

**МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕ-
ЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ “ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ”**

**Факультет компьютерных наук
Кафедра технологий обработки и защиты информации**

Веб-приложение по учету персональных расходов “Smart Expenses”

Курсовой проект

**09.03.02 Информационные системы и технологии
Обработка информации и машинное обучение**

Допущен к защите

Обучающийся _____ С.И. Гайворонский, 3 курс, д/о

Обучающийся _____ З.С. Гостев, 3 курс, д/о

Обучающийся _____ А.М. Гудимов, 3 курс, д/о

Руководитель _____ В.С. Тарасов, преподаватель

Воронеж 2020

Оглавление

Введение	3
1. Постановка задачи.....	4
2. Анализ предметной области	5
2.1. Глоссарий.....	5
2.2. Анализ существующих решений.....	5
2.2.1. CoinKeeper	5
2.2.2. Дребеденьги.....	6
2.2.3. Дзен-мани.....	7
2.3. Анализ поставленной задачи	9
2.3.1. Use-case диаграмма	9
2.3.2. Диаграмма классов и диаграмма объектов.....	11
2.3.3. Диаграмма объектов.....	13
2.3.4. Диаграмма последовательностей и диаграмма коммуникаций	14
2.3.5. Диаграмма состояний.....	20
2.3.6. Диаграмма активностей.....	23
2.3.7. Диаграмма развертывания.....	29
2.4. Обоснование архитектуры	30
2.5. Анализ средств реализации.....	30
3. Реализация	32
3.1. Серверная часть приложения	32
3.2. Графический интерфейс	32
3.2.1. Главная страница	32
3.2.2. Страница Регистрации	33
3.2.3. Страница Авторизации.....	33
3.2.4. Страница «Категории»	34
3.2.5. Страница «Операции».....	34
3.2.6. Страница «Обзор»	35
4. Тестирование	36
4.1. Smoke testing.....	36
4.2. End-to-end testing	38
4.3. Usability-testing	39
5. Статистика.....	41
Заключение	43

Введение

У каждого человека свои цели и видение будущего, свой бюджет. Вместе с тем, в жизни подавляющего большинства людей есть схожие финансовые цели. Для многих это – покупка недвижимости, создание фондов для высшего обучения детей, повышение уровня жизни, формирование личного капитала, создание и передача наследства.

Все эти задачи для своего решения требуют крупных сбережений. А для создания необходимых сбережений нужно контролировать свои расходы.

Однако в современном мире свободного времени у людей становится все меньше и меньше. Не каждый может позволить себе скрупулезно вести учет всех своих затрат в бумажном ежедневнике или в заметках на смартфоне. К тому же самостоятельно систематизировать такие большие объемы данных не выглядит хорошим решением. Возникает вопрос: какие средства могут облегчить планирование расходов?

Желаемое средство должно: помочь пользователю:

- помочь пользователю регистрировать расходы;
- помочь пользователю классифицировать расходы;
- предоставить пользователю некоторые статистические данные о его расходах.

Также интерфейс не должен быть перегружен и предоставлять только необходимые функции

Разработке именно такого приложения посвящен данный курсовой проект.

1. Постановка задачи

Целью данной курсовой работы является создание веб-приложения, которое позволит автоматизировать контроль расходов.

Процессы, подлежащие автоматизации:

- Процесс регистрации расходов
- Процесс классификации расходов
- Процесс формирования ограниченного набора статистических данных о расходах
- Процесс внесения расходов

Интерфейс приложения должен предоставлять пользователю ограниченный набор необходимых функций и не быть перегруженным лишней функциональностью.

Задачи, необходимые для достижения выше поставленной цели:

- Изучить предметную область
- Проанализировать существующие проекты на данную тематику
- Определить и описать необходимые для создания приложения функции и модули
- Разработать приложение, отвечающее указанным требованиям
 - Наличие модуля регистрации пользователя
 - Наличие модуля “Работа с категориями”
 - Наличие модуля “Работа с операциями”
 - Наличие модуля “Работа со статистическими данными по расходам”

2. Анализ предметной области

2.1. Глоссарий

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

Категория - описательная характеристика, предназначенная для структуризации расходов пользователя.

Операция - структурная единица действий пользователя по добавлению расходов.

2.2. Анализ существующих решений

2.2.1. CoinKeeper

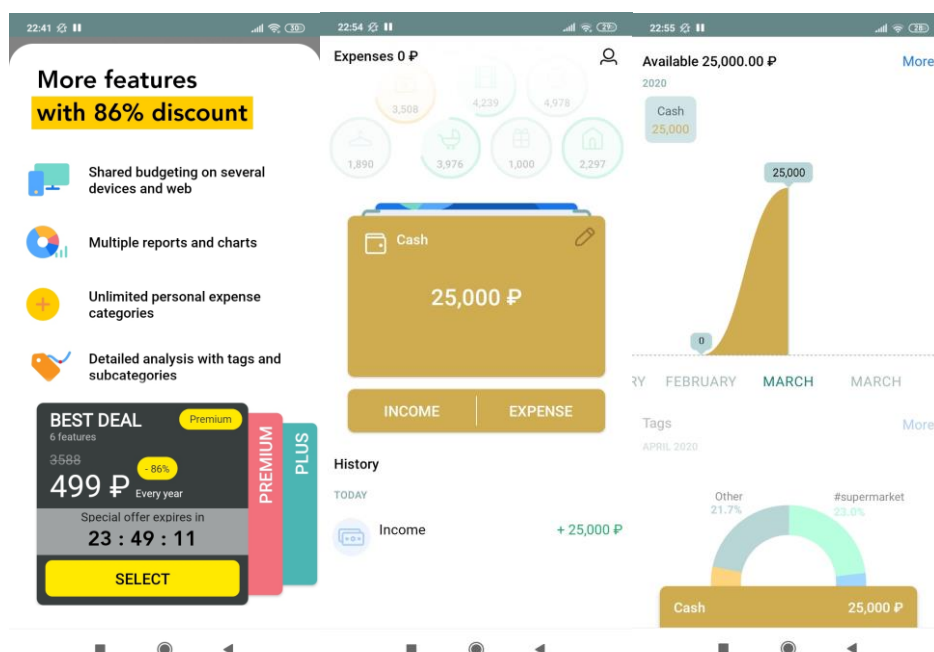
Является одним из лучших и популярных на рынке мобильных приложений по учету расходов. CoinKeeper распространяется в GooglePlay и AppStore на бесплатной основе. При первом запуске предлагает пользователю зарегистрироваться и приобрести платную версию приложения с расширенным функционалом или же использовать пробный период приложения, по истечении которого пользователю придется приобрести платную версию при желании продолжить пользоваться приложением.

Достоинства

- Наличие больших функциональных возможностей
- Возможность приобрести банковскую карту компании и связать её с приложением
- Автоматическая распланировка бюджета
- Синхронизация с другими устройствами
- Возможность прогнозирования бюджета

Недостатки

- Отсутствие бесплатной версии приложения
- Перегруженный интерфейс
- Навязывание акций компании



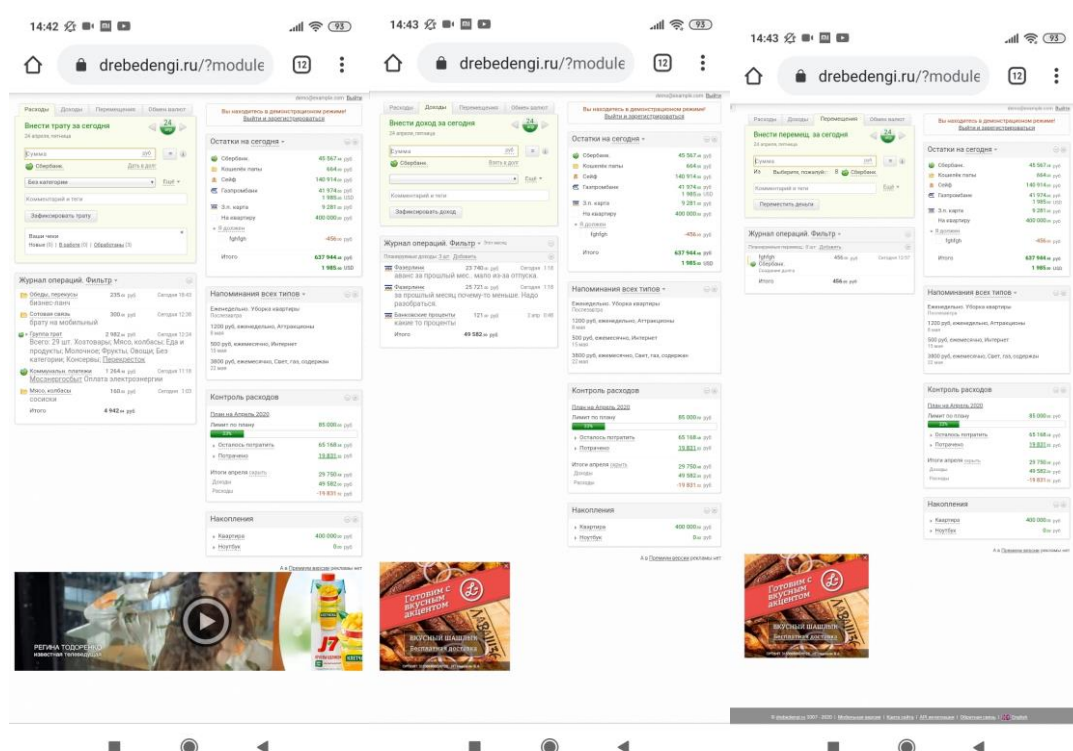
2.2.2. Дребеденьги

Дребеденьги — это онлайн-сервис, состоящий из мобильного приложения (Android, iOS) и браузерной онлайн-версии, для учёта личных и домашних финансов, представляющий собой домашнюю бухгалтерию. В браузерной онлайн-версии находится основной функционал приложения - создание категорий затрат, доходов и мест хранения, планирование трат, вся аналитика и так далее. Мобильное приложение выполняет функцию автоматического отмечания трат, то есть считывает пуш-сообщения смартфона и распознает их, если траты были совершены по безналичной оплате, в противном случае траты вводятся вручную. Пользование сайтом и основным функционалом - доступно бесплатно с ограниченным функционалом. Основной пакет функций доступен в случае годовой подписки.

Достоинства:

- Наличие необходимых функций;
- Наличие мобильной версии и браузерной онлайн-версии
- Автовнесение данных
- Синхронизация с другими устройствами;

- Гибкая система категорий (имеет несколько уровней);
- Недостатки:
- Отсутствует редактирование категорий
 - Отсутствует возможность просмотра отчетов
 - Отсутствует настройка правил для пуш-сообщений
 - Отсутствует возможность визуализирования расходов (графики, диаграммы)



2.2.3. Дзен-мани

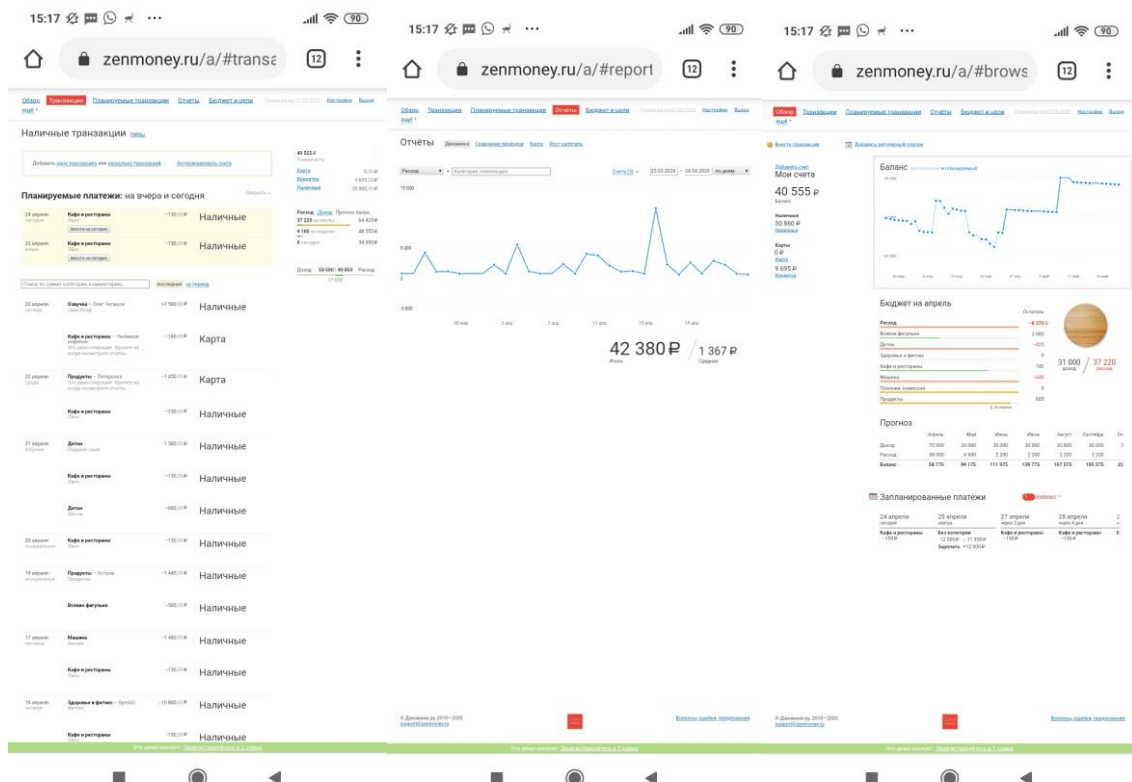
Дзен-мани - веб-сервис для учета личных финансов и планирования бюджета. Так же как и “Дребеденьги” обладает браузерной онлайн-версией и мобильным приложением, функцией распознавания банковских СМС и создания операций, что упрощает учет ведения расходов. У каждого пользователя есть пробный двухнедельный период, а в дальнейшем, при желании продолжить работу с данным сервисом, необходимо приобрести подписку.

Достоинства:

- Наличие основных функций
- Автовношение данных
- Добавление повторяющихся операций по “одному нажатию”
- Возможность синхронизации данных

Недостатки:

- Нестабильная работа мобильной версии приложения
- Невозможность группировать содержимое одной покупки
- Отсутствие бесплатной версии приложения
- Отсутствует возможность формирования отчетов по расходам и доходам независимо друг от друга
- Невозможность редактирования расходов



2.3. Анализ поставленной задачи

2.3.1. Use-case диаграмма



При работе с приложением Smart Expenses неавторизованный пользователь, именуемый далее гостем, имеет следующие возможности:

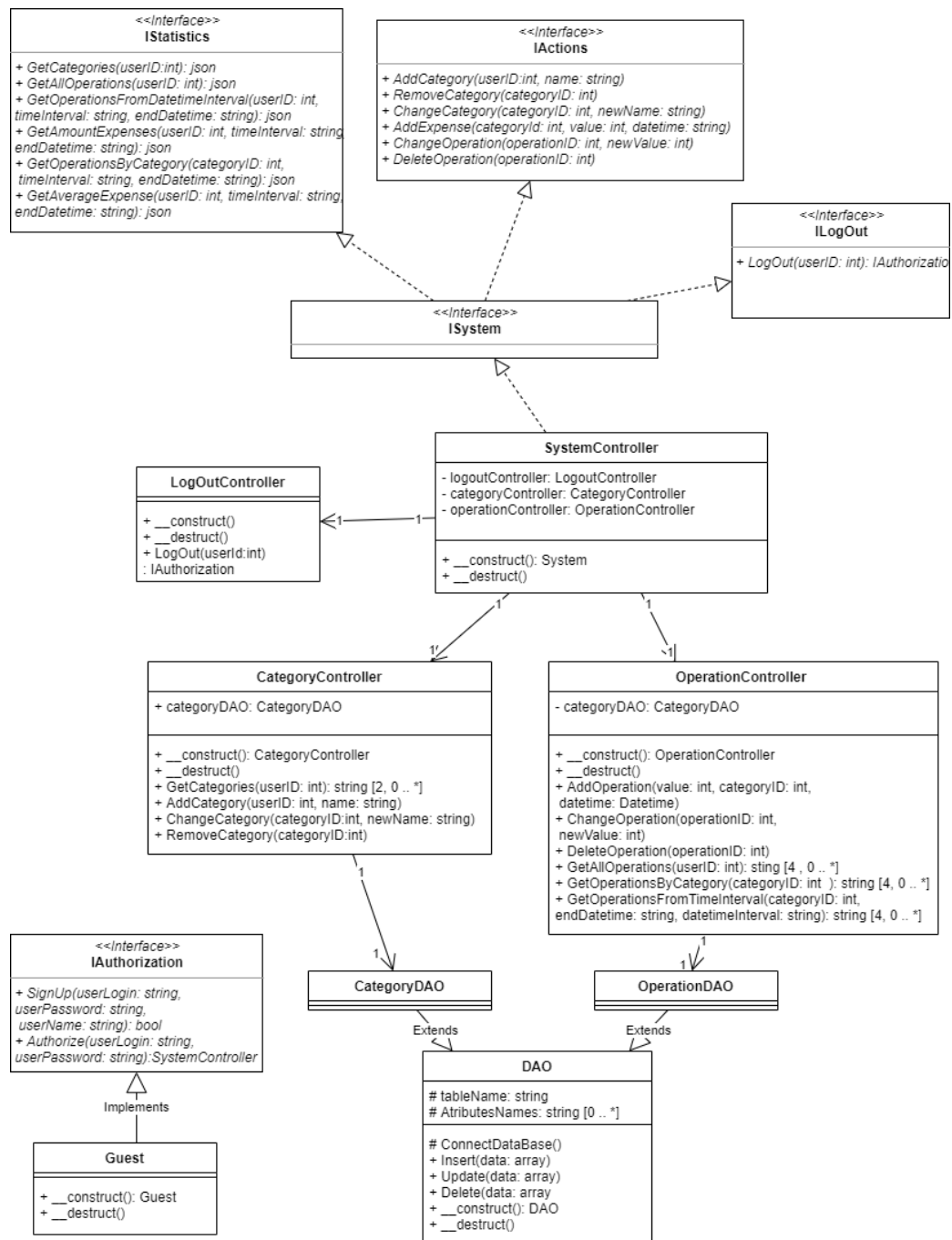
- Просмотр гостевой страницы
- Зарегистрироваться
- Войти в систему

В свою очередь, авторизованный пользователь данного приложения, то есть пользователь, который зарегистрировался и вошел в систему под своими учетными данными, обладает приведенными ниже возможностями:

- Обработать категорию расходов

- Редактировать категорию расходов
- Удалить категорию расходов
- Добавить категорию расходов
- Обработать операцию
 - Редактировать операцию
 - Удалить операцию
 - Добавить расход по категории
- Получить статистику расходов
 - Получить сумму расходов за временной промежуток (общая сумма расходов за день, неделю и средняя сумма расходов за день)
 - Получить диаграмму расходов по категориям (с возможностью сортировки по категориям и за временной промежуток: день, неделя, месяц)
 - Получить историю операций (с возможностью сортировки по категориям и за временной промежуток: день, неделя, месяц)
 - Получить статистику расходов по категориям (с возможностью сортировки за временной промежуток: день, неделя, месяц)
 - Получить процентное соотношение расходов по категориям (с возможностью сортировки за временной промежуток: день, неделя, месяц)
- Выйти из системы

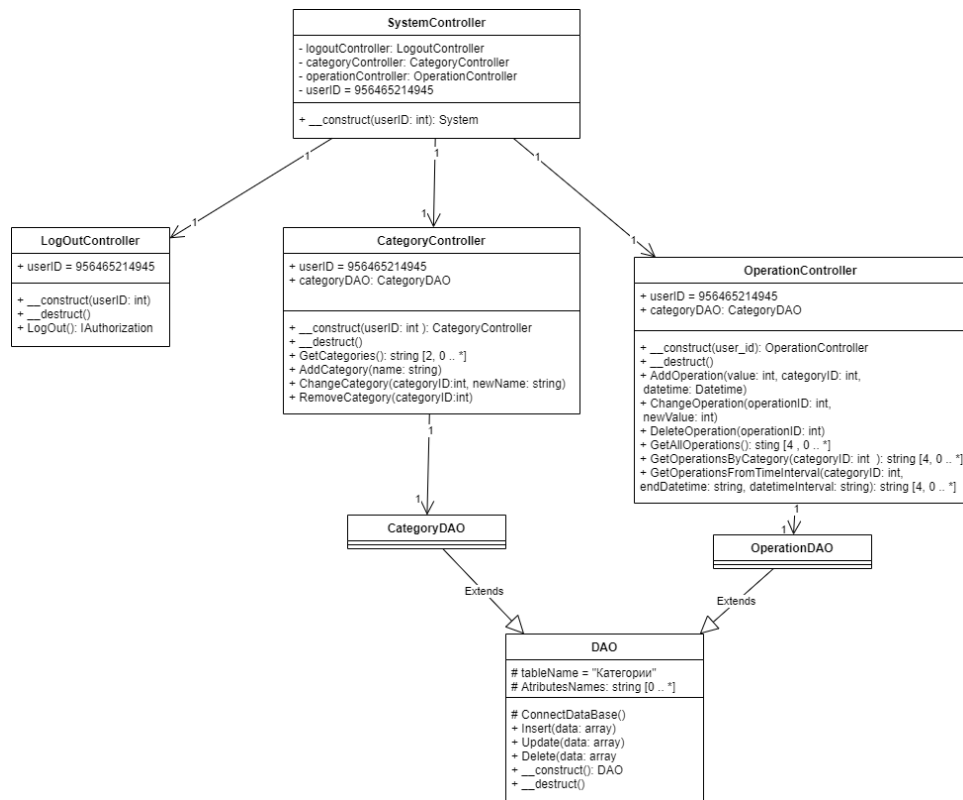
2.3.2. Диаграмма классов и диаграмма объектов



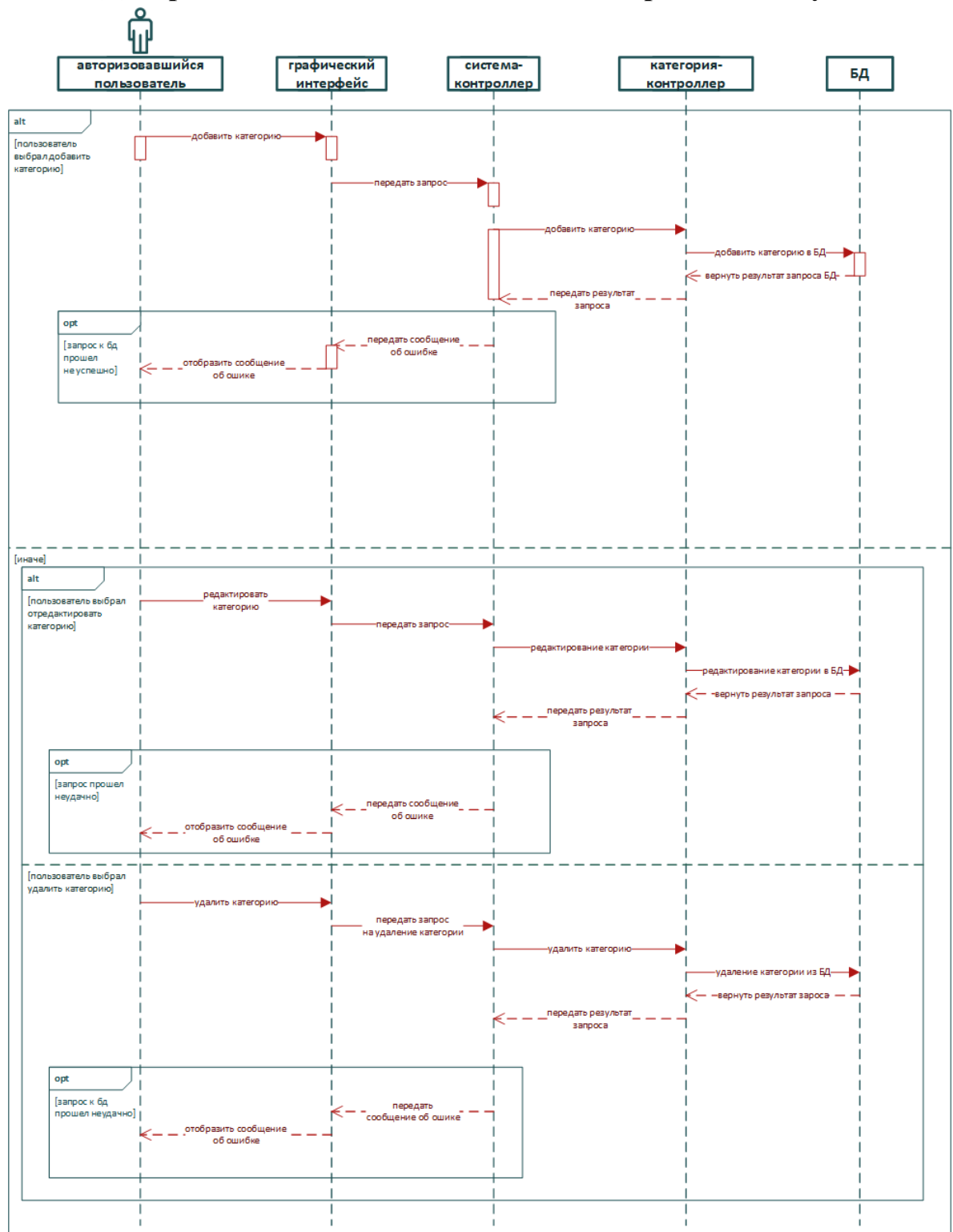
Серверная часть приложения организована следующим образом: классы CategoryDAO и OperationDAO обеспечивают получение и запись данных в БД. SystemController является основным классом системы, через который осуществляется взаимодействие с сервером. SystemController содержит в себе экземпляры классов CategoryController и OperationController, которые реализуют основные

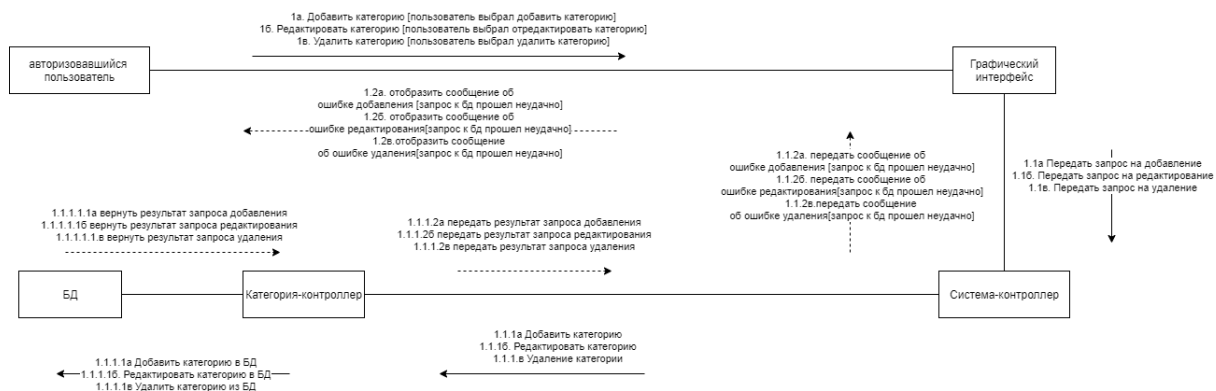
методы получения данных, которые запрашивает клиент. `SystemController` реализует главный интерфейс `ISystem`, который, в свою очередь, реализует интерфейсы `IActions` и `IStatistics`. В `IActions` и `IStatistics` описаны сигнатуры основных методов, которые предоставляет API. Таким образом реализуется паттерн «Фасад».

2.3.3. Диаграмма объектов



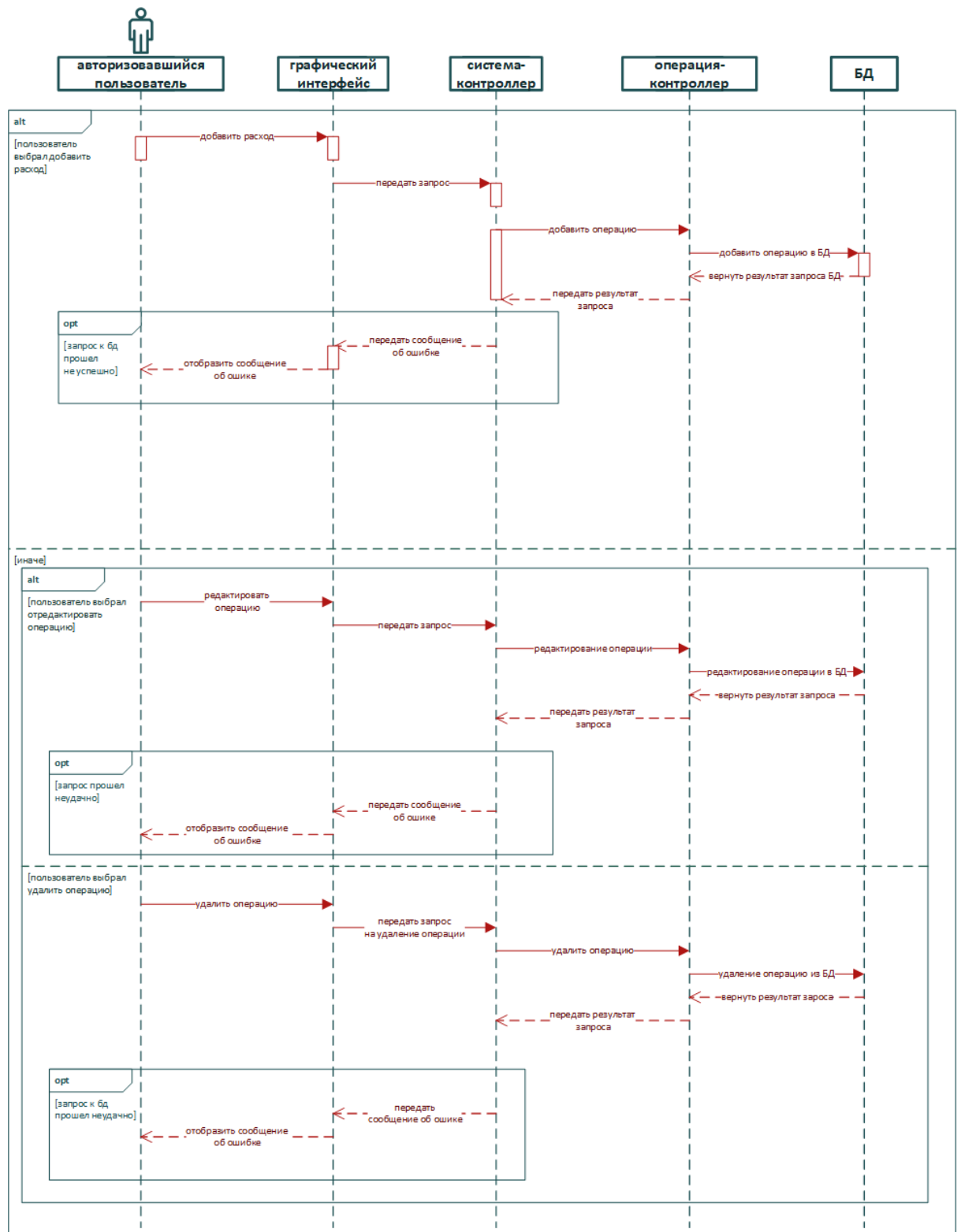
2.3.4. Диаграмма последовательностей и диаграмма коммуникаций

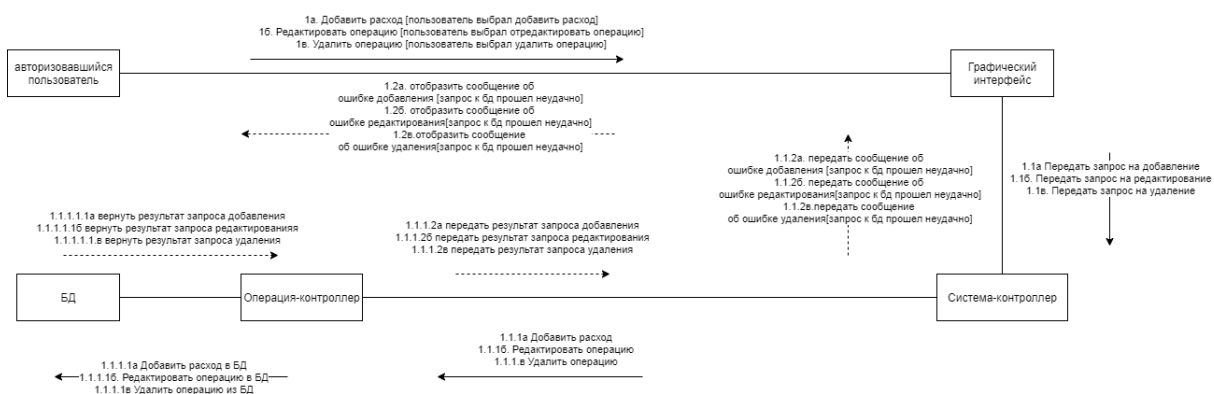




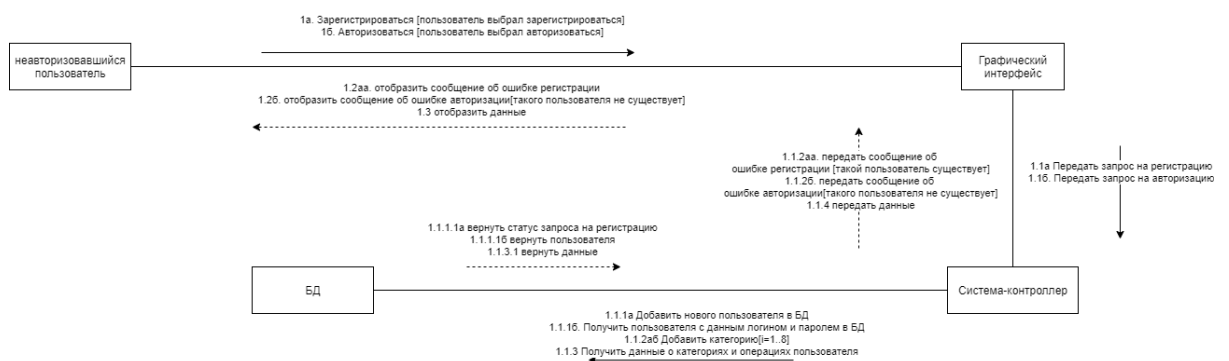
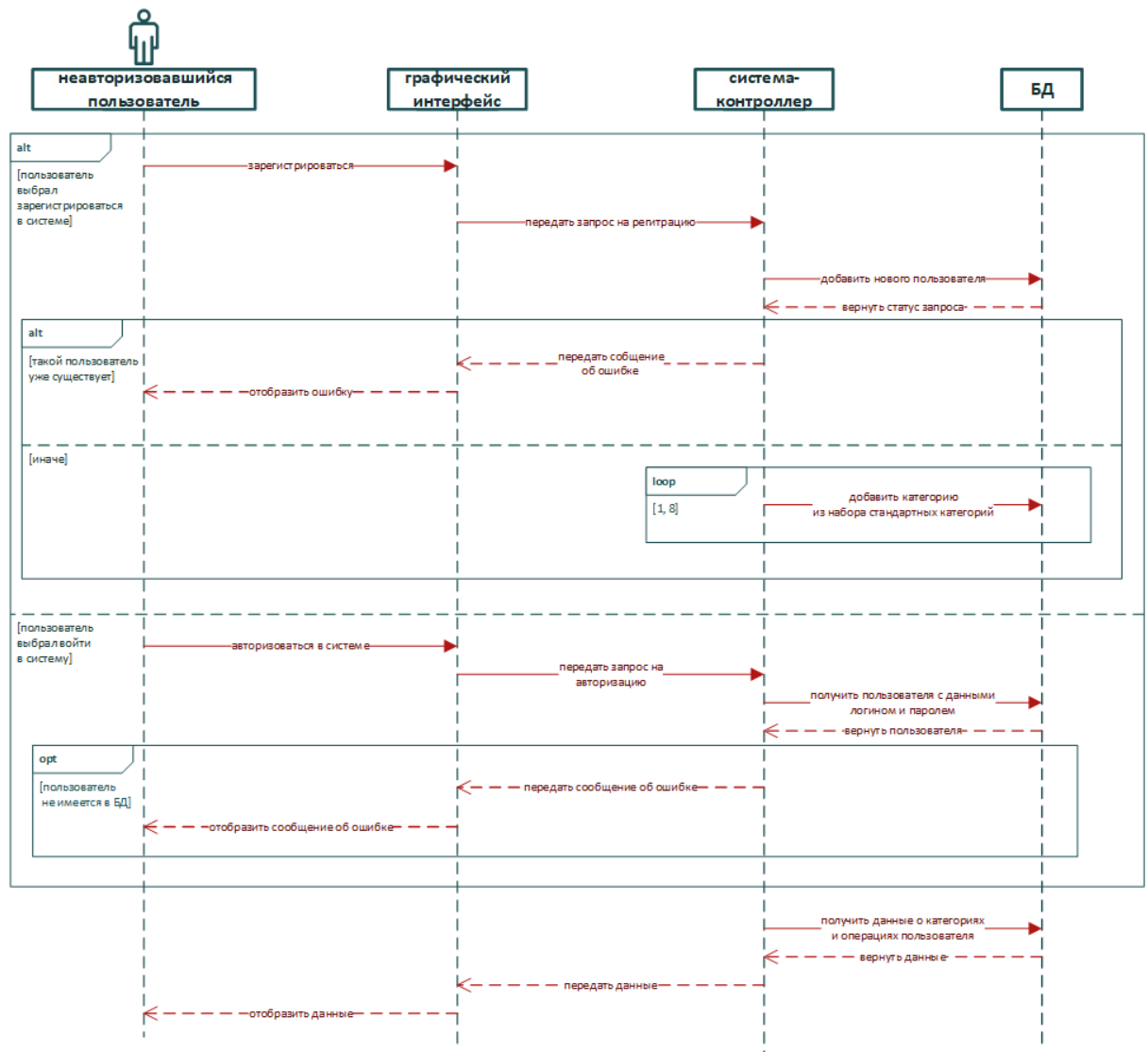
На приведенных рисунках продемонстрированы диаграммы последовательностей и коммуникаций, на которых показаны отношения между объектами при добавлении или редактировании или удалении одной конкретной категории.

Авторизовавшийся пользователь использует графический интерфейс, чтобы добавить новую категорию. Интерфейс в свою очередь передает запрос на добавление системе-контроллеру, система-контроллер обращается к категории-контроллер, который впоследствии добавляет в существующую базу данных запись о новой категории. База данных возвращает результат запроса, и в случае, если запрос прошел неудачно, система-контроллер передает сообщение об ошибке графическому интерфейсу, а тот в свою очередь отображает сообщение об ошибке. Редактирование и удаление категории происходит по точно такому же сценарию.





На приведенных рисунках показаны диаграммы последовательностей и коммуникаций, на которой продемонстрированы отношения между объектами при добавлении или редактировании или удалении операции. Авторизованный пользователь использует графический интерфейс, чтобы добавить новый расход. Интерфейс в свою очередь передает запрос на добавление расхода системе-контроллеру, система-контроллер обращается к операции-контроллер, который и добавляет в существующую базу данных запись о новом расходе. База данных возвращает результат запроса, и в случае, если запрос прошел неудачно, система-контроллер передает сообщение об ошибке графическому интерфейсу, а тот отображает сообщение об ошибке. Редактирование и удаление операции происходит по точно такому же сценарию.

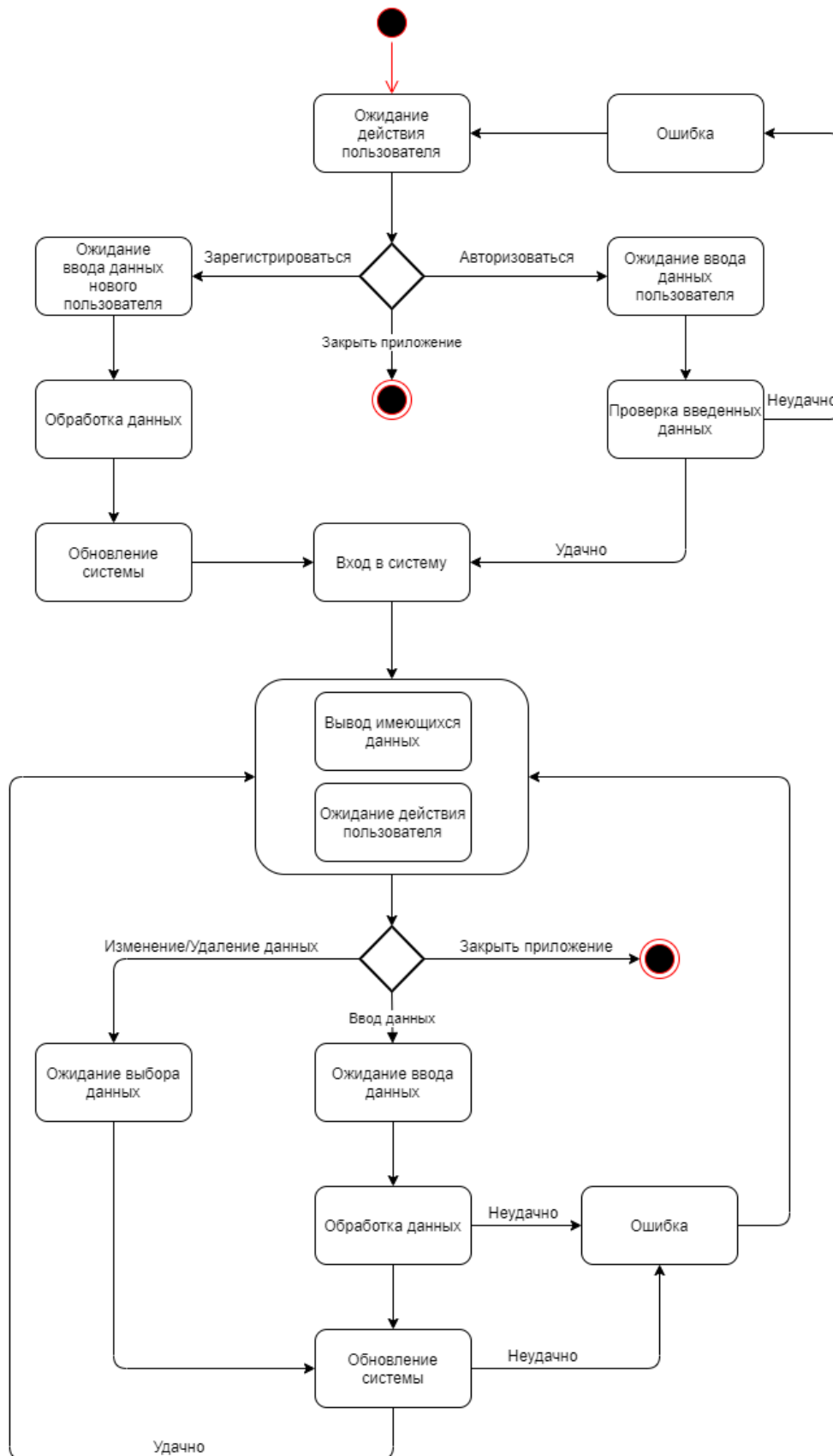


На рисунках показаны диаграммы последовательностей и коммуникаций, на которых продемонстрированы отношения между объектами при регистрации или авторизации пользователя.

Если пользователь выбрал регистрацию происходит следующая последовательность действий: не авторизовавшийся пользователь использует графический интерфейс, чтобы зарегистрироваться в системе. Затем данный графический интерфейс передает запрос на регистрацию системе-контроллеру, которая и добавляет нового пользователя в существующую базу данных. Она (база данных) возвращает результат запроса и в случае неудачного результата система-контроллер передает графическому интерфейсу сообщение об ошибке, который в свою очередь отображает его пользователю. Если запрос прошел удачно, тогда система-контроллер создает восемь стандартных категорий для нового пользователя.

Если пользователь выбрал авторизацию — не авторизовавшийся пользователь использует графический интерфейс, чтобы авторизоваться в системе. После чего графический интерфейс передает запрос на авторизацию системе-контроллеру, которая и получает нового пользователя из существующей базы данных. База данных возвращает запись о пользователе и в случае неудачного результата система-контроллер передает графическому интерфейсу сообщение об ошибке, который в свою очередь отображает его пользователю.

2.3.5. Диаграмма состояний



Данная диаграмма состояний отражает все возможные состояния системы, в которых она может находиться. При входе в приложение Smart Expenses, система находится в состоянии ожидания действий пользователя. Из этого состояния она может перейти в одно из следующих в зависимости от конкретных действий пользователя:

- Зарегистрироваться
- Авторизоваться
- Заккрыть приложение

В случае, если пользователь принял решение зарегистрироваться, система переходит в состояние ожидания ввода данных нового пользователя, затем введенные пользователем данные обрабатываются, то есть система находится в состоянии обработки данных, после чего она (система) обновляется и в последствии происходит вход в систему.

При авторизации пользователя система также входит в состояние ожидания ввода данных пользователя, а затем в состояние проверки введенных данных и, если введенные данные корректны, происходит вход в систему. В противном случае система входит в состояние ошибки и возвращается в начальное состояние ожиданий действий пользователя как при входе в систему.

Если пользователь закроет приложение, произойдет выход из системы, то есть из самого приложения.

После входа в систему, приложение переходит в состояние ожидания действия пользователя, при котором у него (пользователя) есть следующие возможности:

- Изменение/Удаление данных
- Ввод данных
- Заккрыть приложение

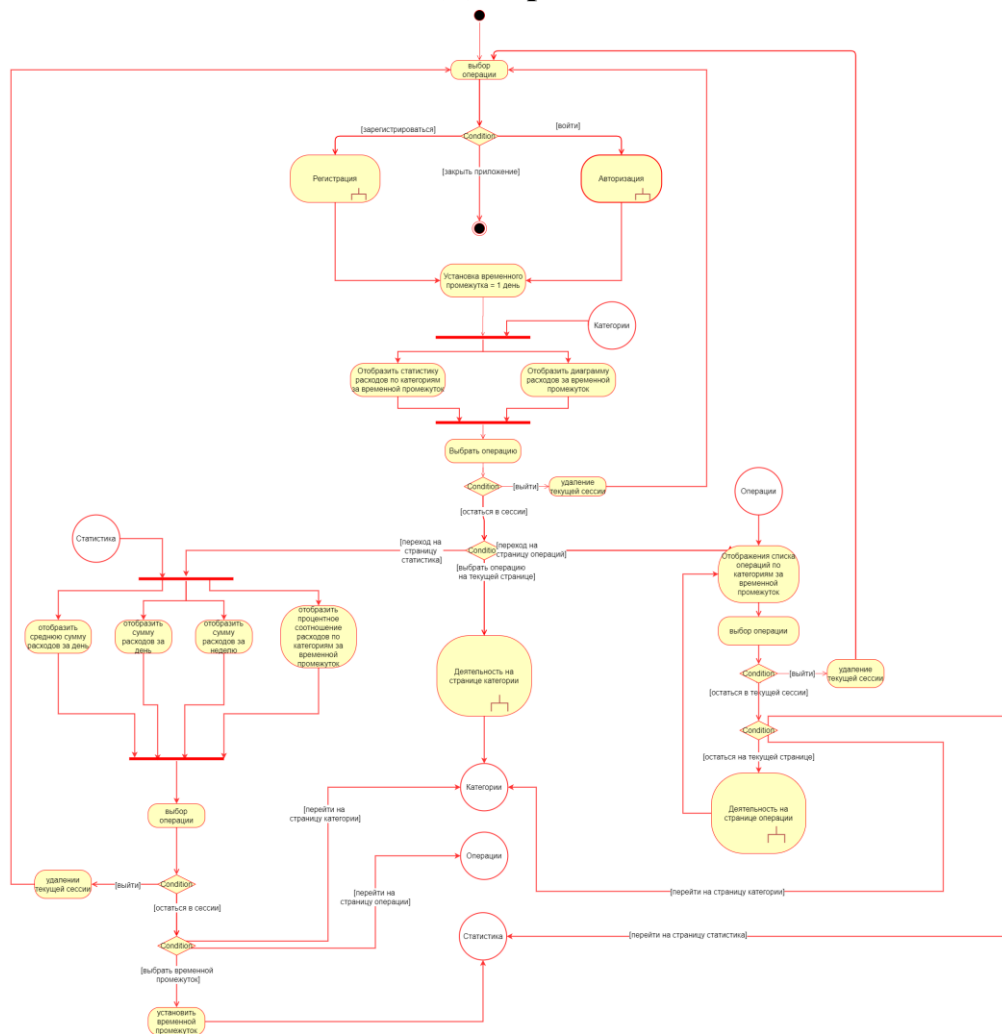
При изменении или удалении данных, система переходит в состояние ожидания выбора тех существующих в базе данных, которые хочет изменить или удалить пользователь приложения Smart Expenses. После чего система входит в

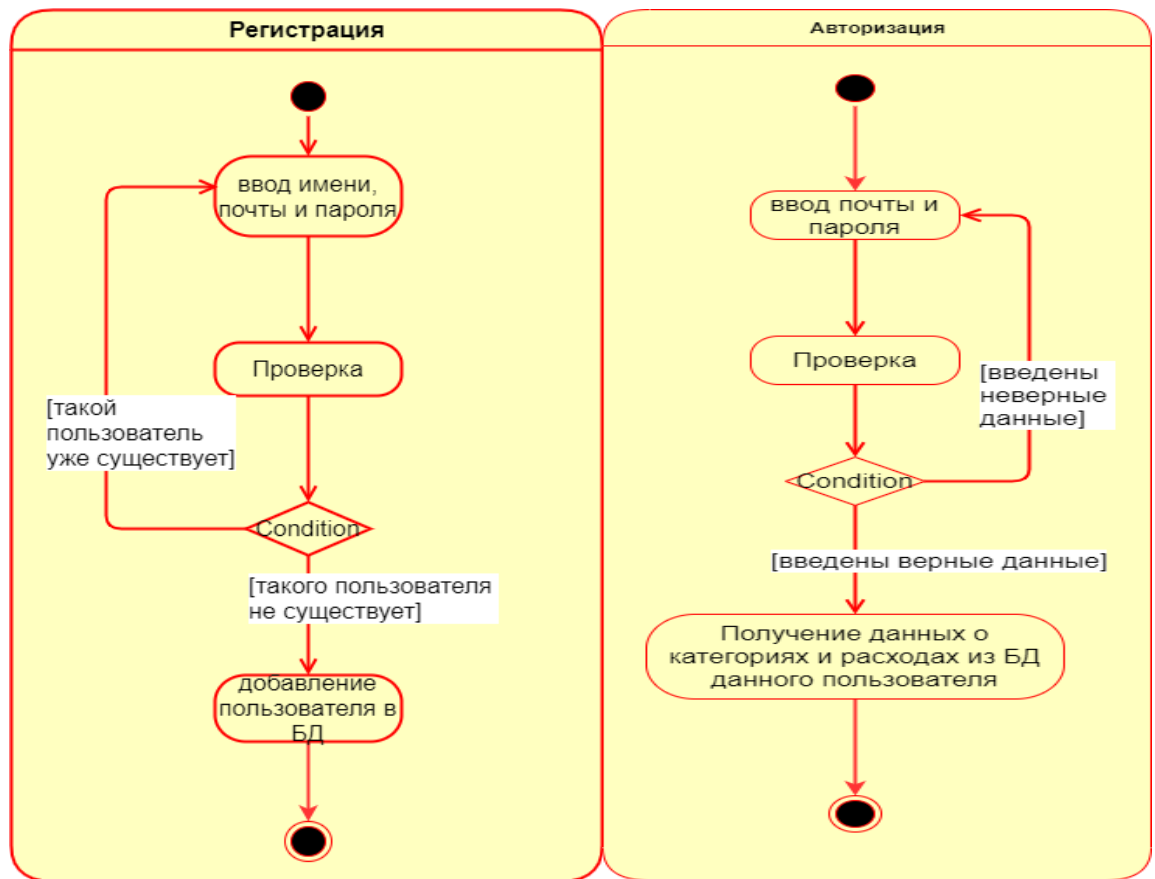
состояние обновления данных. Если данные обновились успешно, система переходит в состояние ожидания тех же действий пользователя как и после входа в систему, иначе — произойдет ошибка.

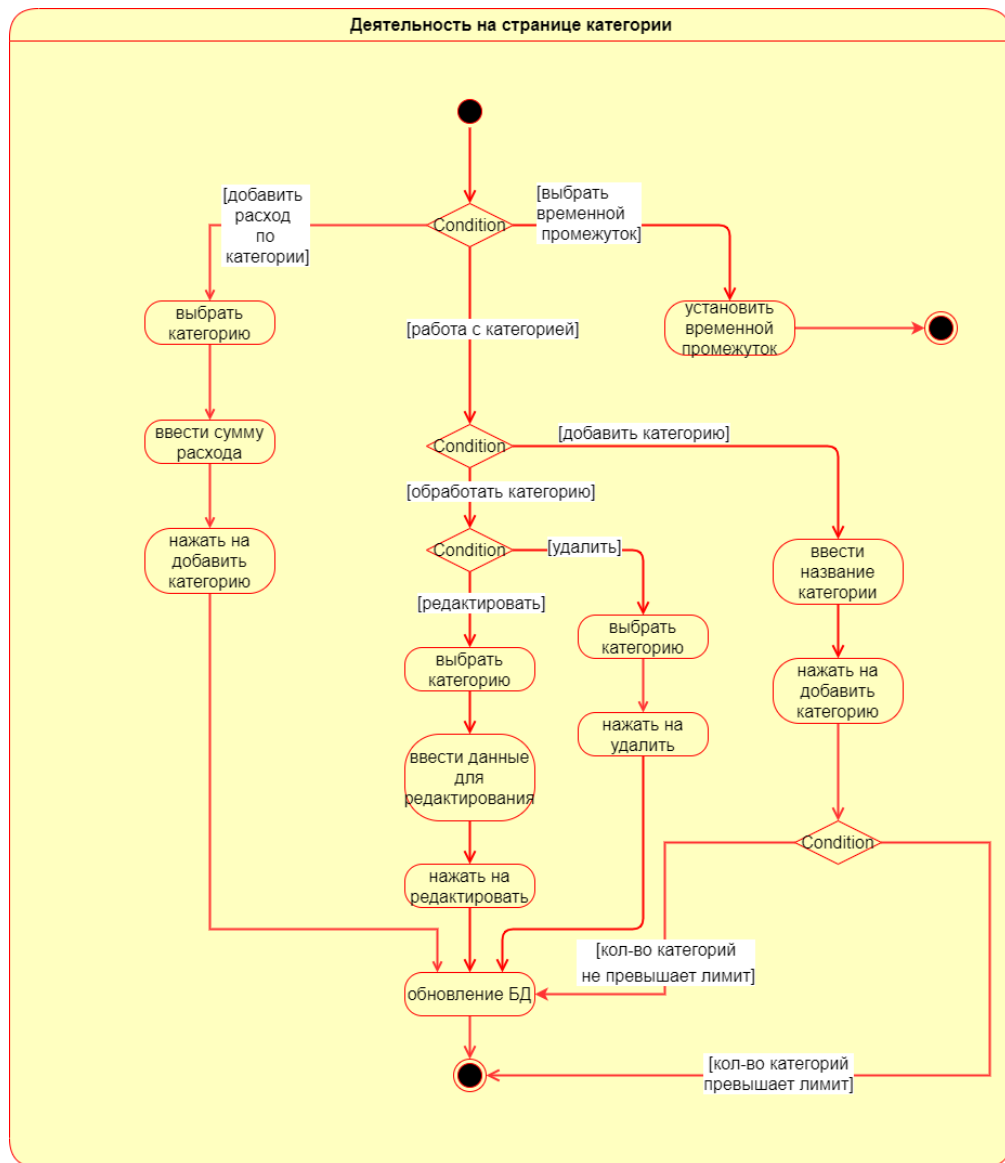
Если пользователь выберет ввод новых данных, система будет ожидать ввода данных, которые впоследствии нужно будет обработать. В случае удачной обработки введенной пользователем информации, произойдет обновление системы, однако, если полученные данные были некорректны, приложение выдаст ошибку и предложит пользователю заново их ввести. При успешном обновлении системы, приложение будет ожидать дальнейших действий пользователя, в противном случае произойдет ошибка, информация не сохранится и пользователю приложения будет необходимо проделать действия заново.

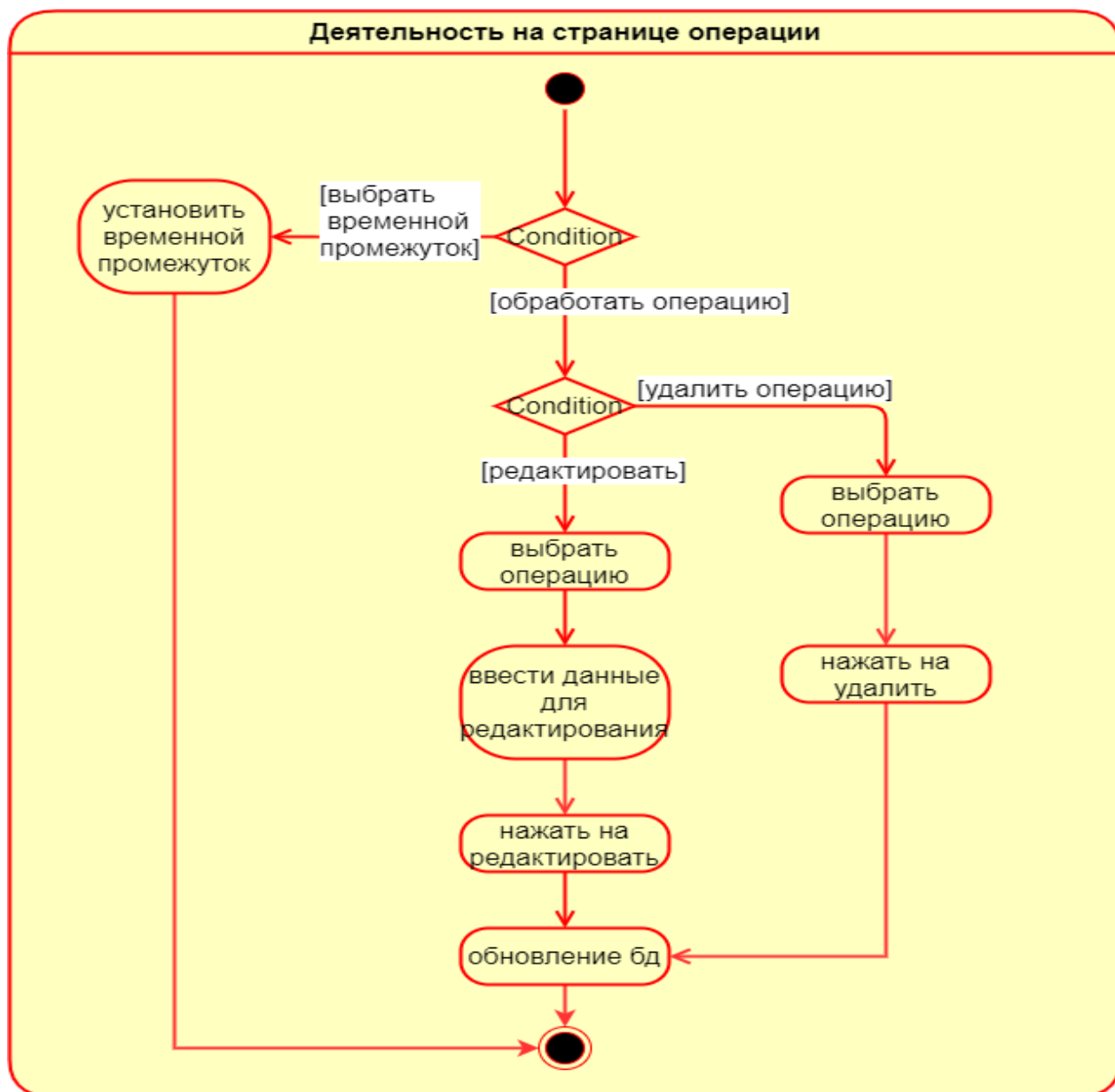
Если приложение закроется пользователем, так же произойдет выход из приложения.

2.3.6. Диаграмма активностей









При запуске приложения пользователь может выбрать 3 действия

- Зарегистрироваться
- Авторизоваться
- Закр​уть приложение

В случае, если пользователь выбрал закрыть приложение, происходит выход из данной системы.

Если пользователь выбрал зарегистрироваться в приложении, то он проходит те действия, которые описаны на диаграмме активности «Регистрация». Пользователю необходимо ввести имя, которое будет использоваться в приложении, адрес электронного почтового ящика и пароль, которые необходимы для

дальнейшего входа в приложение. Затем будет произведена проверка введенных данных и в случае, если такого пользователя не существует, вся информация добавится в базу данных, однако, если уже существует пользователь с такими же данными, система снова предложит ему (пользователю) ввести новые имя, адрес электронной почты и пароль.

Если пользователь выбрал авторизоваться в приложении, то он проходит те действия, которые описаны на диаграмме активности «Авторизация». Сначала нужно ввести указанные при регистрации адрес электронной почты и пароль, после чего произойдет проверка полученных данных и, если они верные, система запросит всю существующую информацию по категориям и расходам пользователя у базы данных и пользователь сможет продолжить работу с приложением, иначе, пользователю будет необходимо заново ввести значения почты и пароля.

Затем происходит установка временного промежутка системой равного 1 день (т.е. все расходы будут показаны за 1 день. По умолчанию тот день, в который был совершен вход).

Далее система отображает статистику расходов по категориям за временной промежуток и диаграмму расходов за временной промежуток.

Следующим шагом пользователь может выбрать одно из нескольких действий:

- Выйти из текущей сессии
- Выбрать действие на текущей странице
- Перейти на страницу «Операции»
- Перейти на страницу «Статистика»

При выходе из текущей сессии происходит удаление всех куки (или сессий) связанных с данным пользователем, и затем он переходит на самую первую деятельность на диаграмме активностей.

В случае выбора действия на текущей странице, пользователь проходит через действия, описанные на диаграмме «Деятельность на странице категории»,

после чего система отображает статистику расходов по категориям и диаграмму расходов за временной промежуток (день, неделя, месяц).

Далее пользователь возвращается к выбору действий на странице категории.

Т.к. с каждой из трех существующих страниц (категории, операции, статистика) можно будет перейти на любую из оставшихся двух, то деятельность на каждой из них мы рассмотрим только один раз, чтобы не повторяться.

В случае, если пользователь выберет перейти на страницу «Операции», система отобразит список операций по категориям за временной промежуток (день, неделя, месяц) и далее пользователь сможет выбрать одно из нескольких действий:

- Выйти из текущей сессии
- Выбрать действие на странице «Операции»
- Перейти на страницу «Категории»
- Перейти на страницу «Статистика»

Случай выхода из текущей сессии был рассмотрен ранее.

В случае работы с приложением на странице «Операции» пользователь проходит через действия описанные на диаграмме деятельности на странице «Операции», после которых действия на этой странице повторяются (система выводит список операций по категориям за временной промежуток, после чего выбор действия).

Деятельность в случае перехода пользователя на страницу «Категории» рассмотрена выше. Так как после входа в систему пользователь сразу попадает на страницу «Категории», которая была описана нами ранее.

В случае перехода на страницу «Статистика» пользователю отображается средняя сумма расходов за день, сумма расходов за день, сумма расходов за неделю, процентное соотношение по категориям за временной промежуток.

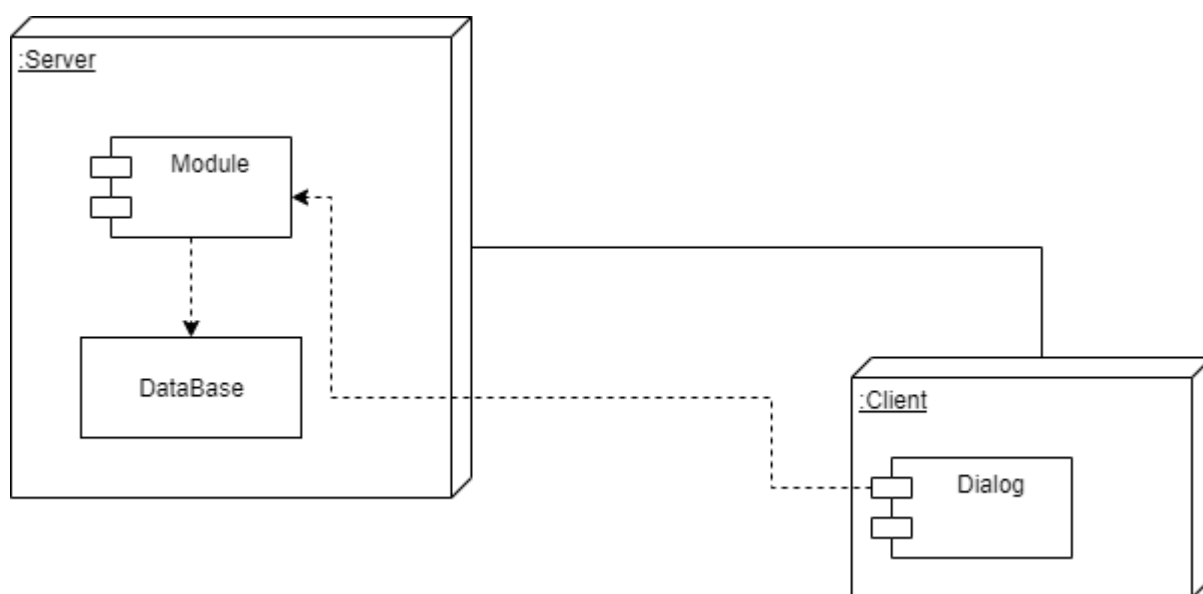
Затем пользователь может выбрать одно из следующих действий:

- Выйти из текущей сессии

- Выбрать операцию на текущей странице
- Перейти на страницу «Категории»
- Перейти на страницу «Операции»

Из вышеперечисленных осталось рассмотреть только выбор операции на текущей странице. В данном случае пользователь имеет возможность установить временной промежуток, на основе которого будет отображаться вышеупомянутая статистика. После соответствующего действия система вновь отображает статистику на текущей странице, затем, пользователь вновь выбирает одно из вышеперечисленных действий.

2.3.7. Диаграмма развертывания



Представленная нами диаграмма развертывания служит для демонстрации существующих аппаратных и программных компонентов приложения Smart Expenses и взаимодействия между ними.

На стороне сервера реализован весь необходимый функционал для успешной работы разрабатываемой системы, который неразрывно связан с базой данных, расположенной здесь же на сервере.

На клиентской стороне приложения находится диалоговое окно, которое необходимо для обработки всех действий пользователя и передачи введенных или выбранных данных.

2.4. Обоснование архитектуры

В качестве основного паттерна проектирования системы был выбран паттерн фасад, который предоставляет унифицированный интерфейс вместо набора интерфейсов некоторой подсистемы. Паттерн фасад определяет интерфейс более высокого уровня, который упрощает использование подсистемы. Использование данного шаблона проектирования при разработке приложения Smart Expenses позволило достичь:

- уменьшение количества связей между клиентом и системой
- возможность выделения подсистем компонентов в одной системе
- возможность упрощения работы с системой, определив одну точку взаимодействия

2.5. Анализ средств реализации

Для реализации и разработки приложения Smart Expenses были выбраны следующие технологии:

- РНР в качестве основного языка программирования вследствие следующих причин:
 - Опенсорсный язык, что значительно сокращает затраты на разработку приложения
 - Отличается гибкостью и хорошей совместимостью. РНР совместим со всеми основными платформами: от Windows, Unix и Linux до MacOS; поддерживает большинство серверов (включая Apache, Microsoft IIS, Netscape, iPlanet, Caudium, Xitami и Tornado) и более 20 баз данных (среди которых — MySQL, MongoDB, PostgreSQL и др.)
 - Сервисы на РНР хорошо масштабируются и документируются. Это позволяет запускать различные модули ПО один за другим, внося по ходу необходимые поправки.

- JavaScript как встраиваемый язык для программного доступа к объектам приложений. Нашел применение в приложении как язык сценариев для придания интерактивности веб-страницам.
- HTML как стандартизированный язык текстовой разметки
- CSS в качестве формального языка описания внешнего вида веб-приложения Smart Expenses
- В качестве СУБД был выбран MySQL по нескольким причинам:
 - Простота в использовании: MySQL достаточно легко устанавливается, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных.
 - Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.
 - Масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных.
- Для администрирования СУБД MySQL было выбрано веб-приложение phpMyAdmin, потому что:
 - phpMyAdmin предоставляет удобный веб-интерфейс для администрирования базы данных.
 - С помощью phpMyAdmin удобно выполнять все работы по управлению базой данных MySQL.
- Взаимодействие между Front-end и Back-end осуществляется за счет REST API
- Обмен данными между Front-end и Back-end происходит с помощью передачи JSON файлов

3. Реализация

3.1. Серверная часть приложения

Серверная часть была реализована с использованием языка Php, а так же MySql. Серверная часть приложения предоставляет клиентской части доступ к набору основных методов, необходимых для реализации основных клиентских сценариев. Данный набор методов называется API. Организация взаимодействия между клиентом и сервером с использованием данного подхода позволяет иметь возможность доступа к набору методов данного приложения из других систем.

3.2. Графический интерфейс

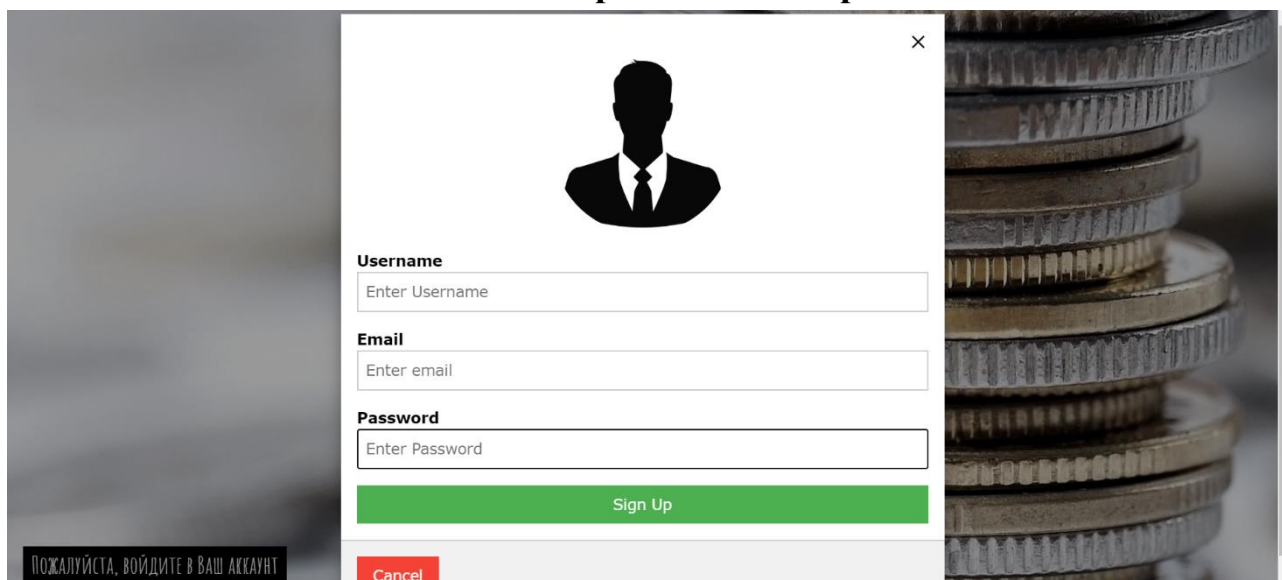
Клиентская часть была реализована с помощью языка разметки гипертекста (html + css), а так же с использованием языка программирования Javascript и его библиотеки – jQuery. Взаимодействие с сервером осуществляется с помощью асинхронных ajax-запросов к серверному API. Сервер посылает данные, а их представлением занимается клиент.

3.2.1. Главная страница



После входа на сайт пользователь попадает на главную страницу. Он имеет возможность авторизоваться или зарегистрироваться.

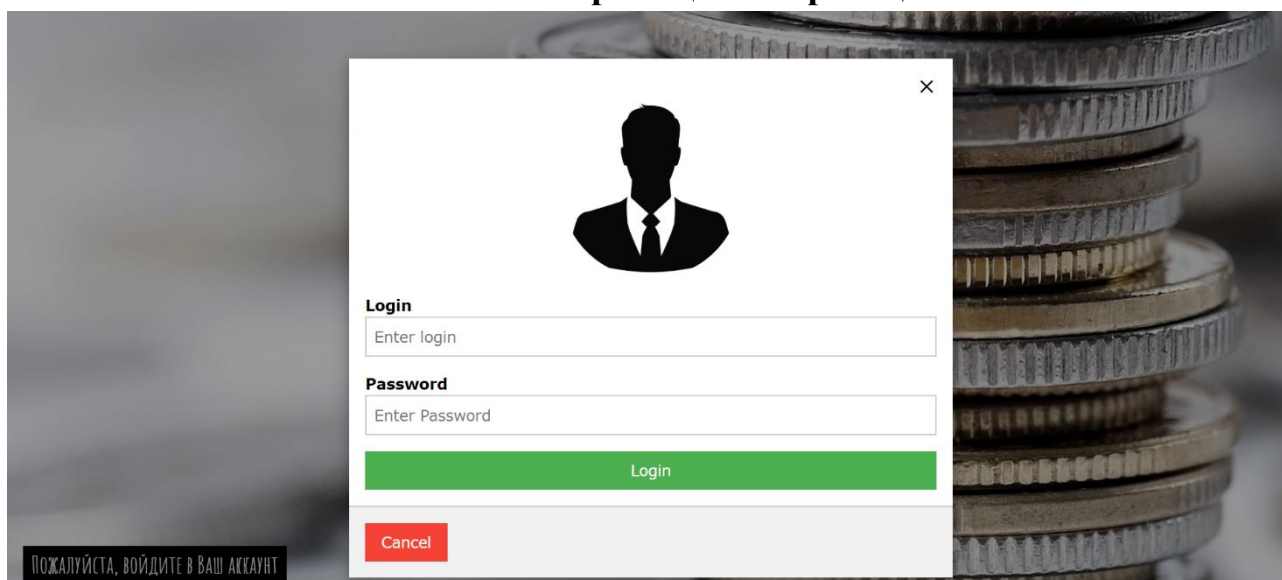
3.2.2. Страница Регистрации



The registration form is a white modal window with a close button (X) in the top right corner. It features a black silhouette of a person in a suit at the top center. Below the silhouette are three input fields: 'Username' with the placeholder 'Enter Username', 'Email' with the placeholder 'Enter email', and 'Password' with the placeholder 'Enter Password'. A green 'Sign Up' button is positioned below the password field. At the bottom left of the modal is a red 'Cancel' button. In the bottom left corner of the background image, there is a black box with the white text 'ПОЖАЛУЙСТА, ВОЙДИТЕ В ВАШ АККАУНТ'.

Для регистрации необходимо ввести имя пользователя, почту и пароль. Если пользователя с таким именем и почтой в системе нет, то будет создан новый.

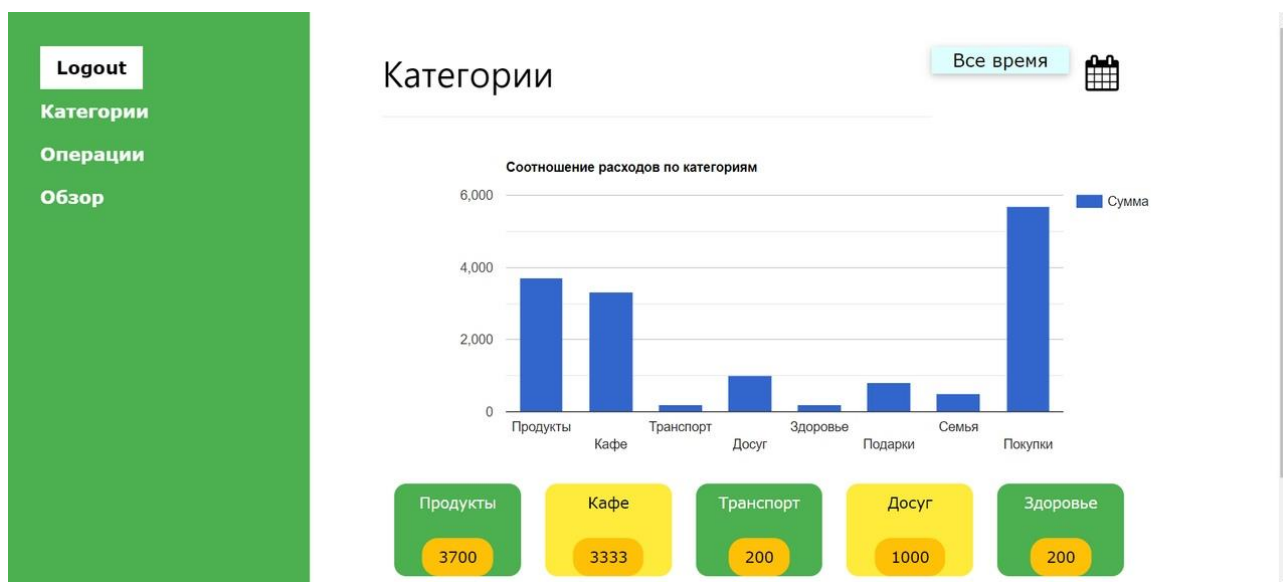
3.2.3. Страница Авторизации



The login form is a white modal window with a close button (X) in the top right corner. It features a black silhouette of a person in a suit at the top center. Below the silhouette are two input fields: 'Login' with the placeholder 'Enter login' and 'Password' with the placeholder 'Enter Password'. A green 'Login' button is positioned below the password field. At the bottom left of the modal is a red 'Cancel' button. In the bottom left corner of the background image, there is a black box with the white text 'ПОЖАЛУЙСТА, ВОЙДИТЕ В ВАШ АККАУНТ'.

Для авторизации достаточно ввести почту в качестве логина и пароль.

3.2.4. Страница «Категории»



После входа на сайт пользователь попадает на страницу «Категории». На странице отображены все категории пользователя и расходы по каждой из них за выбранный промежуток времени. Также здесь есть диаграмма, показывающая соотношение расходов по всем категориям.

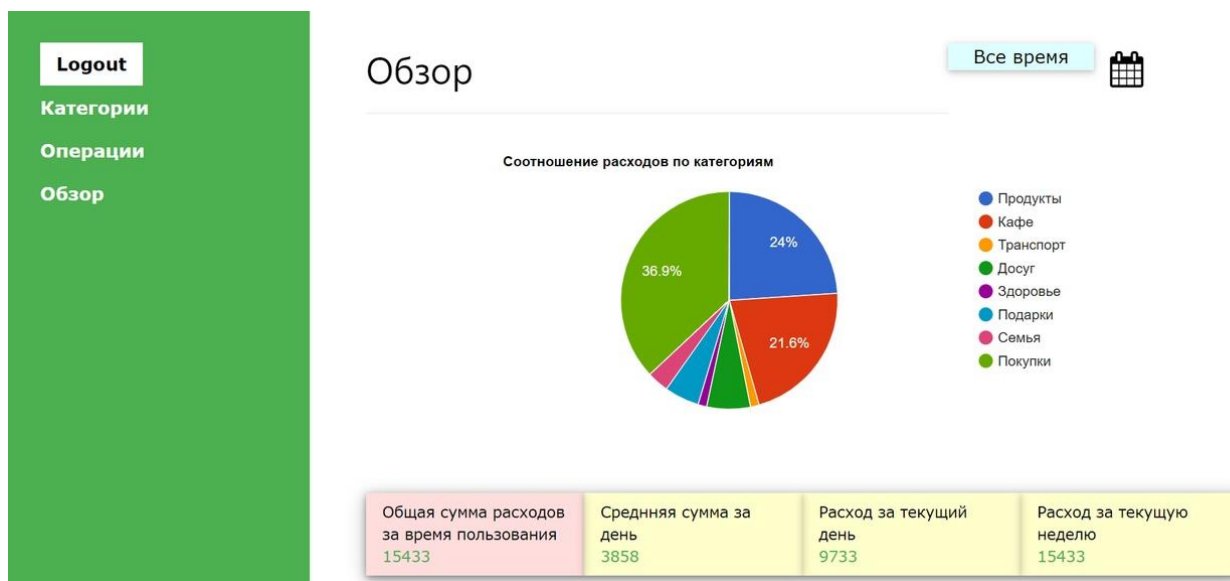
3.2.5. Страница «Операции»

The screenshot shows the 'Operations' page. On the left is a green sidebar with a 'Logout' button and links for 'Категории', 'Операции', and 'Обзор'. The main content area has the title 'Операции' and a date filter set to 'Все время'. A table lists individual transactions with their dates, categories, and amounts. Each row includes edit and delete icons.

Дата	Категория	Сумма	Действия
17.6.2020	Покупки	5700	✎ 🗑
19.6.2020	Семья	500	✎ 🗑
19.6.2020	Транспорт	200	✎ 🗑
19.6.2020	Кафе	3333	✎ 🗑
20.6.2020	—	—	—

На странице операции отображены все операции пользователя за выбранный промежуток. Также здесь можно удалять и редактировать операции

3.2.6. Страница «Обзор»



На странице «Обзор» отображена диаграмма, показывающая соотношение расходов по категориям и остальная статистика.

4. Тестирование

По завершении разработки приложения, было проведено тестирование по работе всего функционала веб-приложения Smart Expenses для проверки соответствия ожидаемого результата работы, указанного в техническом задании, с фактическим. Были выбран следующие виды ручного тестирования:

- Unit testing
- Smoke testing
- End-to-end testing
- Usability-testing

4.1. Smoke testing

Smoke testing представляет собой короткий цикл тестов, выполняемых для подтверждения того, что приложение выполняет основные функции. Для тестирования данного веб-приложения необходимо проверить работоспособность следующих основных возможностей:

- Регистрация
- Авторизация
- Добавление расхода по категории
- Добавление категории
- Удаление категории
- Изменение категории
- Удаление операции
- Редактирование операции

В ходе данного тестирования были получены результаты, представленные в следующей таблице:

To do	To check	Result
Регистрация		
Заполнить мандаторные поля	-	

Нажать кнопку «Зарегистрироваться»	Пользователь зарегистрирован. Произошел вход в систему	Tested/Passed
Авторизация		
Заполнить мандаторные поля	-	
Нажать кнопку «Войти»	Произошел вход в систему	Tested/Passed
Добавление расхода по категории		
Добавить расход по любой стандартной категории	Расход добавлен	Tested/Passed
Добавление категории		
Добавить новую категорию	Категория добавлена	Tested/Passed
Удаление категории		
Удалить любую категорию	Категория удалена	Tested/Passed
Изменение категории		
Изменить любую категорию	Категория изменена	Tested/Passed
Удаление операции		
Добавить расход по любой категории	Расход добавлен	
Удалить операцию	Операция удалена	Tested/Passed
Редактирование операции		
Добавить расход по любой категории	Расход добавлен	

Редактировать операцию	Операция отредактирована	Tested/Passed
------------------------	--------------------------	---------------

4.2. End-to-end testing

End-to-end testing был выбран в качестве самого важного вида тестирования, демонстрирующего работу всего функционала системы в целом. В нашем случае End-to-end testing состоит из одного тестового сценария, который приведен далее:

To do	To check	Result
Перейти в раздел регистрации нового пользователя	Открылась форма регистрации пользователя	Tested/Passed
Заполнить мандаторные поля	-	
Нажать кнопку «Регистрация»	Пользователь зарегистрирован. Произошел вход в систему	Tested/Passed
Во вкладке «Категории» добавить по два расхода на каждую категорию	Расход добавлен	Tested/Passed
Изменить расход по трем категориям	Расход был изменен	Tested/Passed
Добавить новую категорию	Новая категория добавлена	Tested/Passed
Добавить расход по новой категории	Расход добавлен	Tested/Passed
Открыть страницу «Операции»	Переход выполнен	Tested/Passed

	Удостовериться в наличии полной истории операции	Tested/Passed
Удалить одну из операций	Операция удалена	Tested/Passed
Редактировать одну из операций	Операция изменена	Tested/Passed
Открыть страницу «Категории»	Переход выполнен	Tested/Passed
	Проверить, что удаленная операция отсутствует в расходах по категории	Tested/Passed
	Проверить, что изменился расход по категории после изменения операции	Tested/Passed
Открыть страницу «Обзор»	Переход выполнен	Tested/Passed
	Проверить общую сумму расходов	Tested/Passed
	Проверить процентное соотношение расходов по категориям	Tested/Passed

4.3. Usability-testing

Usability-testing направлено на тестирование удобства пользования продуктом конечным пользователем. Данное тестирование показывает, насколько

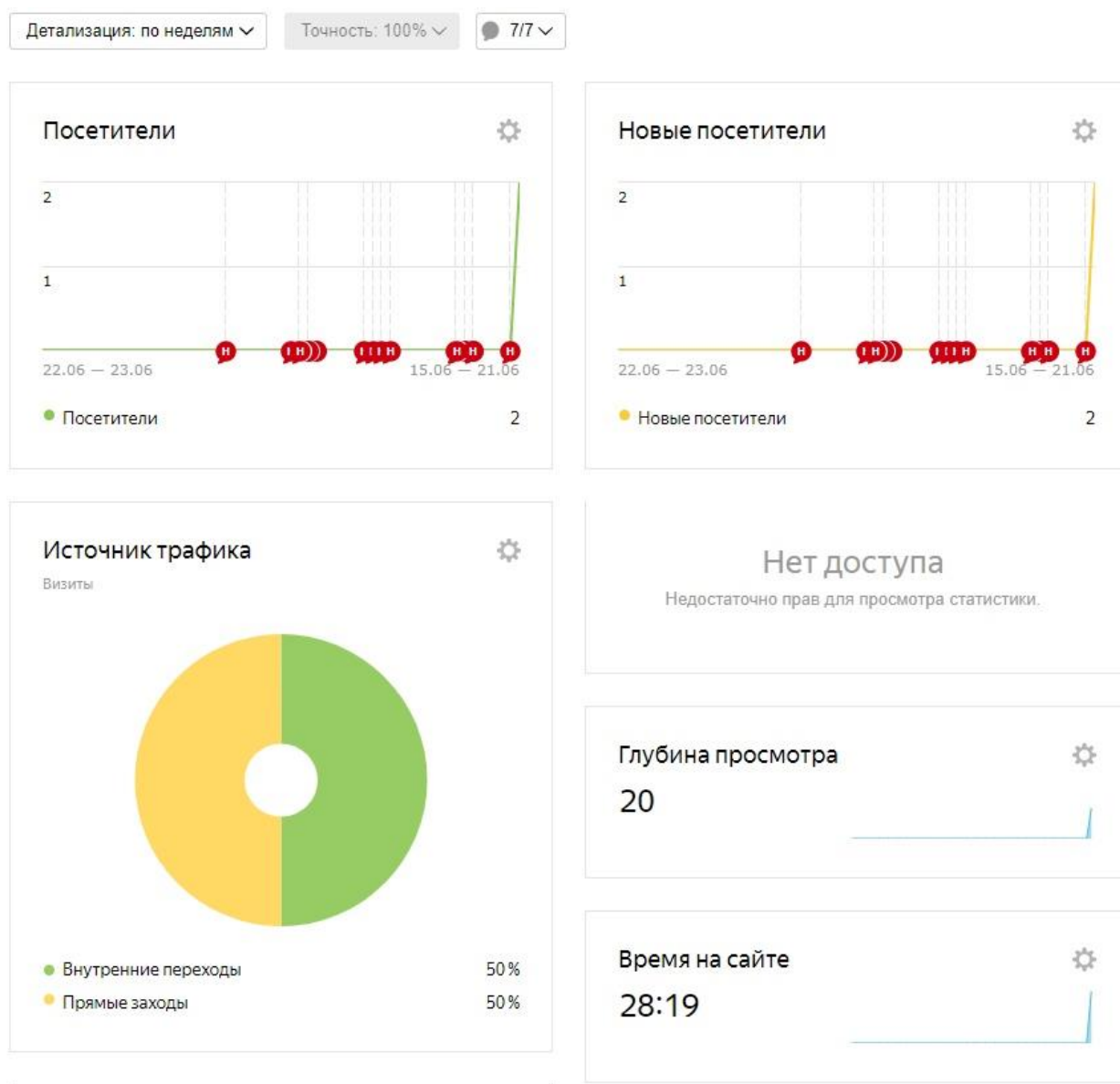
готовый продукт соответствует ожиданиям пользователей. Случайным образом было отобрано 4 человека, которые не были знакомы с данным приложением ранее. В данный вид тестирования мы включили следующие основные сценарии взаимодействия пользователей с приложением:

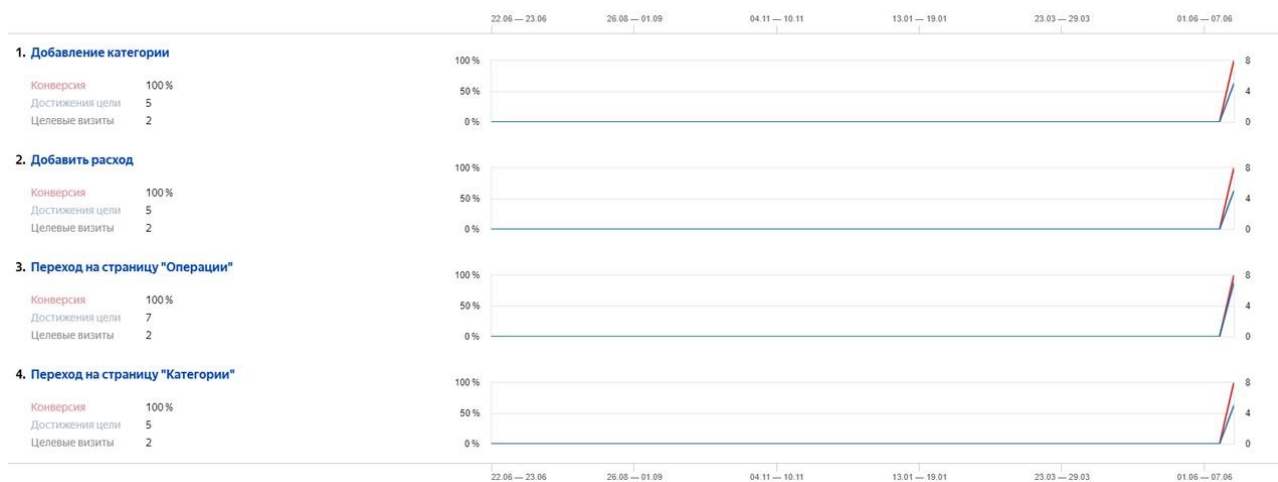
- Добавление расхода по категории
- Добавление категории
- Удаление категории
- Изменение категории
- Редактирование операции
- Удаление операции
- Просмотр диаграммы расходов по категориям

To do	User 1	User 2	User 3	User 4
Добавление расхода по категории	Passed	Passed	Passed	Passed
Добавление категории	Passed	Passed	Passed	Passed
Удаление категории	Passed	Passed	Passed	Passed
Изменение категории	Passed	Passed	Passed	Passed
Редактирование операции	Passed	Passed	Passed	Passed
Удаление операции	Passed	Passed	Passed	Passed
Просмотр диаграммы расходов по категориям	Passed	Passed	Passed	Passed

5. Статистика

Сбор статистики и аналитики был реализован с помощью приложения «Яндекс-метрика». Используя данный сервис была оценена посещаемость нашего веб-приложения и проанализировано поведение пользователей. Скриншоты статистики из приложения «Яндекс-метрика» представлены ниже.





Заключение

В результате проделанной работы было реализовано приложение Smart Expenses, которое позволяет пользователю осуществлять регистрацию и систематизацию своих расходов, а также просматривать различную статистику за определенный временной промежуток. Веб-приложение Smart Expenses обладает визуальным интерактивным интерфейсом, который необходим для взаимодействия пользователя с приложением и позволяет выводить на экран всю нужную информацию для удобства работы пользователя с сервисом.

Были выполнены следующие задачи:

- Разработаны диаграммы, необходимые для всего процесса написания веб-приложения
- Разработан Front-end приложения, который необходим для взаимодействия пользователя с сервисом Smart Expenses
- Разработан Back-end приложения, развернутый на удаленном сервере приложения, необходимы для работы всего функционала сервиса
- Разработана база данных, необходимая для хранения всей необходимой информации о пользователе и его расходов, расположенная на удаленном сервере приложения
- С помощью API была реализована связь между Back-end и Front-end приложения