



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практическим работам №11-14

по дисциплине «Системная и программная инженерия»

Выполнил:

Студенты группы ИКБО-11-22

Берчик А.С.

Андрусенко Л.Д.

Гришин А.В.

Малкин Г.Д.

Гоппен С.Д.

Проверил:

Ст. преподаватель Михайлова Е.К.

СОДЕРЖАНИЕ

1. ПОДГОТОВКА К ТЕСТИРОВАНИЮ ПРОДУКТА.....	3
1.1. Тестовый случай или тестовый сценарий.....	3
1.2. Инструменты для проведения тестирования.....	6
1.3. Тестирование (не-)функциональных требований.....	6
2. ФОРМИРОВАНИЕ ПРОГРАММЫ И МЕТОДИКИ ИСПЫТАНИЙ СИСТЕМЫ.....	7
2.1. Программа и методика испытаний.....	7
3. АНАЛИЗ И ОЦЕНКА РИСКОВ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ПРОЕКТА.....	9
4. АНАЛИЗ ЭКСПЛУАТАЦИИ ПРОЕКТА.....	13
4.1 Технологические угрозы.....	13
4.2 Риск = Угроза + Уязвимость + Актив.....	13
4.3 Обработка рисков.....	14
ПРИЛОЖЕНИЕ А.....	15

1. ПОДГОТОВКА К ТЕСТИРОВАНИЮ ПРОДУКТА

1.1. Тестовый случай или тестовый сценарий

Тестовый случай или тестовый сценарий (test case) – артефакт, описывающий совокупность этапов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части. У стандартного тестового случая есть 5 частей, то есть 5 атрибутов:

1. Порядковый номер тестового сценария

2. **Название тестового сценария.** Из него должно быть понятно, в чем суть тест кейса.

3. **Предусловия тестового сценария.** Это условия, которые необходимы для проведения тест кейса. Они должны быть выполнены еще до запуска тест кейса. Допустим: компания сдает самокаты в поминутную аренду. Нужно провести тест кейс функции, которая уведомляет пользователя о том, что заряд аккумулятора самоката. Предусловием тест кейса будет то, что самокат должен находиться в состоянии аренды.

4. **Порядок действий** в тест кейсе и описания действий в тестовом сценарии.

5. Ожидаемый результат тестового сценария

Таблица 1 – Тестовый сценарий №1

Название:	Регистрация нового пользователя	
Функция:	Регистрация	
Действие Регистрация нового пользователя	Ожидаемый результат Появляется сообщение об успешной регистрации	Результат теста: • пройден
Предусловие		
Открыто приложение, пользователь не зарегистрирован		
Шаги теста		
1. Ввести email и пароль 2. Нажать 'Зарегистрироваться'		

Таблица 2 – Тестовый сценарий №2

Название:	Авторизация с корректными данными	
Функция:	Авторизация	
Действие Авторизация пользователя в системе	Ожидаемый результат Пользователь входит в систему	Результат теста: • пройден
Предусловие		
Пользователь зарегистрирован		
Шаги теста		
1. Ввести email и пароль 2. Нажать кнопку «Войти»		

Таблица 3 – Тестовый сценарий №3

Название:	Авторизация с некорректными данными	
Функция:	Обработка ошибок входа	
Действие Попытка входа с некорректным паролем	Ожидаемый результат Ошибка: неверный логин или пароль	Результат теста: • пройден
Предусловие		
Пользователь зарегистрирован		
Шаги теста		
1. Ввести email и некорректный пароль 2. Нажать кнопку «Войти»		

Таблица 4 – Тестовый сценарий №4

Название:	Добавление доходов	
Функция:	Управление доходами пользователя	
Действие Инициирование добавления новой записи о доходе через интерфейс приложения	Ожидаемый результат Доход добавлен в список	Результат теста: • пройден
Предусловие		
Пользователь авторизован		
Шаги теста		
1. Открыть вкладку “Доходы” 2. Ввести сумму и категорию 3. Нажать кнопку “Сохранить”		

Таблица 5 – Тестовый сценарий №5

Название:	Добавление расходов	
Функция:	Управление расходами пользователя	
Действие Внесение новой расходной операции с указанием суммы и категории	Ожидаемый результат Расход добавлен в список	Результат теста: • пройден
Предусловие		
Пользователь авторизован		
Шаги теста		
1. Открыть вкладку «Расходы» 2. Ввести сумму и категорию 3. Нажать кнопку «Сохранить»		

Таблица 6 – Тестовый сценарий №6

Название:	Установка лимита на категорию		
Функция:	Ограничение расходов по категориям		
Действие Настройка расходов на категорию	ограничения на выбранную категорию	Ожидаемый результат Лимит успешно установлен	Результат теста: ● не пройден
Предусловие			
Пользователь авторизован			
Шаги теста			
<ol style="list-style-type: none"> 1. Перейти во вкладку «Лимиты» 2. Выбрать нужную категорию 3. Ввести значение лимита 4. Сохранить изменения 			

Таблица 7 – Тестовый сценарий №7

Название:	Превышение лимита расходов		
Функция:	Контроль превышения заданных лимитов		
Действие Попытка регистрации расхода, сумма которого превышает установленный лимит		Ожидаемый результат Предупреждение о превышении лимита	Результат теста: ● не пройден
Предусловие			
Установлен лимит по категории			
Шаги теста			
<ol style="list-style-type: none"> 1. Перейти во вкладку «Расходы» 2. Ввести сумму, превышающую лимит 3. Сохранить расход 			

Таблица 8 – Тестовый сценарий №8

Название:	Генерация отчета по расходам		
Функция:	Формирование аналитических отчетов		
Действие Выполнение генерации отчета за определенный период		Ожидаемый результат Отчет отображает данные за выбранный период	Результат теста: ● пройден
Предусловие			
Пользователь авторизован, имеются расходы			
Шаги теста			
<ol style="list-style-type: none"> 1. Перейти в раздел «Отчеты» 2. Выбрать нужный период 3. Нажать кнопку «Сформировать отчет» 			

Таблица 9– Тестовый сценарий №9

Название:	Экспорт данных в CSV		
Функция:	Экспорт пользовательских данных		
Действие Использование функции экспорта данных в формате CSV		Ожидаемый результат Файл CSV успешно сформирован	Результат теста: ● пройден
Предусловие			
Пользователь авторизован, имеются данные			
Шаги теста			
<ol style="list-style-type: none"> 1. Перейти на страницу расхода/дохода 2. Нажать "Экспорт в CSV" 			

Таблица 10 – Тестовый сценарий №10

Название:			Добавление информации о кредите		
Функция:			Управление кредитными обязательствами		
Действие		Ожидаемый результат		Результат теста:	
Внесение новой информации о кредите в систему		Кредит отображается в списке кредитов		● не пройден	
Предусловие					
Пользователь авторизован					
Шаги теста					
1. Открыть кредиты					
2. Ввести данные о кредите					
3. Сохранить данные					

1.2. Инструменты для проведения тестирования

Для тестирования функционала и интерфейса системы UWasting использовались следующие инструменты:

- JUnit — модульное тестирование бизнес-логики и компонентов сервера
- Espresso — UI-тесты для Android-приложения
- Postman — ручное тестирование REST API
- Selenium — автотесты интерфейса Swagger API
- JaCoCo — анализ покрытия кода тестами

1.3. Тестирование (не-)функциональных требований

Check-list (контрольный список задач) — список тестовых проверок, которые необходимо выполнить в тестируемой системе. Простыми словами — это перечисление тест-кейсов без деталей (без последовательных шагов и их результатов), главная цель которого запомнить, что мы проверяем и не дать уйти с намеченного пути. Также чек-лист может выступать в роли отчета по проведенному тестированию.

№	Наименование тестового случая	Действия или входные данные	Ожидаемый результат	Фактический результат	Примечание
1	Регистрация нового пользователя	Ввести email и пароль, нажать 'Зарегистрировать ся'	Появляется сообщение об успешной регистрации	Появляется сообщение об успешной регистрации	UI: Material 3; Android 13
2	Авторизация с корректными данными	Ввести email и пароль, нажать 'Войти'	Пользователь входит в систему	Пользователь входит в систему	JWT login
3	Авторизация с некорректным паролем	Ввести email и неправильный пароль, нажать 'Войти'	Ошибка: неверный логин или пароль	Ошибка: неверный логин или пароль	Проверка обработки ошибок
4	Добавление дохода	Открыть вкладку доходов, ввести сумму и категорию, сохранить	Доход добавлен в список	Доход добавлен в список	RxJava, ViewModel
5	Добавление расхода	Открыть вкладку расходов, ввести сумму и категорию, сохранить	Расход добавлен в список	Расход добавлен в список	RxJava, ViewModel
6	Установка лимита на категорию	Открыть вкладку лимитов, выбрать категорию и ввести лимит	Лимит успешно установлен	-	Сервис лимитов

7	Превышение лимита расходов	Добавить расход, превышающий лимит	Предупреждение о превышении лимита	-	Сработал алерт
8	Генерация отчета по расходам	Открыть отчеты, выбрать период, нажать 'Сформировать отчет'	Отчет отображает данные за выбранный период	Отчет отображает данные за выбранный период	Диаграммы через MPAndroidChart
9	Экспорт данных в CSV	Перейти в настройки, нажать 'Экспорт в CSV'	Файл CSV успешно сформирован	Файл CSV успешно сформирован	Apache Commons CSV
10	Добавление информации о кредите	Открыть кредиты, ввести данные и сохранить	Кредит отображается в списке кредитов	-	Entity Framework: таблица кредитов

Таблица 11 – Чек-лист проведенного тестирования

Дополненная матрица требований находится в Приложении А.

2. ФОРМИРОВАНИЕ ПРОГРАММЫ И МЕТОДИКИ ИСПЫТАНИЙ СИСТЕМЫ

2.1. Программа и методика испытаний

1. Объект испытаний

Объектом испытаний является автоматизированная система управления личными финансами «UWasting». Система реализована в виде Android-приложения с серверной частью на .NET Core и базой данных PostgreSQL. Назначение системы — автоматизация учета, анализа и прогнозирования доходов и расходов пользователей.

2. Цель испытаний

Целью испытаний является проверка соответствия реализованной системы требованиям технического задания, оценка качества и полноты реализации функционала, а также оценка готовности системы к вводу в опытную и постоянную эксплуатацию.

3. Требования к программе

Во время испытаний проверяются следующие функции:

- регистрация и авторизация пользователей;
- добавление и редактирование доходов и расходов;
- установка и контроль лимитов по категориям;
- построение отчетов и графиков;
- прогнозирование трат;
- экспорт данных в формате CSV;
- стабильность работы приложения при типичных и предельных сценариях использования.

4. Требования к программной документации

В состав программной документации, предъявляемой на испытания, входят:

- техническое задание по ГОСТ 34.602-2020;
- руководство пользователя;
- архитектурная и проектная документация (диаграммы UML, DFD, IDEF0);
- отчеты по тестированию (чек-листы, тест-кейсы, протоколы испытаний);

5. Средства и порядок испытаний

Испытания проводятся с использованием следующих средств:

- Android Studio, эмулятор Android;
- Postman и Swagger для API тестирования;
- JUnit и Espresso для автоматических тестов;
- Docker Compose для развёртывания серверной части и базы данных.

Порядок испытаний включает установку системы в тестовом окружении, запуск автоматических и ручных тестов, регистрацию ошибок и их исправление, повторное тестирование, оформление итоговых документов.

6. Методы испытаний

Методы испытаний включают:

- функциональное тестирование;
- регрессионное тестирование;
- нагрузочное тестирование с использованием JMeter;
- тестирование на отказоустойчивость и восстановление после сбоев;
- проверка безопасности (шифрование, аутентификация).

7. Заключение

Программа и методика испытаний составлены в соответствии с ГОСТ 19.301-79 и ГОСТ 34.603-92. Испытания охватывают все основные аспекты функционирования системы UWasting, включая стабильность, надежность, безопасность и соответствие ТЗ. Результаты испытаний подтверждают готовность системы к вводу в эксплуатацию.

3. АНАЛИЗ И ОЦЕНКА РИСКОВ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ПРОЕКТА

Таблица 12 – Описание рисков в формате «причина-риск-эффект»

№	Риск (причина-риск-эффект)
1	<ul style="list-style-type: none">- Изменение требований заказчика в процессе разработки.- Необходимость переработки уже реализованных функций.- Задержка сроков сдачи проекта.
2	<ul style="list-style-type: none">- Недостаточная вовлеченность команды.- Снижение мотивации и продуктивности.- Задержка сроков выполнения задач.
3	<ul style="list-style-type: none">- Нехватка коммуникации внутри команды.- Пробелы в знаниях и дублирование задач- Снижение продуктивности и качества работы.
4	<ul style="list-style-type: none">- Плохая документация.- Неэффективное использование времени.- Увеличение времени на выполнение задач.
5	<ul style="list-style-type: none">- Незапланированное отсутствие ключевого члена команды.- Задержка сроков выполнения задач- Дезорганизация и снижение мотивации команды.
6	<ul style="list-style-type: none">- Слабая коммуникация с заказчиком.- Задержки проекта из-за отсутствия обратной связи.- Демотивация команды и задержки в выполнении задач.
7	<ul style="list-style-type: none">- Невыполнение проекта в срок.- Неудовлетворенность заказчика и плохая рабочая атмосфера.- Задержки в реализации проекта.
8	<ul style="list-style-type: none">- Недостаточное тестирование.- Выпуск продукта с ошибками.- Неудовлетворенность пользователей и затраты на исправление ошибок.
9	<ul style="list-style-type: none">- Проблемы с безопасностью данных.- Утечка данных пользователей.- Потеря доверия пользователей и юридические последствия.
10	<ul style="list-style-type: none">- Недостаточная производительность серверов.- Медленная работа приложения.- Неудовлетворенность пользователей и снижение популярности приложения.

Таблица 13 – Определение важности рисков

№	«Причина-риск-эффект»	Вероятность (1-10)	Последствия (1-10)	Важность (В*П)
1	- Изменение требований заказчика в процессе разработки. - Необходимость переработки уже реализованных функций. - Задержка сроков сдачи проекта.	5	8	40
2	- Недостаточная вовлеченность команды. - Снижение мотивации и продуктивности. - Задержка сроков выполнения задач.	6	7	42
3	- Нехватка коммуникации внутри команды. - Пробелы в знаниях и дублирование задач - Снижение продуктивности и качества работы.	7	7	49
4	- Плохая документация. - Неэффективное использование времени. - Увеличение времени на выполнение задач.	6	6	36
5	- Незапланированное отсутствие ключевого члена команды. - Задержка сроков выполнения задач - Дезорганизация и снижение мотивации команды.	7	8	56
6	- Слабая коммуникация с заказчиком. - Задержки проекта из-за отсутствия обратной связи. - Демотивация команды и задержки в выполнении задач.	7	7	49
7	- Невыполнение проекта в срок. - Неудовлетворенность заказчика и плохая рабочая атмосфера. - Задержки в реализации проекта.	8	9	72
8	- Недостаточное тестирование. - Выпуск продукта с ошибками. - Неудовлетворенность пользователей и затраты на исправление ошибок.	6	8	48
9	- Проблемы с безопасностью данных. - Утечка данных пользователей. - Потеря доверия пользователей и юридические последствия.	5	9	45
10	- Недостаточная производительность серверов. - Медленная работа приложения. - Неудовлетворенность пользователей и снижение популярности приложения.	6	7	42

Таблица 14 – Матрица рисков

ВЕРОЯТНОСТЬ	УРОВЕНЬ УЩЕРБА				
	Несущественные (1-2)	Низкие (3-4)	Средние (5-6)	Существенные (7-8)	Катастрофические (9-10)
Весьма вероятно (9-10)					7
Вероятно (7-8)				5, 6	
Возможно (5-6)			2, 4, 8, 10	1, 3	
Маловероятно (3-4)					
Крайне маловероятно (1-2)					

Для понимания следующей таблицы, необходимо знать о стратегиях реагирования на риски проекта:

- избегание (**avoidance**) – устранение источников риска. Например, если сотрудники собираются попасть в зону урагана, можно отменить их поездку, чтобы избежать риска;
- передача (**transference**) – привлечение третьей стороны, принимающей риски на себя. Например, страховка – форма передачи риска, когда компания платит небольшую сумму третьей стороне, чтобы избежать полного финансового бремени, например, ущерба имуществу;
- снижение (**mitigation**) – уменьшение вероятности воплощения угрозы. Например, промышленная пекарня не может избежать активных источников тепла, но может принять меры, чтобы снизить опасность для сотрудников и имущества;
- принятие (**acceptance**) – пассивная форма предполагает осознанную готовность к угрозе, а активная форма – согласование плана действия при наступлении непредвиденных, но принимаемых обстоятельств. Например, если планируется мероприятие на открытом воздухе с 15%-ной вероятностью дождя, стратегия принятия такого небольшого риска может быть более разумной, чем изменение расписания.

Таблица 15 – Таблица планов реагирования

№	«Причина-риск-эффект»	Стратегия	Основной план	Отходной план
1	<ul style="list-style-type: none"> - Изменение требований заказчика в процессе разработки. - Необходимость переработки уже реализованных функций. - Задержка сроков сдачи проекта. 	Mitigate (снижение)	Регулярные встречи с заказчиком для уточнения требований.	Создание гибкого плана разработки с учетом возможных изменений.
2	<ul style="list-style-type: none"> - Недостаточная вовлеченность команды. - Снижение мотивации и продуктивности. - Задержка сроков выполнения задач. 	Avoid (избегание)	Проведение регулярных встреч и мотивационных мероприятий.	Внедрение системы поощрений за выполнение задач.
3	<ul style="list-style-type: none"> - Нехватка коммуникации внутри команды. - Пробелы в знаниях и дублирование задач - Снижение продуктивности и качества работы. 	Mitigate (снижение)	Внедрение регулярных встреч и использование инструментов для управления задачами.	Назначение координатора для улучшения коммуникации.
4	<ul style="list-style-type: none"> - Плохая документация. - Неэффективное использование времени. - Увеличение времени на выполнение задач. 	Accept (принятие)	Назначение ответственного за документацию и регулярное обновление.	Использование шаблонов и автоматизированных инструментов для документации.
5	<ul style="list-style-type: none"> - Незапланированное отсутствие ключевого члена команды. - Задержка сроков выполнения задач - Дезорганизация и снижение мотивации команды. 	Mitigate (снижение)	Обеспечить кросс-функциональное обучение членов команды для взаимозаменяемости.	Перераспределить задачи между оставшимися членами команды.
6	<ul style="list-style-type: none"> - Слабая коммуникация с заказчиком. - Задержки проекта из-за отсутствия обратной связи. - Демотивация команды и задержки в выполнении задач. 	Mitigate (снижение)	Назначение ответственного за коммуникацию с заказчиком.	Использование автоматизированных систем уведомлений.
7	<ul style="list-style-type: none"> - Невыполнение проекта в срок. - Неудовлетворенность заказчика и плохая рабочая атмосфера. - Задержки в реализации проекта. 	Mitigate (снижение)	Внедрение системы мониторинга прогресса и регулярные отчеты.	Пересмотр приоритетов задач и перераспределение ресурсов.
8	<ul style="list-style-type: none"> - Недостаточное тестирование. - Выпуск продукта с ошибками. - Неудовлетворенность пользователей и затраты на исправление ошибок. 	Mitigate (снижение)	Внедрение автоматизированного тестирования и увеличение покрытия тестами.	Назначение дополнительных ресурсов на тестирование.
9	<ul style="list-style-type: none"> - Проблемы с безопасностью данных. - Утечка данных пользователей. - Потеря доверия пользователей и юридические последствия. 	Mitigate (снижение)	Регулярные аудиты безопасности и обновление мер защиты.	Внедрение дополнительных мер безопасности и обучение команды.
10	<ul style="list-style-type: none"> - Недостаточная производительность серверов. - Медленная работа приложения. - Неудовлетворенность пользователей и снижение популярности приложения. 	Mitigate (снижение)	Оптимизация кода и использование кэширования.	Обновление серверного оборудования.

4. АНАЛИЗ ЭКСПЛУАТАЦИИ ПРОЕКТА

4.1 Технологические угрозы

Таблица 16 - Влияние технологических угроз на функционал

Угроза	Влияние на функционал (0-5)	Обоснование
Потеря данных	5	Полная недоступность системы, так как серверное оборудование критически важно для работы приложения.
Нарушение целостности данных	4	Частичная недоступность функционала, так как данные пользователей не будут доступны для обработки.
Снижение производительности системы	3	Функциональность доступна только в определенные интервалы времени, так как ошибки могут привести к временным сбоям.

4.2 Риск = Угроза + Уязвимость + Актив

Таблица 17 - Влияние технологических угроз на функционал

Риск	Вероятность возникновения (0-10)	Обоснование
Потеря данных из-за отказа серверного оборудования	4	Средняя вероятность, так как серверное оборудование может выйти из строя из-за различных факторов, таких как перегрев или аппаратные сбои.
Нарушение целостности данных из-за отказа базы данных	5	Высокая вероятность, так как базы данных могут быть подвержены атакам или сбоям.
Снижение производительности системы из-за ошибок в программном обеспечении	6	Высокая вероятность, так как программное обеспечение может содержать ошибки, которые проявляются в процессе эксплуатации.

4.3 Обработка рисков

Таблица 18 - Методы обработки рисков

Риск	Метод обработки	Обоснование
Потеря данных из-за отказа серверного оборудования	Снижение риска	Внедрение резервного серверного оборудования для обеспечения непрерывности работы за счет архитектурных и технических изменений.
Нарушение целостности данных из-за отказа базы данных	Принятие риска	Регулярное резервное копирование данных и использование репликации базы данных для обеспечения доступности данных. Однако, полное устранение риска невозможно из-за высокой стоимости и сложности полной защиты.
Снижение производительности системы из-за ошибок в программном обеспечении	Передача риска	Передача риска провайдеру облачных услуг, который будет отвечать за мониторинг и исправление ошибок в программном обеспечении, с заключением SLA для обеспечения качества обслуживания.

ЗАКЛЮЧЕНИЕ

В ходе тестирования программного продукта UWasting была составлена и дополнена матрица требований с результатами тестирования. Каждое функциональное требование было проверено с использованием ручных и автоматических сценариев, охватывающих ключевые функции приложения: регистрацию, управление доходами и расходами, установку лимитов, экспорт данных, формирование отчётов, а также защиту и сохранность пользовательской информации.

Результаты тестирования позволили выявить как успешно реализованные функции, так и проблемные участки, требующие доработки (например, обработка лимитов и кредитов). Дополненная матрица обеспечивает прослеживаемость выполнения требований и является основой для последующей итерации улучшений.

Проведённое тестирование подтверждает, что большинство требований, предъявленных к системе, были успешно реализованы и прошли проверку. Документ может быть использован как часть итоговой отчётной документации по проверке соответствия ПО требованиям технического задания и готовности продукта к внедрению.

ПРИЛОЖЕНИЕ А

Таблица 1 – Дополненная матрица требований

№	Функциональность	Описание требования	Результат тестирования
1	Регистрация и авторизация	Пользователь должен иметь возможность зарегистрироваться и войти в систему	Пройдено
2	Добавление доходов	Пользователь может добавлять записи о доходах	Пройдено
3	Добавление расходов	Пользователь может добавлять записи о расходах	Пройдено
4	Установка лимитов	Пользователь может задать лимит по категории трат	Не пройдено
5	Экспорт данных	Возможность экспорта данных в CSV	Пройдено
6	Прогнозирование	Система может прогнозировать траты	Пройдено
7	Уведомления	Пользователь получает уведомления при достижении лимита	Не пройдено
8	Работа без интернета	Приложение сохраняет данные оффлайн и синхронизирует их позже	Пройдено
9	Шифрование данных	Все пользовательские данные шифруются при хранении и передаче	Пройдено
10	Отчёты и аналитика	Система строит графики и отчёты по тратам	Пройдено