УДК 004.65

5.2.2. Математические, статистические и инструментальные методы экономики (физико-математические науки, экономические науки)

ПРИМЕНЕНИЕ USE CASE и USER STORY ПРИ ПРОЕКТИРОВАНИИ ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ЛИЧНЫМИ ФИНАНСАМИ

Авакимян Наталья Николаевна
К.ф.–м.н., доцент
РИНЦ SPIN–код: 6082–4770
email: avnatali@mail.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Агеенко Богдан Михайлович
студент группы БИ2102
email: ageen.bogdan@yandex.ru
ФГБОУ «Кубанский государственный аграрный университет», 350044, Россия, г. Краснодар, ул. Калинина 13

Приложение для управления личными финансами предназначено для упрощения учета доходов, расходов и планирования бюджета пользователями, стремящимися к финансовой стабильности и грамотному распределению средств. Система позволяет хранить информацию о различных категориях расходов и доходов, отслеживать динамику накоплений, формировать финансовые цели и вести аналитику по заданным периодам. Благодаря удобному интерфейсу и автоматическим напоминаниям пользователи могут оптимизировать свои финансовые привычки и принимать взвешенные решения, приводящие к улучшению финансового здоровья. В данной статье для выбранной предметной области выделены основные пользователи системы и роли, бизнес-требования, Epic-задача, детализированы User Story для данного проекта для пользователя и клиента. Расписаны критерии приемки и Task-задачи для некоторых Story. Приведены примеры UseCase и шаблон для его описания. Описаны входные и выходные параметры. Приведен пример запроса и ответа

Ключевые слова: БИЗНЕС-ТРЕБОВАНИЯ, USER CASE, USER STORY, СУЩНОСТИ, МОНОЛИТНАЯ АРХИТЕКТУРА

http://dx.doi.org/10.21515/1990-4665-208-030

UDC 004.65

5.2.2. Mathematical, statistical and instrumental methods of economics (physical and mathematical sciences, economic sciences)

USING USE CASE AND USER STORY IN DESIGNING A PERSONAL FINANCE MANAGEMENT APPLICATION

Avakimyan Natalia Nikolaevna
Cand.Phys.–Math.Sci., associate professor
RSCI SPIN–code: 6082–4770
email: avnatali@mail.ru.
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

Ageenko Bogdan Mikhailovich
student of group BI2102
email: ageen.bogdan@yandex.ru
Kuban State Agricultural university, 350044, Russia, Krasnodar, Kalinina, 13

The personal finance management application is designed to simplify the accounting of income, expenses and budget planning for users striving for financial stability and competent distribution of funds. The system allows you to store information on various categories of expenses and income, track the dynamics of savings, form financial goals and conduct analytics for specified periods. With a user-friendly interface and automatic reminders, users can optimize their financial habits and make informed decisions that lead to improved financial health. In this article, for the selected subject area, the main users of the system and roles, business requirements, Epic task are highlighted, User Stories for this project for the user and the client are detailed. Acceptance criteria and Tasks for some Stories are described. Examples of UseCase and a template for its description are given. Input and output parameters are described. An example of a request and response is given

Keywords: BUSINESS REQUIREMENTS, USE CASE, USER STORY, ENTITIES, MONOLITHIC ARCHITECTURE

Приложение для управления личными финансами предназначено для

упрощения учета доходов, расходов и планирования бюджета пользователями, стремящимися к финансовой стабильности и грамотному распределению средств. Система позволяет хранить информацию о различных категориях расходов и доходов, отслеживать динамику накоплений, формировать финансовые цели и вести аналитику по заданным периодам. Благодаря удобному интерфейсу и автоматическим напоминаниям пользователи могут оптимизировать свои финансовые привычки и принимать взвешенные решения, приводящие к улучшению финансового здоровья. Приложение способствует повышению прозрачности личных финансов и помогает исключить лишние траты за счет регулярных отчетов и статистики. Функционал продукта дополнительно дает возможность устанавливать лимиты по категориям расходов, что способствует формированию более осознанного бюджета и достижению поставленных целей.

Основные пользователи системы и роли:

1) Незарегистрированный пользователь (Guest):

– может ознакомиться с возможностями приложения (демо-режим или страница с описанием функционала);

– имеет доступ только к общим информационным разделам (FAQ, страница регистрации/логина);

– не имеет возможности вносить и редактировать данные по личным финансам.

2) Зарегистрированный пользователь (User):

– имеет полный доступ к личному кабинету;

– может добавлять доходы и расходы, редактировать и удалять их, устанавливать бюджеты и цели, просматривать аналитику;

– может изменять настройки своей учетной записи (пароль, email и т. д.).

3) Администратор (Admin):

– имеет доступ ко всем данным в административной панели;

– может модифицировать справочники категорий (общие для всех пользователей, если такие используются);

– контролирует безопасность (блокировка/разблокировка аккаунтов при нарушении правил);

– настраивает технические параметры приложения (резервное копирование, мониторинг системы, тарифные планы, если предусмотрено).

Ниже перечислены высокоуровневые бизнес-требования к приложению для управления личными финансами (без детальной прорисовки пошаговых сценариев):

1. Безопасная аутентификация и регистрация Приложение должно предоставлять безопасный процесс регистрации и входа, включая возможность авторизации через электронную почту и пароль, а также через сторонние социальные или почтовые сервисы (например, Google). Обязательно должна поддерживаться функция восстановления пароля через email, чтобы обеспечить удобство и доверие пользователей.

2. Управление доходами и расходами по категориям Система должна позволять пользователям добавлять доходы и расходы в разные категории (например, «Продукты», «Транспорт», «Зарплата», «Подарки», «Развлечения» и т. д.), редактировать и удалять их. Пользователь должен иметь возможность создавать собственные категории, а также изменять названия/иконки уже существующих.

3. Установка бюджетов и финансовых целей Необходимо предоставить механизм постановки финансовых целей (накопить определенную сумму к конкретной дате) и установки месячного/недельного бюджета на определенные категории расходов. По достижении либо превышении лимита система должна уведомлять пользователя (push-уведомлением или e-mail).

4. Аналитика и отчеты в разных разрезах Приложение должно формировать отчеты о доходах и расходах за выбранный период с удобной визуализацией (диаграммы, графики, сводные таблицы). Аналитика должна помогать пользователю отслеживать динамику расходов, понимать, где можно сэкономить, и видеть прогресс по накоплению средств на финансовую цель.

5. Интеграция с платежными системами (при необходимости) Приложение может предлагать возможность подключения к внешним финансовым сервисам (например, интеграция с банками или электронными кошельками) для автоматической загрузки операций по счетам пользователя. Это повысит актуальность данных и снизит усилия пользователя, связанные с ручным вводом расходов/доходов.

В рамках данного приложения можно создать один крупный Epic для всей системы управления личными финансами, так как все основные пользовательские сценарии укладываются в один контекст: Epic: PFM-001 Разработка системы управления личными финансами (Personal Finance Management)

Формирование Story и описание по технике UserStory предусматривает выделение:

1. PFM-002 Регистрация пользователя.
2. PFM-003 Авторизация пользователя.
3. PFM-004 Управление доходами.
4. PFM-005 Управление расходами.
5. PFM-006 Создание и отслеживание финансовых целей.
6. PFM-007 Установка и контроль бюджета по категориям.
7. PFM-008 Аналитика и формирование отчетов.
8. PFM-009 Интеграция с платежными системами (опционально).
9. PFM-010 Управление собственными категориями.
10. PFM-011 Управление общими настройками профиля.

User Stories для незарегистрированного пользователя (Guest):

– Story: «Регистрация пользователя» (PFM-002). As a Guest, I can create a new account by providing the required personal and credential information so that I can access the full functionality of the application and securely store my financial data.

– Story: «Авторизация пользователя» (PFM-003). As a Guest, I can log into the system using my email and password (или сторонний сервис авторизации) so that I can get access to my personal finance dashboard and features.

User Stories для зарегистрированного пользователя (User):

– Story: «Управление доходами» (PFM-004). As a Registered User, I can add, edit, and delete my income entries (зарплата, бонусы, проценты по депозитам и т. д.) so that I can keep track of my earnings and analyze them over time.

– Story: «Управление расходами» (PFM-005). As a Registered User, I can add, edit, and delete my expense records (продукты, транспорт, развлечения и т. д.) so that I can effectively control my outflow of money and maintain an accurate budget.

– Story: «Создание и отслеживание финансовых целей» (PFM-006). As a Registered User, I can set specific financial goals (накопить на отпуск, погасить кредит и пр.) so that I can monitor my progress and see how close I am to achieving each goal.

– Story: «Установка и контроль бюджета по категориям» (PFM-007). As a Registered User, I can define monthly (or weekly) spending limits on specific categories (e.g., «Развлечения») so that I receive alerts if I approach or exceed the limit and thereby avoid overspending.

– Story: «Аналитика и формирование отчетов» (PFM-008). As a Registered User, I can generate visual reports (diagrams, charts) on my income, expenses, and savings for selected periods so that I can analyze my financial

behavior and identify potential savings or overspending.

– Story: «Интеграция с платежными системами» (PFM-009) (необязательно). As a Registered User, I can connect my external bank account or e-wallet so that transaction data is automatically imported into the application, saving me time and reducing the chance of manual entry errors.

– Story: «Управление собственными категориями» (PFM-010). As a Registered User, I can create, rename, or delete my own categories of expenses and incomes so that I can personalize the application according to my specific needs.

– Story: «Управление общими настройками профиля» (PFM-011). As a Registered User, I can change my password, email, notification settings, etc. so that I can maintain control over my personal data and how I receive alerts.

User Stories для администратора (Admin):

– Story: «Управление пользователями и настройками приложения» (не имеет отдельного Story в списке выше, но может быть добавлен при необходимости). As an Admin, I can manage user accounts, block or unblock them, and modify system-wide settings so that I can ensure the platform runs smoothly and securely for all users.

Критерии приемки (Acceptance Criteria):

1) Регистрация пользователя:

Scenario: User Registration

Given the Guest is on the "Sign Up" page

When the Guest enters "valid email" in the Email field

And the Guest enters "password" in the Password field

And the Guest enters "firstName" and "lastName"

And the Guest clicks on "Sign Up" button

Then the system should create a new user account

And a confirmation email should be sent

And the user should be redirected to a welcome page

2) Авторизация пользователя:

Scenario: User Login

Given the Guest is on the "Login" page

When the Guest enters "registered email" in the Email field

And the Guest enters "correct password" in the Password field

And the Guest clicks on "Login" button

Then the user should be authenticated

And the user sees the personal finance dashboard

3) Добавление новой записи о расходе:

Scenario: Add an expense

Given the User is logged in

And the User is on the "Expenses" page

When the User clicks on "Add Expense"

And the User enters "Groceries" in the Category field

And the User enters "1000" in the Amount field

And the User selects today's date

And the User clicks "Save"

Then the new expense is recorded

And the user sees the updated expenses list

And the total expenses amount is recalculated

4) Установка бюджета на категорию «Питание»

Scenario: Set monthly budget

Given the User is logged in

And the User is on the "Budgets" page

When the User sets category "Food" with limit "20000" for the current month

And the User clicks "Save"

Then the budget limit is saved

And the system will notify the User upon exceeding 80% of that limit

5) Создание финансовой цели

Scenario: Create a financial goal

Given the User is on the "Goals" page

When the User enters "New Laptop" as the goal name

And the User enters "60000" as the target amount

And the User sets the target date 3 months from now

And the User clicks "Create Goal"

Then the goal is added to the user’s goals list

And the system starts tracking progress

Сформируем Task-задачи для каждой Story:

– Story «PFM-002 Регистрация пользователя»:

а) PFM-012 Разработка формы регистрации – создать web-форму, включающую поля email, пароль, подтверждение пароля, имя/фамилию.

б) PFM-013 Валидация данных на клиентской стороне – проверять корректность формата email, соответствие пароля требованиям (длина, сложность) и совпадение пароля с подтверждением.

в) PFM-014 Настройка серверного API для регистрации – создать эндпоинт для приема POST-запросов с данными. Реализовать логику сохранения в базу данных.

г) PFM-015 Реализация отправки подтверждения по email – после успешного добавления пользователя в БД отсылать письмо со ссылкой для активации аккаунта (опционально).

– Story «PFM-003 Авторизация пользователя»:

а) PFM-016 Разработка формы авторизации – создать web-форму, включающую поля email и пароль.

б) PFM-017 Настройка механизма сессий и токенов – реализовать хранение сессий на сервере / JWT-токен при использовании SPA/мобильного клиента.

в) PFM-018 Настройка серверного API для обработки запросов на

вход – проверять соответствие email/пароля, возвращать успешный ответ при совпадении данных.

г) PFM-019 Реализация функционала «Забыли пароль?» – отправка email со ссылкой для восстановления пароля, валидация ссылки и смена пароля.

– Story «PFM-004 Управление доходами»

а) PFM-020 Создать таблицу/модель «Доходы». Структура: сумма, дата, категория, описание, пользователь.

б) PFM-021 Реализовать интерфейс для добавления дохода. Форма добавления/редактирования, привязка к пользовательскому аккаунту.

в) PFM-022 Разработать API для CRUD-операций по доходам. POST/GET/PUT/DELETE для работы с записями.

г) PFM-023 Обновить общую аналитику при добавлении/изменении дохода – пересчитывать сводные показатели (общая сумма доходов за месяц, графики).

– Story «PFM-005 Управление расходами»:

а) PFM-024 Создать таблицу/модель «Расходы».

б) PFM-025 Реализовать интерфейс для добавления расхода.

в) PFM-026 Разработать API для CRUD-операций по расходам.

г) PFM-027 Обновить аналитику и уведомлять, если происходит превышение бюджета (при настройке лимитов).

– Story «PFM-006 Создание и отслеживание финансовых целей»:

а) PFM-028 Создание таблицы/модели «Цели».

б) PFM-029 Интерфейс для добавления/редактирования цели.

в) PFM-030 API для работы с целями.

г) PFM-031 Отображение прогресса по каждой цели (график, шкала выполнения).

– Story «PFM-007 Установка и контроль бюджета по категориям»:

а) PFM-032 Создание таблицы/модели «Бюджеты».

б) PFM-033 Интерфейс для установки месячного/недельного бюджета на категорию.

в) PFM-034 API для работы с бюджетами.

г) PFM-035 Логика уведомления при достижении/превышении заданного лимита.

– Story «PFM-008 Аналитика и формирование отчетов»:

а) PFM-036 Модуль расчета статистики и агрегированных данных;

б) PFM-037 Разработка диаграмм и графиков;

в) PFM-038 Реализовать фильтрацию (период, категории);

г) PFM-039 Экспорт отчетов (PDF/Excel) (опционально).

– Story «PFM-009 Интеграция с платежными системами»:

а) PFM-040 Подключение к API банка или электронного кошелька.

б) PFM-041 Реализация автоматической загрузки операций.

в) PFM-042 Сопоставление полученных данных с категориями расходов/доходов.

г) PFM-043 Настройка расписания (cron-job) для регулярного обновления.

– Story «PFM-010 Управление собственными категориями»:

а) PFM-044 Таблица «Пользовательские категории».

б) PFM-045 Интерфейс для создания/редактирования пользовательских категорий.

в) PFM-046 Связь пользовательских категорий с доходами и расходами.

г) PFM-047 Уведомления при удалении категории, используемой в записях.

– Story «PFM-011 Управление общими настройками профиля»:

а) PFM-048 Интерфейс настроек (имя, email, язык приложения, валюта по умолчанию).

б) PFM-049 API для сохранения изменений.

в) PFM-050 Настройка push-уведомлений и e-mail.

г) PFM-051 Просмотр истории входов/действий (при необходимости).

Проведем описание функциональности (Use Case) на примере UseCase: «Добавить расход».

Код и наименование: UC-Expense-Add. Добавление расхода Краткое описание: Позволяет пользователю (User) внести информацию о новом расходе (сумма, категория, дата, описание) и сохранить ее в системе.

Акторы:

– главный (инициирует ВИ): Пользователь (User);

– второстепенный: —.

Предусловия:

– пользователь авторизован в системе;

– у пользователя есть хотя бы одна доступная категория расходов.

Основной поток:

– пользователь открывает раздел «Мои расходы»;

– нажимает кнопку «Добавить расход»;

– заполняет форму (сумма, дата, категория, комментарий);

– нажимает «Сохранить»;

– система валидирует введенные данные;

– система сохраняет данные в базу и обновляет информацию об аналитике.

Приложение отображает обновленный список расходов.

Постусловия:

– новый расход успешно сохранен;

– аналитика по расходам обновлена.

Альтернативный поток:

– если сумма не указана или указана некорректно, система выдает

сообщение об ошибке (АП-1);

– если пользователь передумал сохранять расход, он может нажать «Отмена» (АП-2).

АП-1: Сумма не указана или недопустимый формат:

– АП начинается на шаге 5 основного потока;

– система отображает сообщение «Укажите корректную сумму»;

– пользователь возвращается к редактированию формы.

АП-2: Пользователь отменил операцию:

– АП начинается на шаге 4 основного потока;

– система закрывает форму добавления без сохранения данных.

Специальные требования: быть авторизованным.

Точки расширения: Нет.

Дополнительная информация: Нет.

Рассмотрим пример UseCase: «Установить бюджет на категорию». Код и наименование: UC-Budget-Set. Установка бюджета. Краткое описание: позволяет пользователю установить лимит расходов на месяц для выбранной категории.

Акторы: главный – пользователь (User).

Предусловия: пользователь авторизован; категория существует.

Основной поток:

– пользователь переходит в раздел «Бюджеты»;

– нажимает «Установить новый бюджет»;

– выбирает категорию, указывает сумму и период (например, месяц);

– сохраняет изменения;

– система создает запись бюджета в базе и настраивает механизм уведомления при превышении лимита.

Постусловия:

– лимит установлен;

– при добавлении новых расходов по данной категории система

проверяет и уведомляет при достижении порога.

Альтернативные потоки:

– если категория не выбрана, система уведомляет об ошибке;

– если сумма лимита меньше 0, система уведомляет об ошибке.

Выбор архитектуры зависит от масштабов проекта и требований к развитию. На начальном этапе монолитная архитектура будет самым простым и понятным решением. Преимущества:

– простота развертывания: одно приложение, единая база данных;

– быстрое прототипирование: удобно делать MVP;

– упрощенное тестирование: все модули в одном кодовом пространстве.

В монолитном варианте можно описать основные сущности (таблицы БД):

– Пользователи (Users). Поля: id, email, passwordHash, firstName, lastName, createdAt, role (User/Admin), etc.

– Доходы (Incomes). Поля: id, userId, amount, categoryId, date, comment, createdAt, updatedAt.

– Расходы (Expenses). Поля: id, userId, amount, categoryId, date, comment, createdAt, updatedAt.

– Категории (Categories). Поля: id, userId (для пользовательских), name, type (доход/расход), icon (опционально).

– Бюджеты (Budgets). Поля: id, userId, categoryId, limitAmount, startDate, endDate.

– Цели (Goals). Поля: id, userId, name, targetAmount, currentAmount, targetDate, status.

– Настройки (Settings). Поля: id, userId, currency, language, notificationsEnabled, etc.

– Интеграции (Integrations) (Если нужно хранить параметры подключения к банкам). Поля: id, userId, provider, accessToken,

refreshToken, createdAt, expiresAt.

В случае микросервисной архитектуры сущности могли бы быть распределены по сервисам, например:

– AuthService (регистрация, авторизация, управление пользователями);

– FinanceService (доходы, расходы, категории, бюджеты, цели);

– AnalyticsService (формирование отчетов и аналитики);

– IntegrationService (работа с внешними платежными системами);

Рассмотрим интеграцию с внешним платежным сервисом (или сервисом банка), который может предоставлять транзакции пользователя через API.

Метод HTTP:

– Метод: GET;

– URL: /api/integration/bank/transactions?fromDate=YYYY-MM-DD&toDate=YYYY-MM-DD.

Описание входных параметров приведено в таблице 1.

Таблица 1 – Описание входных параметров

№	Параметр	Тип	Обязательность	Описание	Пример
1	fromDate	string	Да	Начальная дата периода	2025-01-01
2	toDate	string	Да	Конечная дата периода	2025-01-31
3	accountId	string	Нет	Идентификатор счета, если нужно получить конкретный счет	123456789
4	token	string	Да (Auth)	Токен аутентификации, выданный пользователю	abcdef123456

Описание выходных параметров (JSON-ответ) отражено в таблице 2.

Таблица 2 – Описание выходных параметров

№	Поле	Тип	Описание	Пример
1	status	string	Статус запроса (success/error)	success
2	transactions	array	Список транзакций	См. ниже
3	message	string	Сообщение об ошибке (если status=error)	"Invalid token"

Приведем пример запроса:

GET /api/integration/bank/transactions?fromDate=2025-01-01&toDate=2025-01-31&accountId=123456789

Authorization: Bearer abcdef123456

Рассмотрим пример ответа:

{"status": "success",
"transactions": [
{"date": "2025-01-05",
"amount": 50000.00,
"currency": "RUB",
"category": "SALARY",
"description": "Monthly salary"},
{"date": "2025-01-10",
"amount": -1500.00,
"currency": "RUB",
"category": "FOOD",
"description": "Groceries shop"},
{"date": "2025-01-15",
"amount": -3000.00,
"currency": "RUB",
"category": "ENTERTAINMENT",
"description": "Cinema and snacks"}]}

В данном сценарии приложение после получения такого ответа сопоставит категории банка со своими внутренними категориями пользователя и автоматически подгрузит данные в раздел «Доходы» и «Расходы» (при необходимости – с использованием правил трансформации).

Список использованной литературы:

Разработка требований к программному обеспечению / К. Вигерс, Д. Битти. – СПб : БХВ, 2024. – 736 с.

References:

Razrabotka trebovanij k programmnomu obespecheniju / K. Vigers, D. Bitti. – SPb : BHV, 2024. – 736 s.