|  |  |
| --- | --- |
|  |  |
| MoneyKeeper  Наименование продукта  MK  Сокращенное наименование продукта | |

**СОГЛАСОВАНО**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

**РАЗРАБОТЧИК**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

Воронеж 2019

**Содержание**

[1. Общие сведения 3](#_Toc10386313)

[1.1. Наименование Заказчика 3](#_Toc10386314)

[1.2. Наименование Исполнителя 3](#_Toc10386315)

[1.3. Основание для разработки 3](#_Toc10386316)

[1.4. Плановые сроки начала и окончания работ 3](#_Toc10386317)

[1.5. Термины и сокращения 3](#_Toc10386318)

[2. Назначение и цели создания приложения 4](#_Toc10386319)

[2.1. Назначение приложения 4](#_Toc10386320)

[2.2. Цели создания приложения 4](#_Toc10386321)

[3. Требования к приложению 5](#_Toc10386322)

[3.1. Требования к приложению в целом 5](#_Toc10386323)

[3.1.1. Требования к структуре 5](#_Toc10386324)

[3.1.2. Показатели назначения системы 7](#_Toc10386325)

[3.1.3. Требования к безопасности и защите информации 8](#_Toc10386326)

[3.1.4. Требования к численности и квалификации персонала 8](#_Toc10386327)

[3.1.5. Требования к патентной чистоте 8](#_Toc10386328)

[3.1.6. Требования к масштабируемости и открытости 8](#_Toc10386329)

[3.2. Функциональные требования 8](#_Toc10386330)

[3.3. Обработка ошибок 9](#_Toc10386331)

[3.4. Интерфейс 10](#_Toc10386332)

[3.4.1. Основные требования 10](#_Toc10386333)

[3.4.2. Дизайн и навигация 11](#_Toc10386334)

[3.5. Требования к видам обеспечения 12](#_Toc10386335)

[3.5.1. Требования к программному обеспечению 12](#_Toc10386336)

[3.5.2. Требования к лингвистическому обеспечению 12](#_Toc10386337)

[3.5.3. Требования к техническому обеспечению 12](#_Toc10386338)

[4. План тестирования 13](#_Toc10386339)

[5. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 15](#_Toc10386340)

# Общие сведения

## Наименование Заказчика

Ассистент Тарасов Вячеслав Сергеевич, кафедра программирования и информационных технологий.

## Наименование Исполнителя

Студент Борисов Александр Дмитриевич, кафедра программирования и информационных технологий.

Студент Никонов Иван Евгеньевич, кафедра программирования и информационных технологий.

## Основание для разработки

Необходимость заказчика в наличие мобильного приложения, позволяющего вести учет расходов и доходов.

## Плановые сроки начала и окончания работ

Плановый срок начала работ – Февраль 2019 г.

Плановый срок окончания работ – Июнь 2019 г.

## Термины и сокращения

Таблица 1 - Термины и аббревиатуры специального назначения

| Термин | Полная форма |
| --- | --- |
| Доходы | денежные средства, полученные пользователем в результате какой-либо деятельности за определённый период времени |
| Расходы | денежные средства, потраченные пользователем  на какую-либо группу товаров или услуг |
| Инкрементные затраты и поступления | разница между понесенными расходами и поступлениями |
| Расчетный период | отрывок времени, в течение которого ведется учёт бюджета |
| Категория | группа товаров или услуг |
| Предел расходов | намерение пользователя потратить не более конкретной, заранее указанной, суммы |
| Пороговое значение | денежная сумма , которую нельзя превышать |

# Назначение и цели создания приложения

## Назначение приложения

Назначением системы является автоматизация процесса учета расходов и доходов.

## Цели создания приложения

. Упростить учет доходов и расходов конечного пользователя

# Требования к приложению

## Требования к приложению в целом

Разрабатываемое приложение должно удовлетворять следующим основным требованиям:

* + Стабильная работа на базе операционной системы android 4.1 и выше;
  + Создание интуитивного пользовательского интерфейса;
  + Возможность выполнения основных задач приложения:
    - Добавление/Удаление расходов;
    - Добавление/Удаление доходов;
    - Добавление/Удаление порогов на расходы;
    - Просмотр истории по расходам.

### Требования к структуре

Приложение должно иметь архитектуру, основанную на шаблоне проектирования Model-View-Controller, а также иметь разделение на front-end и back-end, общение которых происходит с помощью REST API.

Шаблон MVC делится на три части и в реализуемом приложение должен быть представлен в виде:

* Базы данных (Model), представляющей собой фундаментальные данные, необходимые для работы приложения.
* Графического интерфейса (View), являющегося подписчиком на событие изменения значений свойств, запросов или команд, предоставляемых бизнес-логикой. В случае, если произошли какие-либо изменения свойств или данных базы данных, то происходит оповещение об этом, и пользовательский интерфейс, в свою очередь, запрашивает обновленное значение свойства из блока бизнес-логики. В случае, если пользователь воздействует на какой-либо элемент интерфейса, вызывается соответствующая команда, предоставленная бизнес-логикой.
* Бизнес-логика (Controller), реализуется на языке Java и предоставляет обёртку данных из базы данных, которые подлежат связыванию. Таким образом, она является посредником между графическим интерфейсом и базой данных, предоставляя пользователю возможность для обработки вводимой им информации, извлекая данные из базы данных и форматируя их для отображения в графическом интерфейсе.

Базовая архитектура приложения представлена на Рис. 1.

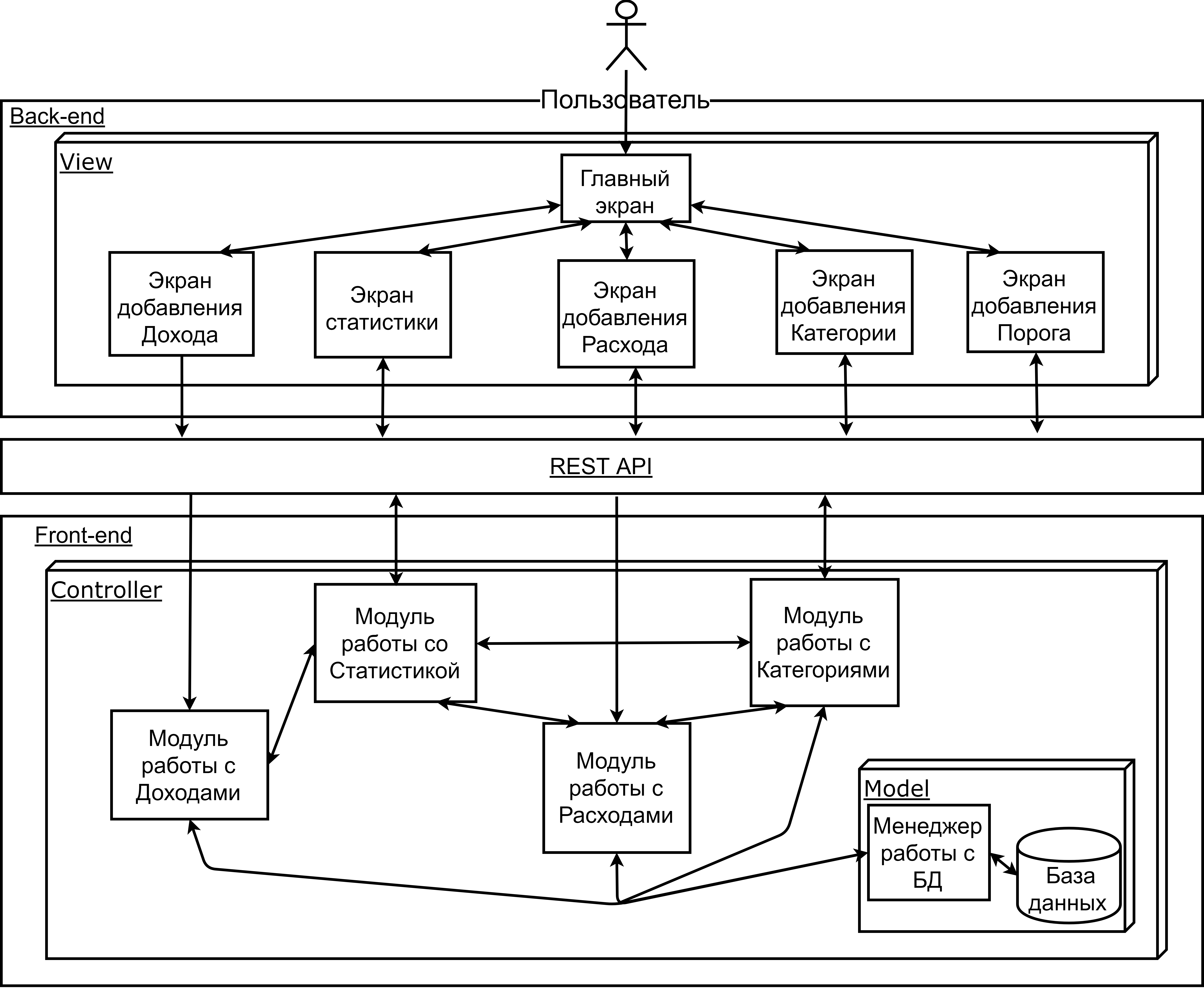


Рис. 1 – Архитектура приложения

### Показатели назначения системы

* Возможность создавать категории расходов;
* Возможность удалять категории расходов;
* Возможность добавлять расходы;
* Возможность добавлять доходы;
* Возможность просмотра статистики;
* Возможность добавления порогового значения расходов;
* Возможность удаления порогового значения расходов.

### Требования к безопасности и защите информации

#### Требования к аутентификации

В приложение отсутствует аутентификация. Подразумевается наличие только одного пользователя на одно мобильное устройство.

#### Требования к защите информации от несанкционированного доступа

Приложение должно предусматривать возможность защиты от самых простых попыток получения доступа к информации пользователя, в частности с помощью SQL инъекций.

### Требования к численности и квалификации персонала

Команда состоит из двух человек, каждый из которых является студентом старших курсов факультета компьютерных наук Воронежского государственного университета, имеет представление о базовой мобильной разработки и владеет языком Java на базовом уровне.

### Требования к патентной чистоте

Данный проект должен не нарушать никаких лицензий и патентов. В случаи нарушения всю ответственность несет сторона Исполнителя.

### Требования к масштабируемости и открытости

Проект должен предоставлять возможность добавлять новую функциональность с минимальным изменением существующего кода. Код находится в свободном доступе.

## Функциональные требования

Приложение должно удовлетворять следующим основным Функциональным требованиям:

* Приложение должно предоставлять возможность хранить следующую информацию о расходах:
  + сумма;
  + дата совершения расхода;
  + категория расхода;
  + комментарий.
* Приложение должно предоставлять возможность хранить следующую информацию о доходах:
  + сумма;
  + дата совершения расхода;
  + комментарий.
* Приложение должно предоставлять возможность создавать новые расходы/доходы/категории/пороговые суммы;
* Приложение должно предоставлять возможность просматривать список расходов/доходов/категорий;
* Приложение должно предоставлять возможность удалять ранее созданные расходы/доходы/категорий/пороговые суммы;
* После удаления расхода/дохода приложение должно осуществлять перерасчет затрагиваемых сумм;
* Приложение должно иметь возможность отобразить базовую статистику по расходам;

## Обработка ошибок

В случае возникновения ошибок пользователь должен получать соответствующие сообщение.

Приложение должно поддерживать обработку следующих основных ошибок:

* Некорректно введенные данные;
* Невозможность подключится к базе данных.

## Интерфейс

### Основные требования

Пользовательский интерфейс приложения должен удовлетворять следующим основным требованиям:

* Отсутствие нагромождений;
* Отсутствие броских цветов;
* Возможность перехода на все основные экраны приложения с главного экрана.

### Дизайн и навигация

Базовый дизайн приложения и навигация по приложению продемонстрированы на Рис. 2.

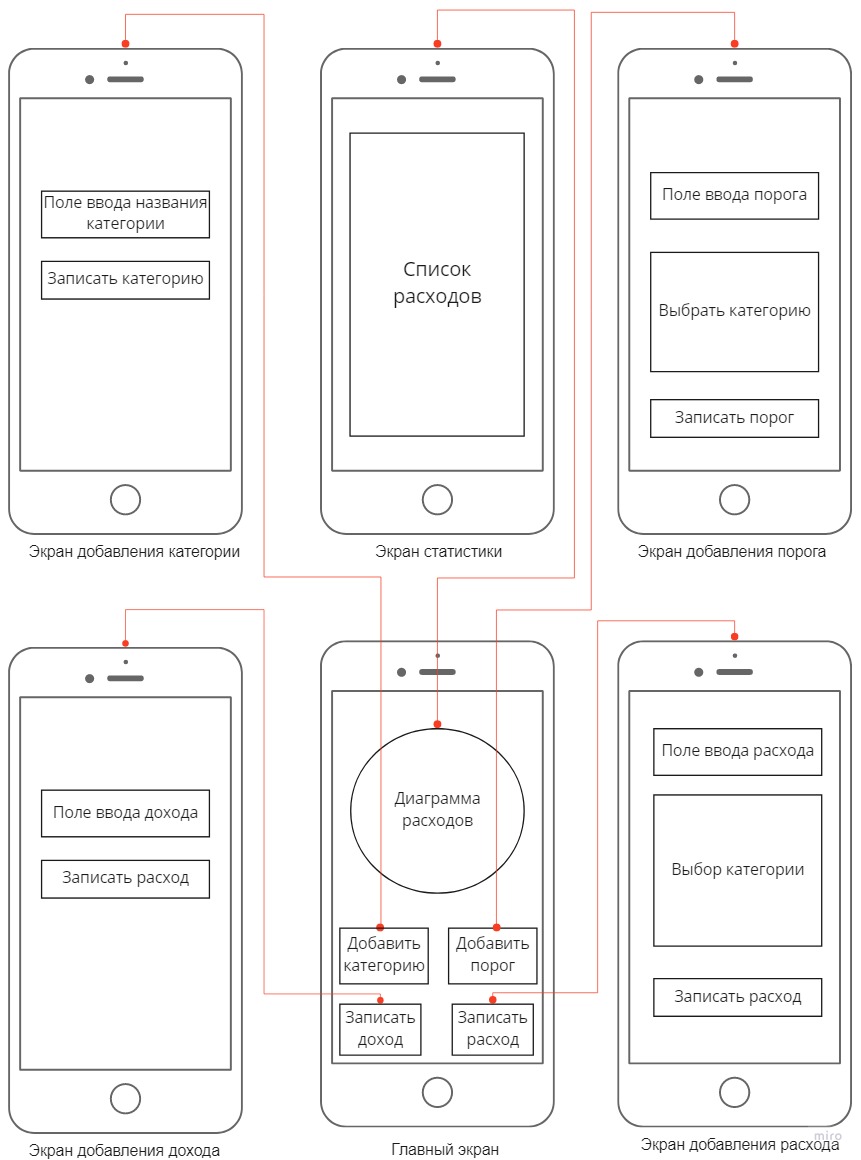


Рис. 2 – Дизайн приложения

Дизайн и процесс навигации в процессе разработки может изменяться.

## Требования к видам обеспечения

### Требования к программному обеспечению

Основные требования к программному обеспечению:

* Java
* SQLite

В результате разработки данные требования могут расширяться.

### Требования к лингвистическому обеспечению

Приложение должно поддерживать русский язык.

### Требования к техническому обеспечению

Приложение должно стабильно работать на большинстве современных мобильных устройств на базе Android 4.1 и выше. Точные технические характеристики приложения будут уточнены после завершения разработки.

# План тестирования

Для тестирования приложения необходимо выполнить 3 вида тестов:

* Дымовое тестирование

Необходимо проверить работоспособность приложение на следующих основных сценариях:

* + Добавление доходов
  + Удаление доходов
  + Добавление расходов
  + Удаление расходов
  + Добавление порога
  + Удаление порога
  + Отображение статистики
  + Добавление категории
  + Удаление категории
* UI тесты

Необходимо проверить работу всех интерфейсных элементов приложения, с которыми взаимодействует пользователь.

* Юзабилити тесты

Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с приложением:

* + Добавление доходов
  + Удаление доходов
  + Добавление расходов
  + Удаление расходов
  + Добавление порога
  + Удаление порога
  + Проверка статистики
  + Проверка работы порога
  + Добавление категории
  + Удаление категории

# ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Заказчик принимает разработанное приложение при выполнении следующих пунктов:

* Успешное прохождение всех дымовых тестов и большей части остальных тестов;
* Приложение соответствует заявленным требованиям.