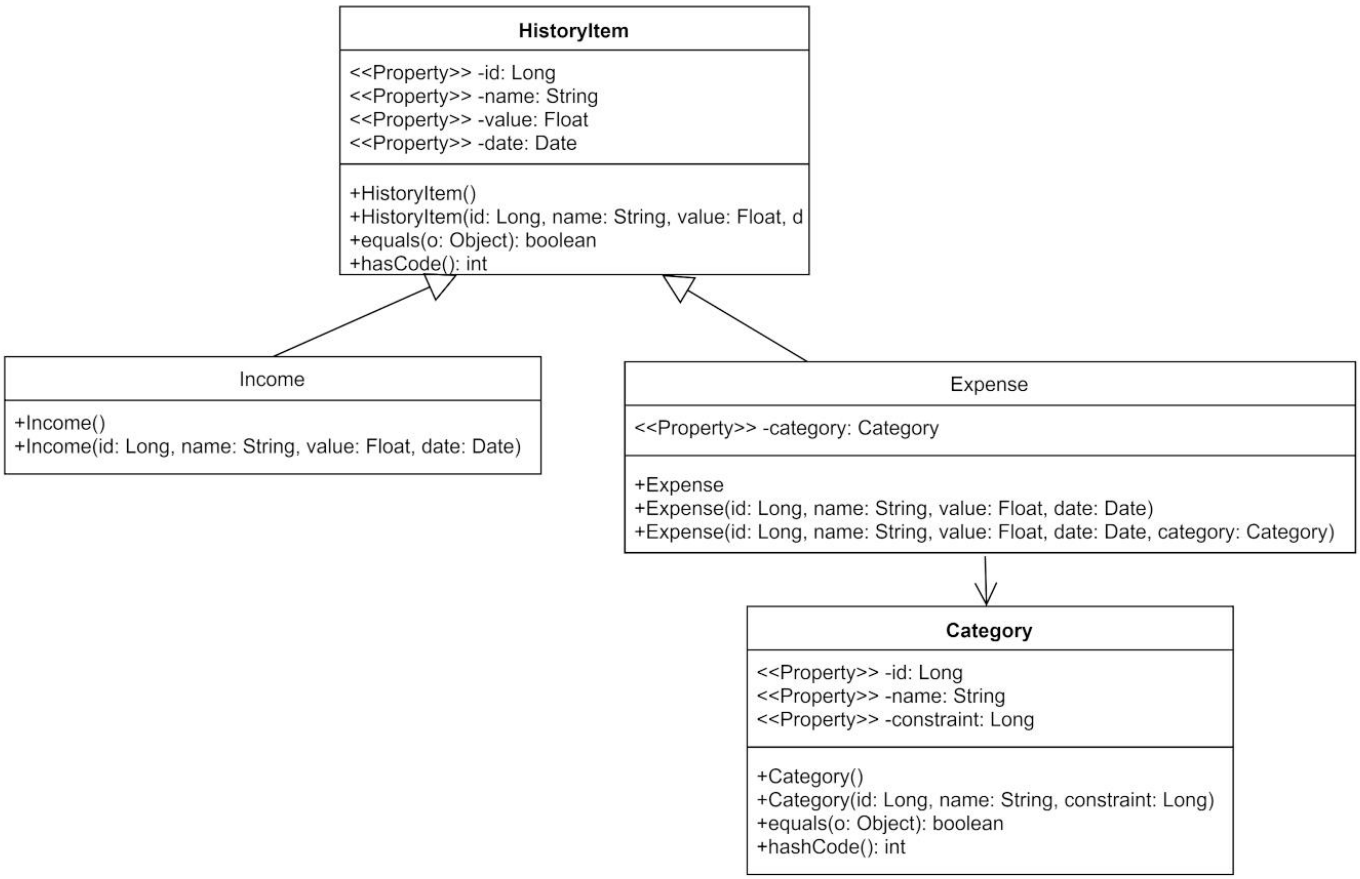

Рисунок 16. Диаграмма классов

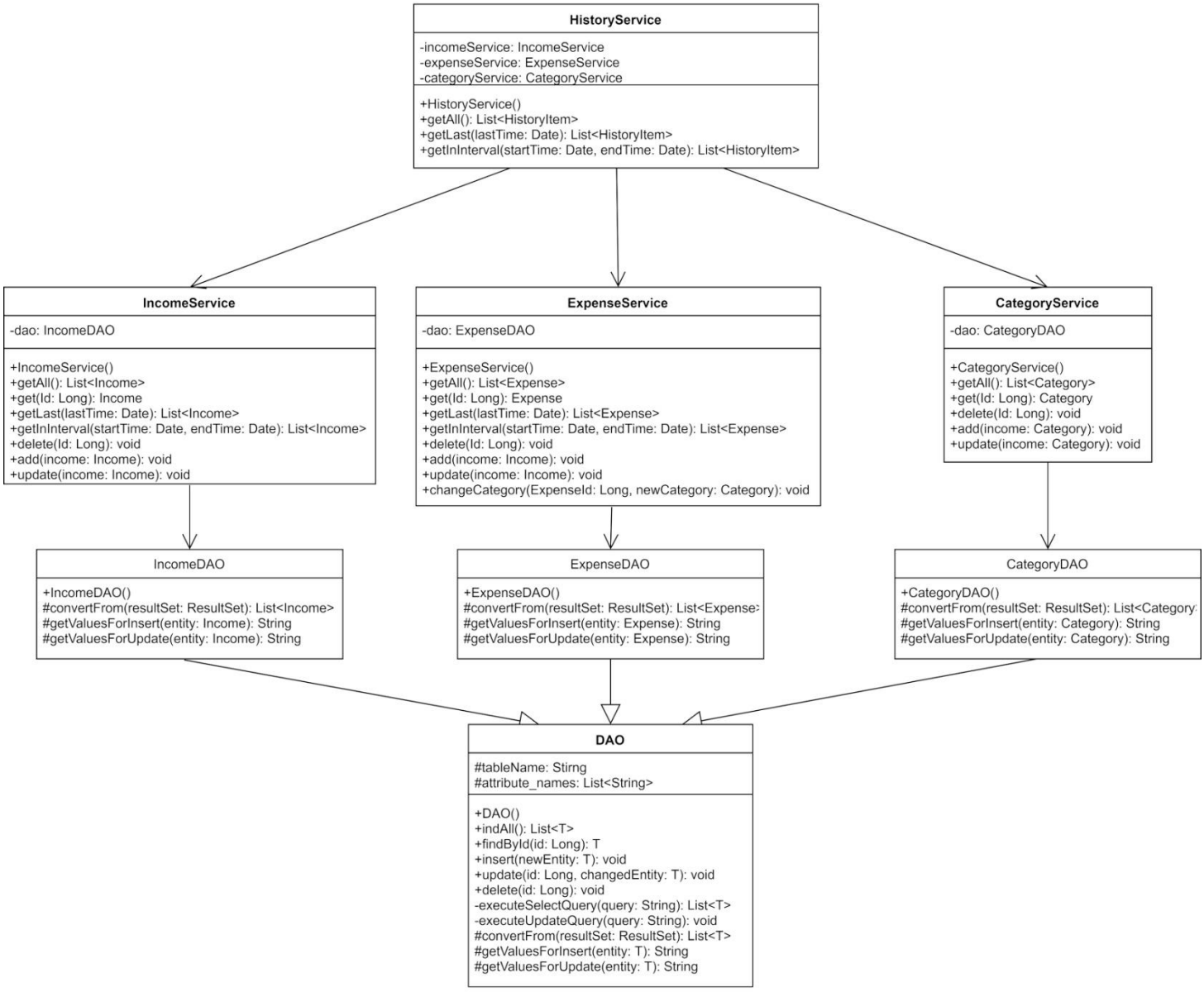

Рисунок 17. Диаграмма развертывания.

1.4.7 Развертывание приложения

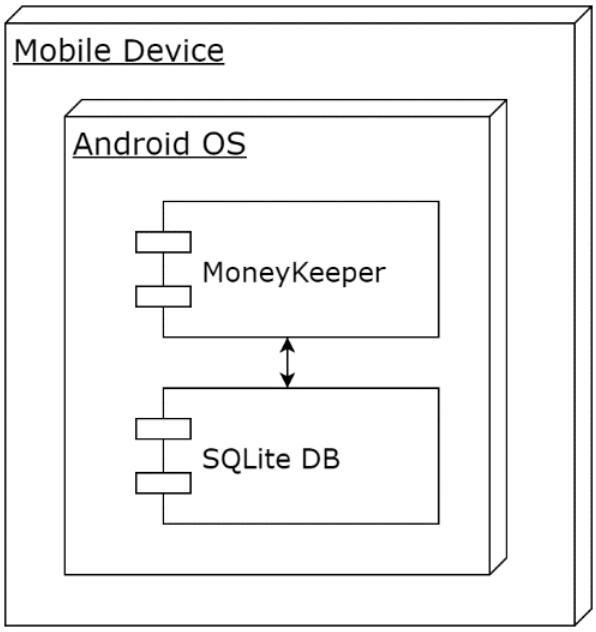

Рисунок 18. Диаграмма развертывания.

На Рисунке 18. представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты(«артефакты») работают на каждом узле и как различные части этого комплекса соединяются друг с другом.

Для разрабатываемого мобильного приложения узлом устройства является мобильное устройство, а в качестве узла среды выполнения выступает операционная системаAndroid. В операционной системе развернуты следующие программные компоненты:

- Само приложение MoneyKeeper
- База данных SQLite