МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра информационные технологии управления

Курсовая работа по курсу  
«Технологии программирования»

Мобильное приложение «Мои расходы»

Выполнили: студенты 3 курса, группы 3.2

Добрынина Е.А., Немчанинова Ю.Е.

Руководитель: Полещук Х. А.

Воронеж 2019

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc10128324)

[1. ПОСТАНОВКА ЗАДАЧИ 4](#_Toc10128325)

[2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc10128326)

[2.1. Глоссарий 5](#_Toc10128327)

[2.2. Анализ существующих решений 5](#_Toc10128328)

[2.2.1. Expense Manager 5](#_Toc10128329)

[2.2.2. CoinKeeper 6](#_Toc10128330)

[2.2.3. EasyMoney 6](#_Toc10128331)

[2.3. Анализ средств реализации 7](#_Toc10128332)

[2.4. UML диаграммы 8](#_Toc10128333)

[2.4.1. Диаграмма вариантов использования 8](#_Toc10128334)

[2.4.2. Диаграмма последовательности 9](#_Toc10128335)

[2.4.3. Диаграмма коммуникаций 11](#_Toc10128336)

[2.4.4. Диаграмма состояний 13](#_Toc10128337)

[2.4.5. Диаграмма классов 14](#_Toc10128338)

[2.4.6. Диаграмма развертывания 16](#_Toc10128339)

[2.4.7. Диаграмма объектов 17](#_Toc10128340)

[2.4.8. Диаграмма деятельности 17](#_Toc10128341)

[2.5. Схема базы данных 20](#_Toc10128342)

[3. ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОЙ СИСТЕМЕ 21](#_Toc10128343)

[4. КРИТЕРИИ ПРИЕМКИ 22](#_Toc10128344)

[5. ПЛАНИРОВАНИЕ РАБОТ 25](#_Toc10128345)

[5.1. Выбор методологии проектирования 25](#_Toc10128346)

[5.2. Виды работ, которые необходимо выполнить в процессе разработки программного средства 25](#_Toc10128347)

[5.3. Состав команды, распределение задач по участникам и по времени 26](#_Toc10128348)

[6. РЕАЛИЗАЦИЯ 28](#_Toc10128349)

[6.1. Использованные средства реализации 28](#_Toc10128350)

[7. ТЕСТИРОВАНИЕ 28](#_Toc10128351)

[8. ИТОГИ РАБОТЫ 28](#_Toc10128352)

[ЗАКЛЮЧЕНИЕ 29](#_Toc10128353)

[СПИСОК ЛИТЕРАТУРЫ 30](#_Toc10128354)

# ВВЕДЕНИЕ

В настоящее время каждому человеку приходится осуществлять большое количество финансовых операций. Поэтому становится очевидной необходимость ведения личной бухгалтерии. С ее помощью возможно увидеть уязвимости в личном бюджете, которые до этого на глаза не попадались. Но главное — благодаря личной бухгалтерии будет возможность найти простые способы для оптимизации вашего бюджета.

Существует множество способов ведения личной бухгалтерии, к примеру, записи в блокнот или онлайн-сервисы. Тем не менее, наиболее предпочтительным является мобильное приложение. В современном мире смартфон находится рядом с владельцем в течение всего дня, поэтому создавать новые записи о расходах, просматривать отчеты и анализ финансов становится очень быстро и удобно, все записи находятся «под рукой», нет привязки к компьютеру.

Но большинство существующих мобильных приложений не обладает интуитивно понятным интерфейсом, необходимыми пользователю функциями, не являются легковесными и удобными, а поэтому не предпочтительны. Кроме того, к сожалению, эргономичные, удобные, простые приложения, предоставляющие пользователю широкий спектр возможностей, являются платными, наполнены рекламой или предоставляют бесплатно весьма ограниченный функционал.

В данной курсовой работе рассматривается проблема создания простого, легковесного и не нагруженного излишней функциональностью, но обладающего всеми необходимыми функциями мобильного приложения «Мои расходы».

# ПОСТАНОВКА ЗАДАЧИ

Цель курсовой работы - реализовать клиент-серверное Android-приложение для отслеживания личных расходов.

Приложение должно позволять пользователю:

1. Создавать категории расходов

2. Добавлять критическую сумму расходов для каждой категории

3. Вносить информацию о расходах по категориям

4. Получать уведомления о превышении критической суммы

5. Получать отчет о расходах в виде круговой диаграммы

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 2.1. Глоссарий

*Доход* — 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.

*Цель* — намерение пользователя накопить определенную сумму денег до определенной даты.

*Категория* — группа товаров или услуг.

*Предел расходов* — намерение пользователя потратить сумму, не превышающую заданной.

*Чек* — фотография ценной бумаги, содержащей информацию о совершенной пользователем покупке.

## 2.2. Анализ существующих решений

Рассмотрим существующие решения.

### 2.2.1. Expense Manager

Приложений с подобным названием в данный момент в Google Play несколько, но именно приложение разработанное компанией Bishinews пользуется популярностью. Об этом можно судить по счетчику скачиваний в Google Play, который совсем недавно достиг отметки в 5 000 000 скачиваний.

Функционал приложения огромен, но самым главным является планировка своих финансов. Однако разработчики не наделили программу русским интерфейсом. При первом входе в приложение будет показан шаблон с несколькими пунктами расходов, которые при желании можно изменить, перейдя в настройки. Так же можно очень легко просматривать необходимые платежи или же взглянуть на подробный и наглядный график расходов.

Достоинства:

* много полезных функций;
* просмотр отчетов в виде графиков;

Недостатки:

* ограниченный функционал бесплатной версии;
* отсутствие русского языка.

### 2.2.2. CoinKeeper

Одно из самых популярных приложений для ведения учета расходов и доходов прямо на смартфоне или планшете на Android. После первого запуска приложение предложит выбрать наиболее подходящий способ планирования доходов: автоматический (в этом случае необходимо указать сумму ежемесячного дохода) или же ручной (более тонкая настройка каждого из параметров). Приложение распространяется в Google Play на условно-бесплатной основе, что означает бесплатное использование всего функционала в течение 15 дней, а в дальнейшем при желании продолжать работу с приложением необходимо приобрести полную версию.

Достоинства:

* любые операции можно провести перетягиванием одной иконки на другую;
* подсказки;
* автоматическое планирование бюджета;
* синхронизация с другими устройствами;
* возможность установки пароля.

Недостатки:

* невозможность представления отчета в виде графика;
* медленная анимация, которая тормозит даже на современных устройствах.

### 2.2.3. EasyMoney

Дорогое и самое функциональное приложение из данной категории приложений. Но, судя по количеству скачиваний в Google Play, пользователей цена приложения вовсе не останавливает, тем более что есть бесплатная версия программы и можно протестировать начальный функционал на ней, а уже потом решить стоит ли тратиться на данную программу или же нет.

Достоинства:

* огромный функционал;
* наглядные отчеты в виде интерактивных графиков;
* наличие удобного виджета.

Недостатки:

* не совсем понятный интерфейс;
* отсутствие русского языка;
* высокая стоимость за полную версию программы.

## 2.3. Анализ средств реализации

Ниже представлены технологии, которые используются для реализации клиент-серверного мобильного приложения для отслеживания личных расходов:

* Android SDK - набор компонентов и средств, необходимых для разработки приложений на Android, обладающий широкими функциональными возможностями, позволяющими запускать тестирование и отладку исходных кодов, оценивать работу приложения в режиме совместимости с различными версиями ОС Android и наблюдать результат в реальном времени.
* Язык реализации клиентской части приложения - Java.
* Серверная часть приложения реализована на языке Python с использованием инструмента Flask - фреймворк, имеющий встроенный веб-сервер.
* В качестве СУБД выбрана SQLite.
* ORM для взаимодействия с базой данных – Peewee.

## 2.4. UML диаграммы

### 2.4.1. Диаграмма вариантов использования



Рис. 1. Диаграмма вариантов использования.

Рассмотрим диаграмму вариантов использования для приложения на рис. 1.

Актер – пользователь приложения.

Отношения ассоциации - действия, которые может осуществлять пользователь:

* добавить расход;
* редактировать расход;
* удалить расход;
* добавить категорию;
* редактировать категорию;
* редактировать порог;
* удалить категорию;
* получить отчет.

Отношение включения: установить порог категории при ее добавлении.

### 2.4.2. Диаграмма последовательности

* Диаграмма последовательности для запуска приложения, рис. 2:

Рис. 2. Диаграмма последовательности для запуска приложения.

При запуске приложение делает запрос на сервер, который в свою очередь делает запрос в БД на данные по расходам. Когда БД возвращает список всех расходов, сервер посылает ответ в формате json. Затем на начальном экране отображается список всех расходов, содержащий дату, категорию, сумму и комментарий для каждого расхода.

* Диаграмма последовательности для добавления категории, рис. 3:

Рис. 3. Диаграмма последовательности для добавления категории.

Для добавления новой категории пользователю необходимо на панели управления перейти на вкладку «Категории» и нажать кнопку «+». После этого отображается окно для ввода названия категории и ежемесячного порога. Пользователю необходимо ввести данные и нажать кнопку «Сохранить». В случае успешно введенных данных, приложение делает запрос на сервер, который обращается в БД с запросом на сохранение. Когда данные сохранятся, сервер посылает ответ приложению. Затем пользователю отображается обновленный список категорий.

* Диаграмма последовательности для добавления расхода, рис. 4:

Рис. 4. Диаграмма последовательности для добавления расхода.

Для добавления новой категории пользователю необходимо на главном экране нажать кнопку «+». После этого отображается окно для ввода данных о расходе. Пользователю необходимо ввести данные и нажать кнопку «Сохранить». В случае успешно введенных данных, приложение делает запрос на сервер, который обращается в БД с запросом на сохранение. Когда данные сохранятся, сервер посылает ответ приложению. Затем пользователю отображается обновленный список расходов.

### 2.4.3. Диаграмма коммуникаций

* Диаграмма коммуникации для запуска приложения, рис. 5:

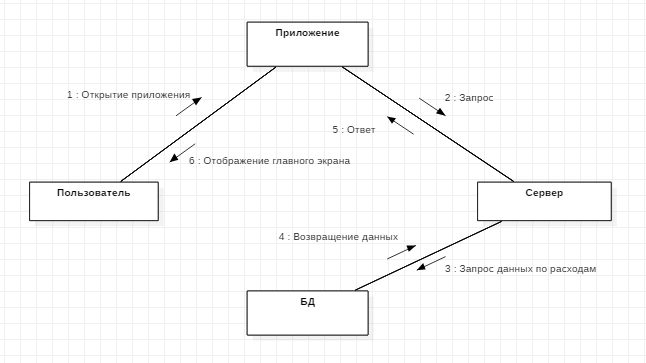


Рис. 5. Диаграмма коммуникации для запуска приложения.

* Диаграмма коммуникации для добавления категории, рис. 6:

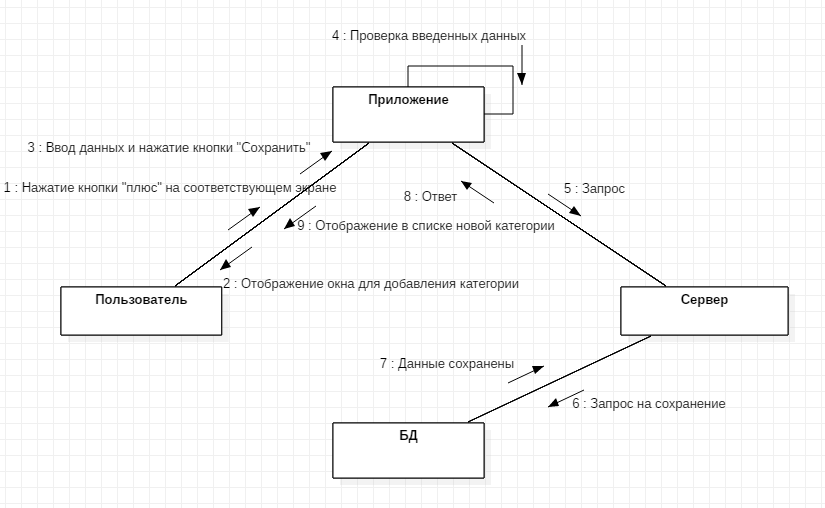


Рис. 6. Диаграмма коммуникации для добавления категории.

* Диаграмма коммуникации для добавления расхода, рис. 7:

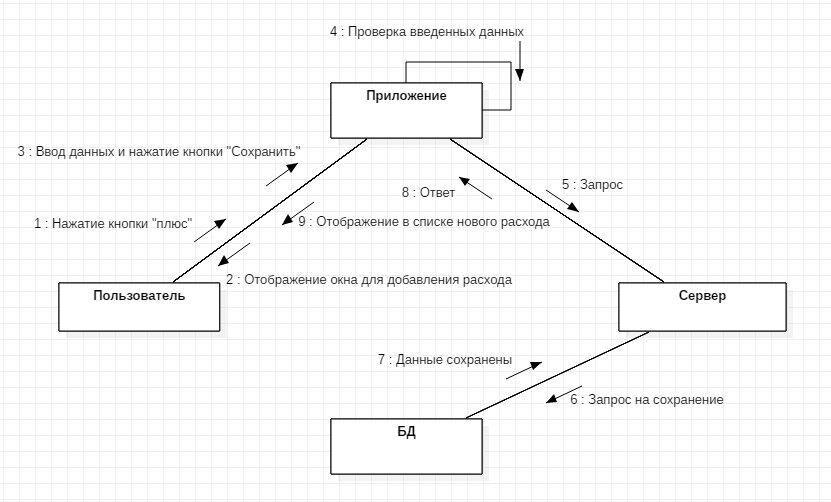


Рис. 7. Диаграмма коммуникации для добавления расхода.

Сообщения в диаграмме коммуникаций такие же, что и в диаграмме последовательности, поэтому их повторное описание не приводится.

### 2.4.4. Диаграмма состояний

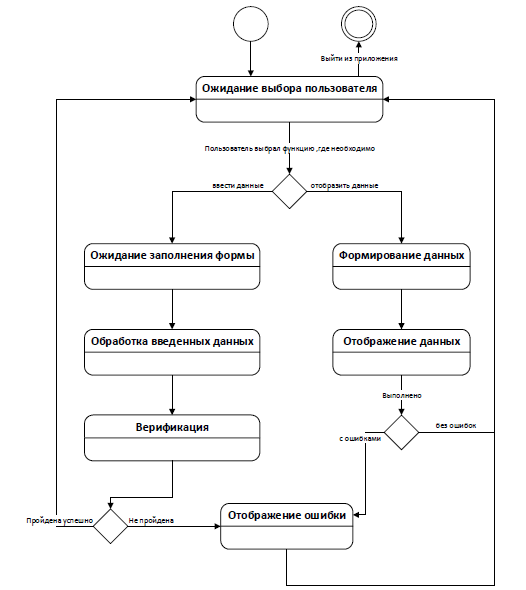


Рис. 8. Диаграмма состояний.

Диаграмма состояний отражает возможные состояния системы. При входе в приложение система находится в ожидании выбора пользователя. В зависимости от выбора пользователя возможны 2 основные цепочки состояний, связанных с вводом данных и с их отображением. Если пользователя выбирает функцию, где необходимо вводить данные, система переходит в состоянии ожидания заполнения формы, после ввода данных пользователем система обрабатывает введенные данные, далее верифицирует эти данных. Если верификация пройдена успешно, система переходит в состояние ожидание выбора пользователя, иначе в состояние отображение ошибки. Другая цепочка событий связанна с отображение данных, которые запросил пользователь. При этом система сначала формирует данных, затем отображает. В случае корректного выполнения система переходит в состояние ожидание выбора пользователя, иначе в состояние отображение ошибки.

### 2.4.5. Диаграмма классов

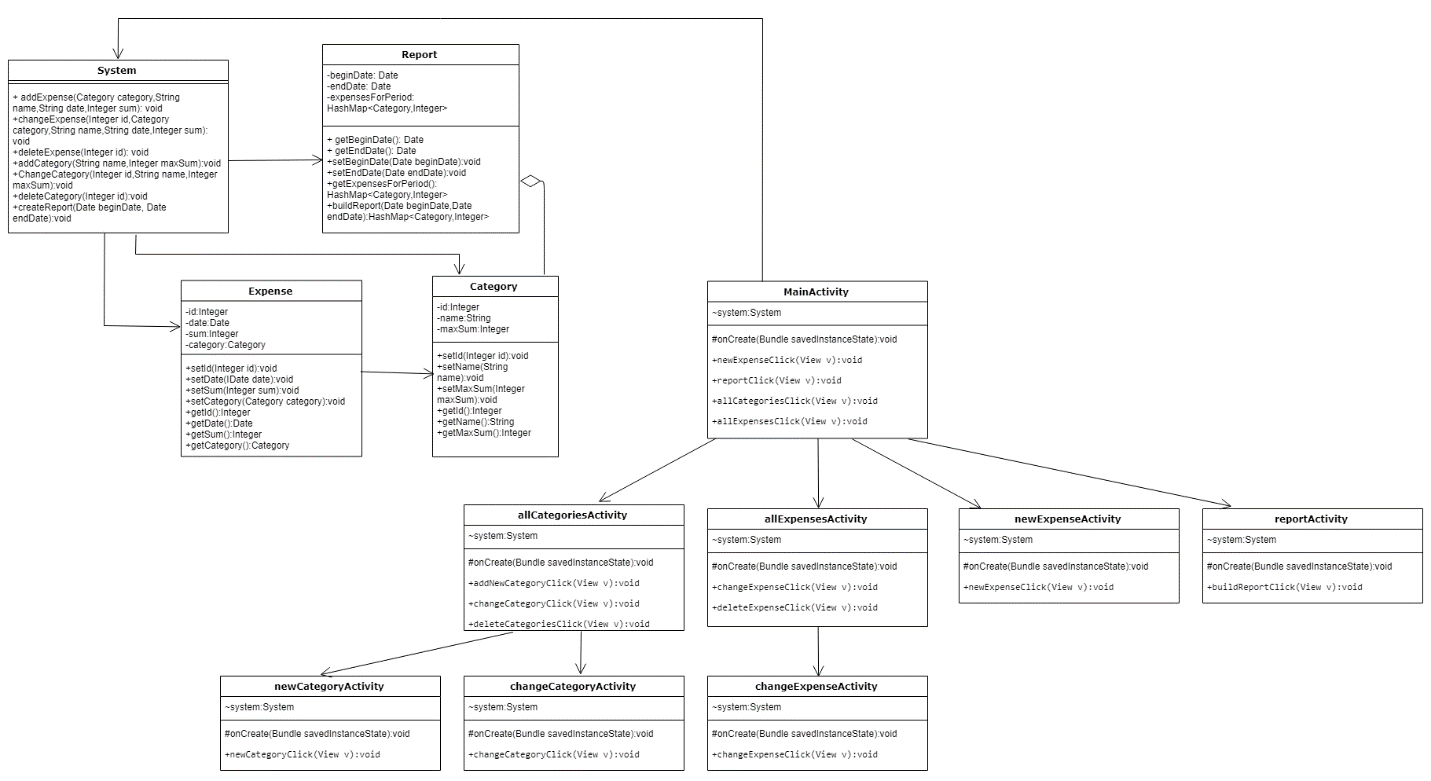


Рис. 9. Диаграмма классов.

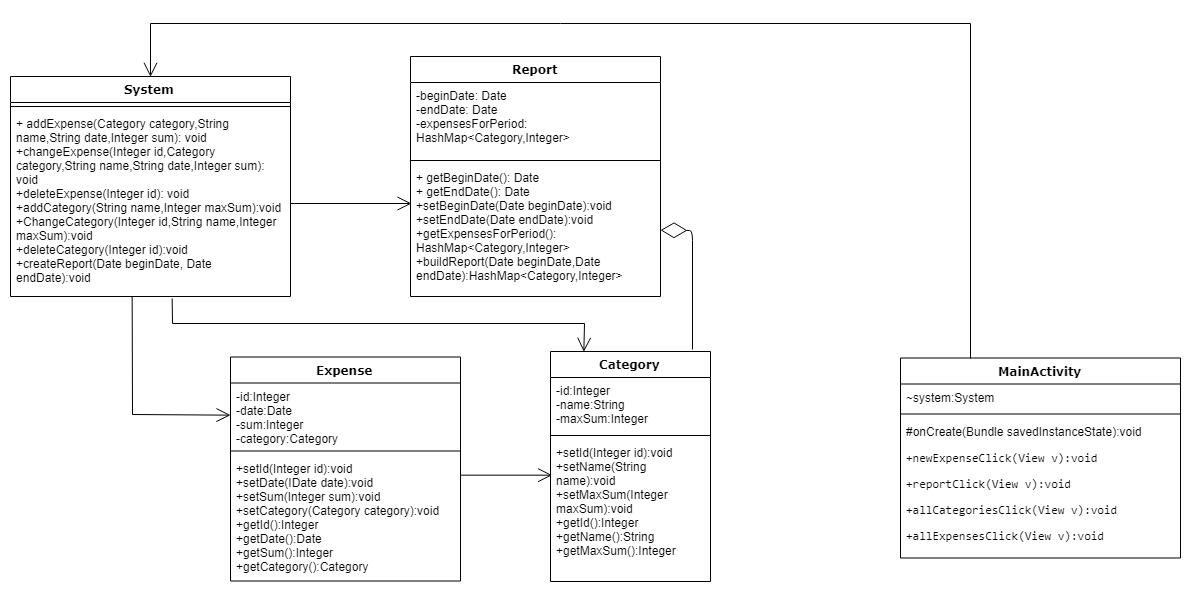


Рис. 10. Диаграмма классов(основные классы).

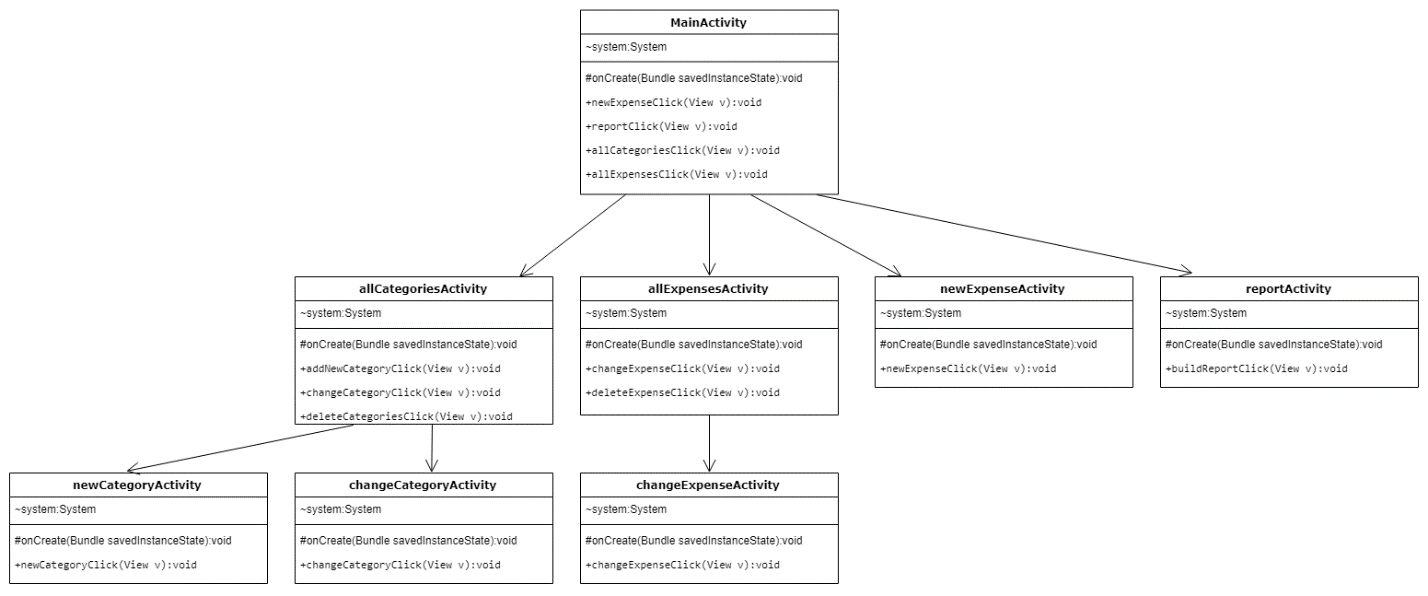


Рис. 11. Диаграмма классов(классы экранов приложения).

Класс Category соответствует категории расходов.

Класс Report отвечает за создание отчетов по расходам.

Класс Expense соответствует записи расхода пользователя.

Класс System отвечает за работу с категориями, рассходами и отчетами. Этот класс имеет методы для создания, изменения и удаления расходов и категорий ,а так же метод для создания отчета.

Класс MainActivity отвечает за главный экран приложения. На главном экране есть кнопки, а в классе методы для событий нажатия на эти кнопки: “Добавить расход”- newExpenseClick(), “Все расходы”-allExpensesClick(), “Все категории”- allCategoryClick(), “Отчет ”- reportClick().

Класс allCategoryActivity отвечает за экран, на котором отображается список все категорий.

Класс allExpensesActivity отвечает за экран, на котором отображается список все расходов.

Класс newExpensesActivity отвечает за экран, на котором происходит ввод данных для создания нового расхода.

Класс reportActivity отвечает за экран, на котором происходит создание отчета по расходам.

Класс newCategoryActivity отвечает за экран, на котором происходит ввод данных для создания новой категории.

Класс changeCategoryActivity отвечает за экран, на котором происходит изменение уже существующей категории.

Класс changeExpenseActivity отвечает за экран, на котором происходит изменение уже существующей записи о расходе.

### 2.4.6. Диаграмма развертывания

На рис. 12 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты работают на каждом узле и как различные части этого комплекса соединяются друг с другом.

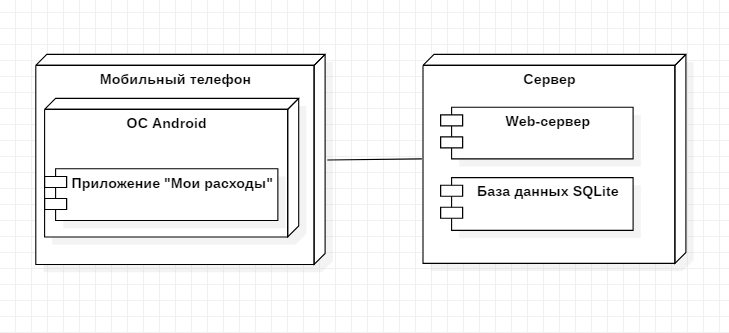


Рис. 12. Диаграмма развертывания.

### 2.4.7. Диаграмма объектов

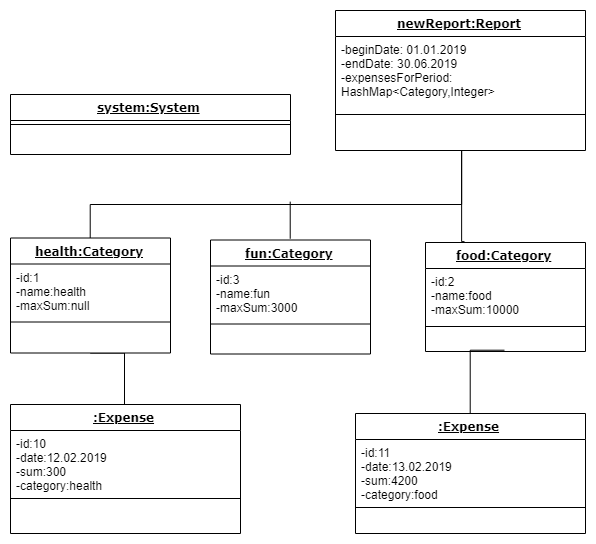


Рис. 13. Диаграмма объектов.

В момент времени после генерации отчета система может выглядеть следующим образом: в системе имеется три экземпляра класса Category (food-еда, health- здоровье, fun-развлечения) и 2 экземпляра класс Expense без названия, которые взаимодействуют с экземплярами категории. Экземпляр класс Report знает о всех объектах класса Category и использует их для создания хэш-таблицы expensesForPeriod.

### 2.4.8. Диаграмма деятельности

* Диаграмма деятельности для варианта использования “Добавить расход”



Рис. 13.Диаграмма деятельности для варианта использования “Добавить расход”

Чтобы добавить расход необходимо на главном экране нажать кнопку

“Добавить расход”, далее заполнить поля (дата, категория, сумма) и нажать кнопку “Ок”. Если данные введены корректно и не возникло ошибок в системе пользователь переходит на главный экран. В случае ошибок пользователь должен просмотреть ошибку, после этого может ввести данные заново или вернуться на главный экран.

* Диаграмма деятельности для варианта использования “Получить отчёт”



Рис. 14.Диаграмма деятельности для варианта использования “ Получить отчёт”

Чтобы получить отчёт необходимо на главном экране нажать кнопку

“Отчёт”, далее заполнить поля (дата начала и дата конца периода) и нажать кнопку “Ок”. Если данные введены корректно и не возникло ошибок в системе отображается отчёт на этом же экране, и пользователь может просмотреть этот отчёт. В случае ошибок пользователь должен просмотреть ошибку, после этого может ввести данные заново или вернуться на главный экран.

## 2.5. Схема базы данных



Рис. 15. Схема базы данных

# ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОЙ СИСТЕМЕ

1. Пользователь может добавлять запись о расходах с помощью ввода суммы расхода и даты и выбора категории из существующих в базе данных. При корректном вводе всех данных запись о расходах добавляется в базу данных.
2. Если после пересчета суммы по категории эта сумма превышает заданный пользователем лимит, система вывод уведомление о превышении лимита.
3. Пользователь может просматривать все записи с помощью клика на советующую кнопку в главном меню.
4. Пользователь может изменять существующие записи о расходах, выбрав необходимую запись в общем списке записей. Он может изменить дату, сумму или категорию расхода.
5. Пользователь может удалить существующую запись. Для этого необходимо перейти в общий список записей , выбрать необходимую запись, где он может нажать кнопку удаления.
6. Пользователь может добавлять новые категории расходов. Ему нужно указать название категории и по желанию критическую сумму расходов по этой категории.
7. Пользователь может редактировать категории. Можно изменить название категории ,добавить лимит или изменить лимит.
8. Пользователь может удалить категорию. При этом все записи расходов для этой категории удаляются.
9. Пользователь может получать сводный отчет: список категорий с общей суммой расходов по каждой категории для выбранного периода времени.

# КРИТЕРИИ ПРИЕМКИ

1. Главный экран(первый при входе в приложение) содержит:
   1. Таблицу с названием категорий, суммой расходов за текущий месяц и критической суммой для каждой категории.
   2. Кнопку «Категории» – для просмотра всех созданных категорий.
   3. Кнопку «Отчет» – для перехода на экран для генерации сводного отчета.
   4. Кнопку «Все расходы» – для просмотра всех созданных записей расходов.
   5. Кнопку «Добавить расход» – для перехода на экран для добавления нового расхода.
2. При нажатии кнопки «Добавить расход» пользователь переходит на экран «Добавление новой записи расхода». Экран «Добавление новой записи расхода» содержит:
   1. Поле для ввода даты. Это поле является обязательным.
   2. Текстовое поле для ввода суммы расходов. В это поле могут быть введены только положительные числа. Это поле является обязательным.
   3. Список категорий ,из которого пользователь может выбрать только одну категорию.
   4. Кнопку для подтверждения введенных данных.
      1. После нажатия на кнопку система верифицирует введенные пользователем данные. Если эти данные корректны (введена положительная сумма, выбрана категория), система проводит новую запись и перенаправляет пользователя на главный экран. Если данные не корректны – выдать сообщение об ошибке.
3. При нажатии кнопки «Категории» пользователь переходит на экран «Список категорий». Экран «Список категорий» содержит:
   1. Список всех созданных категорий.
      1. При нажатии на строку пользователь переходит на экран «Редактирование категории».
      2. Каждая строка содержит кнопку для удаления категории.
   2. Кнопку для добавления новой категории. При нажатии на эту кнопку пользователь переходит на экран «Добавление новой категории».
4. Экран «Добавление новой категории» содержит:
   1. Текстовое поле для ввода названия категории. Это поле обязательное. Длина не более 30 символов.
   2. Текстовое поле для ввода критической суммы расходов по категории. Это поле не является обязательным. В поле могут быть введены только неотрицательные числа.
   3. Кнопку для подтверждения введенных данных.
      1. Если введенные данные являются корректными, система добавляет новую категорию и перенаправляет пользователя на главный экран. Если данные не корректны – выдать сообщение об ошибке.
5. Экран «Редактирование категории» содержит те же поля и ограничения что и экран «Добавление новой категории».
6. Экран «Все расходы» содержит список всех записей расходов ,которые создал пользователь.
   1. Каждая строка содержит сумму расхода, категорию, дату создания записи.
   2. Пользователь может удалить или изменить запись, нажав на соответствующую кнопку для каждой строки.
7. Экран «Редактирование записи» содержит те же поля и ограничения что и экран «Добавление новой записи расхода».
8. Экран «Отчет» содержит:
   1. Поле для ввода даты начала периода и даты конца периода, за который будет формироваться отчет.
   2. Кнопку для подтверждения выбранного периода. После нажатия пользователю выводиться отчет в виде списка. Каждая строка списка содержит категорию и общую сумму расхода для каждой категории.
9. Каждый экран содержит кнопку для перехода на главный экран.

# 5. ПЛАНИРОВАНИЕ РАБОТ

## 5.1. Выбор методологии проектирования

Для проекта выбрана методология Agile. Главная причина отказа от классического Waterfall является то, что в этой методологии предполагает последовательный переход к каждому этапу разработки и невозможностью вернуться на шаг назад. Внести какие-либо изменения будет возможно только после релиза проекта.

Преимущества методологии Agile:

* 1. Внесение необходимых изменений и внедрение нового функционала может происходить независимо от цикла разработки продукта.
  2. Проект состоит из коротких и понятных циклов, по окончании которых создается рабочая версия продукта. Каждый цикл представляет собой завершенный мини-проект, в котором есть этапы анализа, планирования, тестирования и реализации. В итоге клиент получает продукт, который, при необходимости, корректируется.
  3. Гибкий процесс корректировок в любой итерации позволяет снизить производственные риски. - Довольно быстрый релиз пробной версии для дальнейших корректировок и тестирования.

## 5.2. Виды работ, которые необходимо выполнить в процессе разработки программного средства

Разработка приложения должна быть проведена в два этапа:

* 1. разработка документации;
  2. разработка приложения.

Рассмотрим подробнее каждый из них.

На этапе разработки документации должны быть выполнены следующие работы:

* написание введения;
* определение актуальности задачи;
* составление анализа аналогов;
* выбор методологии разработки;
* определение требований;
* создание диаграммы вариантов использования;
* создание диаграммы коммуникаций;
* создание диаграммы состояний;
* создание диаграммы последовательности;
* создание диаграммы развертывания;
* создание диаграммы деятельности;
* создание диаграммы объектов;
* написание глоссария проекта;
* написание ТЗ;
* создание функциональной схемы;
* создание прототипов экранов;
* создание диаграммы классов.

На этапе разработки приложения должна быть выполнены следующие виды работ:

● создание репозитория в системе контроля версия GitHub;

● установка и настройка необходимых инструментальных средств;

● проектирование и создание БД;

● реализация классов приложения;

● создание интерфейсной части приложения;

● создание логической части приложения.

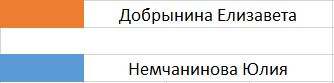
## 5.3. Состав команды, распределение задач по участникам и по времени

В состав команды разработчиков вошли:

1. Добрынина Елизавета;

2. Немчанинова Юлия.

Ниже представлена диаграмма Ганта для выполненных работ на два этапа, описанных в пункте 5.2.



# 6. РЕАЛИЗАЦИЯ

## 6.1. Использованные средства реализации

# 7. ТЕСТИРОВАНИЕ

# 8. ИТОГИ РАБОТЫ

# ЗАКЛЮЧЕНИЕ

# СПИСОК ЛИТЕРАТУРЫ